California Instruments

Revision A
September 2005

P/N 7004-988

CS Series
AC Current Source
Programming Manual

TEL: +1 (858) 677-9040
FAX: +1 (858) 677-0940
Email: sales@calinst.com
Web Site: http://www.calinst.com

Refers to:

CS Series AC Current Source

Models:

Single chassis:	3000CS, 4500CS
Multiple chassis:	6000CS/2, 9000CS/2, 13500CS/3, 18000CS/4

Manual revision: A.

SAFETY SUMMARY

This power source contains high voltage and current circuits, which are potentially lethal. Because of its size and weight, mechanical stability must be ensured. The following safety guidelines must be followed when operating or servicing this equipment. These guidelines are not a substitute for vigilance and common sense. California Instruments assumes no liability for the customer's failure to comply with these requirements. If the power source is used in a manner not specified by California Instruments, the protection provided by the equipment may be impaired.

BEFORE APPLYING POWER

1. Verify the correct three phase input voltage is applied to the unit. Input ratings are shown on the model and serial number tag located at the rear of the unit.

2. The chassis and cabinet of this power source must be grounded to minimize shock hazard. A chassis ground is provided at the input terminal block. This is located in the front of the cabinet on the lower left hand side. The lower front cover panel must be removed to access the line input and ground connections. The chassis ground must be connected to an electrical ground through an insulated wire of sufficient gauge.

FUSES

Use only fuses of the specified current, voltage, and protection speed (slow blow, normal blow, fast blow) rating. Do not short out the fuse holder or use a repaired fuse.

DO NOT OPERATE IN A VOLATILE ATMOSPHERE

Do not operate the power source in the presence of flammable gases or fumes.

DO NOT TOUCH ENERGIZED CIRCUITS

Disconnect the power cable before servicing this equipment. Even with the power cable disconnected, high voltage can still exist on some circuits. Discharge these voltages before servicing. Only qualified service personnel may remove covers, replace components or make adjustments.

DO NOT SERVICE ALONE

Do not remove covers, replace components, or make adjustments unless another person, who can administer first aid, is present.

DO NOT EXCEED INPUT RATINGS

Do not exceed the rated input voltage or frequency. Additional hazards may be introduced because of component failure or improper operation.

DO NOT MODIFY INSTRUMENT OR SUBSTITUTE PARTS

Do not modify this instrument or substitute any parts. Additional hazards may be introduced because of component failure or improper operation.

MOVING THE POWER SOURCE

When moving the power source, observe the following:

1. Remove all AC power to unit.
2. Do not attempt to lift alone. Each power source chassis must be lifted by at least two persons or by mechanical means to prevent the chance of injury.
3. When installing / removing from an instrument cabinet, use a suitable lift platform. Rack slides are recommended for cabinet installation.

ALLOW CAPACITORS TO DISCHARGE

Capacitors in the power source may hold a hazardous electrical charge even if the power source has been disconnected from the mains supply. Allow capacitors to discharge to a safe voltage before servicing internal circuits or touching exposed pins of mains supply connectors.

SAFETY SYMBOLS:

⎓	THIS SYMBOL INDICATES DIRECT CURRENT
∼	THIS SYMBOL INDICATES ALTERNATING CURRENT
≂	THIS SYMBOL INDICATES BOTH DIRECT AND ALTERNATING CURRENT
3∼	THIS SYMBOL INDICATES THREE—PHASE ALTERNATING CURRENT
⏚	THIS SYMBOL INDICATES EARTH (GROUND) TERMINAL
(⏚)	THIS SYMBOL INDICATES PROTECTIVE CONDUCTOR TERMINAL
⫨	THIS SYMBOL INDICATES FRAME OR CHASSIS TERMINAL
|	THIS SYMBOL INDICATES ON (SUPPLY)
○	THIS SYMBOL INDICATES OFF (SUPPLY)
⚡	THIS SYMBOL INDICATES CAUTION, RISK OF ELECTRIC SHOCK
⚠	THIS SYMBOL INDICATES CAUTION (REFER TO ACCOMPANYING DOCUMENTS)

Table of Contents

1. Introduction 8
1.1 Documentation Summary 8
1.2 Manual organization and format 9
1.3 Introduction to Programming 9
2. Introduction to SCPI 11
2.1 Conventions Used in This Manual 11
2.2 The SCPI Commands and Messages 11
2.3 Using Queries 14
2.4 Coupled Commands 14
2.5 Structure of a SCPI Message 14
2.6 SCPI Data Formats 17
3. System Considerations 19
3.1 IEEE-488 / GPIB Interface 19
3.2 USB Interface 20
3.3 -LAN Option 26
3.4 RS232C Serial Interface 28
3.5 Instrument Drivers and Application Software 28
4. SCPI Command Reference 29
4.1 Introduction 29
4.2 Subsystem Commands 30
4.3 Calibration Subsystem 31
4.4 Diagnostic Subsystem 33
4.5 Instrument Subsystem 34
4.6 Limit Subsystem 35
4.7 Array Measurement Subsystem 37
4.8 Current Measurement Subsystem 43
4.9 Frequency Measurement Subsystem 48
4.10 Power Measurement Subsystem 49
4.11 Voltage Measurement Subsystem 51
4.12 Output Subsystem 55
4.13 Power On Subsystem 59
4.14 Sense Subsystem - Sweep 61
4.15 Source Subsystem - Voltage 63
4.16 Source Subsystem - Frequency 65
4.17 Source Subsystem - Function 68
4.18 Source Subsystem - List 70
4.19 Source Subsystem - Phase 75
4.20 Source Subsystem - Pulse 77
4.21 Source Subsystem - Current 80
4.22 Status Subsystem Commands 83
4.23 System Commands 90
4.24 Trace Subsystem Commands 93
4.25 Trigger Subsystem 95

5. Common Commands 100
5.1 *CLS 101
5.2 *ESR? 101
5.3 *IDN? 102
5.4 *OPC 102
5.5 *OPT? 102
5.6 *PSC 102
5.7 *RCL 103
5.8 *RST 104
5.9 *SAV 105
5.10 *SRE 105
5.11 *STB? 105
5.12 *TRG 106
5.13 *TST? 106
5.14 *WAI 107
6. Programming Examples 108
6.1 Introduction 108
6.2 Programming the Output 108
6.3 Programming Output Transients 110
6.4 Step and Pulse Transients 112
6.5 List Transients 114
6.6 Triggering Output Changes 115
6.7 Making Measurements 119
6.8 Controlling the Instantaneous Voltage and Current Data Buffers 123
6.9 Downloading Arbitrary Waveforms 127
6.10 Command Processing Times 128
7. Programming the Status and Event Registers 129
7.1 Power-On Conditions 129
7.2 Operation Status Group 129
7.3 Questionable Status Group 132
7.4 Questionable Instrument Isummary Status Group 133
7.5 Standard Event Status Group 134
7.6 Status Byte Register 135
7.7 Examples 136
7.8 Remote Inhibit and Discrete Fault Indicator 139
7.9 SCPI Command Completion 140
8. Option Commands 141
8.1 Introduction 141
Appendix A: SCPI Command tree 142
Appendix B: SCPI Conformance Information 144
Appendix C: Error Messages 146
Index 153

Table of Figures

Figure 2-1: Partial Command Tree 12
Figure 2-2: Command Message Structure.......... 15
Figure 3-1: Windows XP Device Manager - USB Port 22
Figure 3-2: Windows XP Device Manager – Virtual Com Port 25
Figure 3-3: Pinging AC Source LAN IP address. 27
Figure 6-1: Model of transient system.......... 112
Figure 6-2: Model of output trigger system. 116
Figure 6-3: Model of Measurement triggers.......... 121
Figure 6-4: Pre- and Post Event Triggering. 126
Figure 7-1: Status Register Model. 130
Figure 7-2: SMA Connector Trigger Model. 138

Table of Tables

Table 2-1: Command parameters Suffixes and Multipliers.......... 17
Table 4-1: PULSe:HOLD = WIDTh parameters.......... 78
Table 4-2: PULSe:HOLD = DCYCle parameters 78
Table 4-3: Bit Configuration of Status Operation Registers.......... 84
Table 4-4: Bit Configuration of Questionable Registers 85
Table 4-5: Bit Configuration of Questionable Instrument Summary Registers.......... 87
Table 5-1: Bit Configuration of Standard Event Status Enable Register 101
Table 5-2 : factory-defined *RST states.......... 104
Table 5-3: Bit Configuration of Status Byte Register 106
Table 6-1: Command Processing Times. 128
Table 7-1: Operation Status registers.......... 129
Table 7-2: Bit Configurations of Status Registers.......... 131
Table 7-3: Questionable Status registers 132
Table 7-4: Questionable Instrument Isummary Status registers 133
Table 8-1: SCPI error codes and messages.......... 151

1. Introduction

This instruction manual (P/N 7004-988) contains programming information for the CS Series AC current source.

You will find the following information in this manual:

Chapter 2	Introduction to SCPI
Chapter 3	System Considerations
Chapter 4	SCPI Command Reference
Chapter 5	Common Commands
Chapter 6	Programming Examples
Chapter 7	Programming the Status and Event Registers
Chapter 8	Options
Appendix A	SCPI command tree
Appendix B	SCPI conformance information
Appendix C	Error messages

1.1 Documentation Summary

The following document is related to this Programming Manual and may have additional helpful information for using the AC source.

- CS Series User's Manual. P/N 7004-987 Includes specifications and supplemental characteristics, how to use the front panel, how to connect to the instrument, and calibration procedures.

1.1.1 External References

SCPI References

The following documents will assist you with programming in SCPI:

- Beginner's Manual to SCPI. Highly recommended for anyone who has not had previous experience programming with SCPI.
- Controller programming manuals: consult the documentation supplied with the IEEE-488 controller or IEEE-488 PC plug in card for information concerning general IEEE-488.2 conventions and concepts.

The following are two formal documents concerning the IEEE-488 interface:

- ANSI/IEEE Std. 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation. Defines the technical details of the IEEE-488 interface. While much of the information is beyond the need of most programmers, it can serve to clarify terms used in this manual and in related documents.
- ANSI/IEEE Std. 488.2-1987 IEEE Standard Codes, Formats, Protocols, and Common Commands. Recommended as a reference only if you intend to do fairly sophisticated programming. Helpful for finding precise definitions of certain types of SCPI message formats, data types, or common commands.

The above two documents are available from the IEEE (Institute of Electrical and Electronics Engineers), 345 East 47th Street, New York, NY 10017, USA or via the web at www.ieee.org .

1.2 Manual organization and format

All user documentation for California Instruments power sources is provided on CDROM in electronic format. (Adobe Portable Document Format) The required Adobe PDF viewer is supplied on the same CDROM. This manual may be printed for personal use if a hardcopy is desired. To request a hardcopy from California Instruments, contact customer service at support@calinst.com. There will be an additional charge for printed manuals.

This manual contains information on programming the CS using either the GPIB, USB and RS232C interface. Refer to the CS Series User manual (P/N 7004-987) for information on front panel operation, installation and calibration.

1.3 Introduction to Programming

This section provides some general information regarding programming instrumentation and available interface types.

1.3.1 IEEE-488 Capabilities of the AC source

All AC source functions except for setting the IEEE-488 address are programmable over the IEEE-488. The IEEE 488.2 capabilities of the AC source are listed in Chapter 2 of the User's Manual.

The AC source operates from an IEEE-488 address that is set from the front panel. To set the IEEE-488 address, press the MENU key on the front panel repeatedly until the CONFIGURATION entry is shown on the LCD display.

Move the indicator on the right hand side of the display to point to CONFIGURATION and press the ENTER key.

This will display the IEEE ADRRESS currently set. To change the address, use the shuttle knob to increment or decrement the value. Press the ENTER key to confirm your selection.

To set up the GPIB/IEEE-488 interface on a Windows XP PC, refer to section 3.1, “IEEE-488 / GPIB Interface”.

1.3.2 USB Capabilities of the AC source

All AC source functions are programmable over the USB interface. The USB capabilities of the AC source are listed in Chapter 2 of the User's Manual. Some capabilities supported on the GPIB interface such as ATN, GET and SRQ interrupts do not apply to the USB interface. The USB interface operates internally at a fixed baudrate of 460800 baud but USB 2.0 burst transfer rates are supported.

To set up the USB interface on a Windows XP PC, refer to section 3.2, “ USB Interface”.

The USB interface may be used to install updated firmware for the CS controller if needed. Firmware updates and a Flash Loader utility program and instructions are available from the California Instruments website for this purpose. (www.calinst.com)

1.3.3 LAN Capabilities of the AC source

All AC source functions are programmable over the LAN (Ethernet) interface if the –LAN option is installed. The LAN capabilities of the AC source are listed in Chapter 2 of the User's Manual. Some capabilities supported on the GPIB interface such as ATN, GET and SRQ interrupts do not apply to the LAN interface. The LAN interface operates internally at a fixed baudrate of 460800 baud but autodetection of 10Base-T, 100Base-T and 1000Base-T is supported.

To set up the LAN interface on a Windows XP PC, refer to section 3.3, -LAN Option.

1.3.4 RS232C Capabilities of the AC source

All AC source functions are programmable over the RS232C interface. The RS232C capabilities of the AC source are listed in Chapter 2 of the User's Manual. Some capabilities supported on the GPIB interface such as ATN, GET and SRQ interrupts do not apply to the RS232C interface. Baudrates from 9600 to 115200 are supported.

To set up the RS232C interface, refer to section 3.4, “RS232C Serial Interface”.

The RS232C interface may be used to install updated firmware for the controller if needed. Firmware updates and a Flash Loader utility program and instructions are available from the California Instruments website for this purpose. (www.calinst.com)

2. Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over the IEEE-488. SCPI is layered on top of the hardware-portion of IEEE 488.2. The same SCPI commands and parameters control the same functions in different classes of instruments.

2.1 Conventions Used in This Manual

Angle brackets <> Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.

Vertical bar | Vertical bars separate alternative parameters. For example, NORM | TEXT indicates that either "TEXT" or "NORM" can be used as a parameter.

Square Brackets [] Items within square brackets are optional. The representation [SOURce:]LIST means that SOURce: may be omitted.

Braces {} Braces indicate parameters that may be repeated zero or more times. It is used especially for showing arrays. The notation <A> <,B> shows that parameter "A" must be entered, while parameter "B" may be omitted or may be entered one or more times.

2.2 The SCPI Commands and Messages

2.2.1 Types of SCPI Commands

SCPI has two types of commands, common and subsystem.

- Common commands generally are not related to specific operation but to

  controlling overall AC source functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: *RST, *IDN?, *SRE 8

- Subsystem commands perform specific AC source functions. They are organized

  into an inverted tree structure with the "root" at the top. Some are single commands while others are grouped within specific subsystems.

Refer to appendix A for the AC source SCPI tree structure.

2.2.2 Types of SCPI Messages

There are two types of SCPI messages, *program* and *response*.

- A *program* message consists of one or more properly formatted SCPI commands sent from the controller to the AC source. The message, which may be sent at any time, requests the AC source to perform some action.
- A *response* message consists of data in a specific SCPI format sent from the AC source to the controller. The AC source sends the message only when commanded by a program message called a "query."

2.2.3 The SCPI Command Tree

As previously explained, the basic SCPI communication method involves sending one or more properly formatted commands from the SCPI command tree to the instrument as program messages. Figure 2-1 shows a portion of a subsystem command tree, from which you access the commands located along the various paths (you can see the complete tree in appendix A).

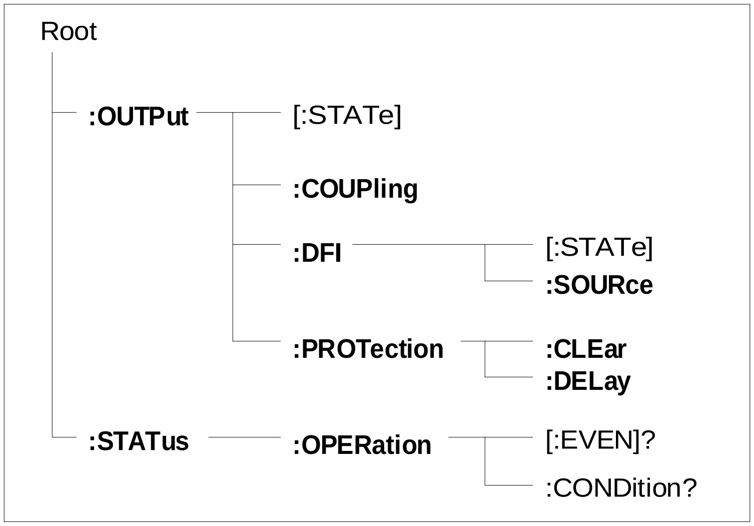


Figure 2-1: Partial Command Tree

The Root Level

Note the location of the ROOT node at the top of the tree. Commands at the root level are at the top level of the command tree. The SCPI interface is at this location when:

- the AC source is powered on
- a device clear (DCL) is sent to the AC source
- the SCPI interface encounters a message terminator (LF)
- the SCPI interface encounters a root specifier (:)

Active Header Path

In order to properly traverse the command tree, you must understand the concept of the active header path. When the AC source is turned on (or under any of the other conditions listed above), the active path is at the root. That means the SCPI interface is ready to accept any command at the root level, such as OUTPut or STATe.

If you enter OUTPut, the active header path moves one colon to the right. The interface is now ready to accept :STATe, :COUPling,:DFI, or :PROTection as the next header. You must include the colon, because it is required between headers.

If you now enter :PROTection, the active path again moves one colon to the right. The interface is now ready to accept either :CLEar or :DELay as the next header.

If you now enter :CLEar, you have reached the end of the command string. The active header path remains at :CLEar. If you wished, you could have entered :CLEar;DELay 20 and it would be accepted as a compound message consisting of:

1. OUTPut:PROTection:CLEAr and

2. OUTPut:PROTection:DELay 20.

The entire message would be:

```
OUTPut:PROTection:CLEar;DELay 20
```

The message terminator after DELay 20 returns the path to the root.

The Effect of Optional Headers

If a command includes optional headers, the interface assumes they are there. For example, if you enter OUTPut OFF, the interface recognizes it as OUTPut:STATe OFF. This returns the active path to the root (:OUTPut). But if you enter OUTPut:STATe OFF, then the active path remains at :STATe. This allows you to send

```
OUTPut:STATe OFF;PROTection:CLEar
```

in one message. If you tried to send

```
OUTPut OFF;PROTection:CLEar
```

the header path would return to :OUTPut instead of :PROTection.

The optional header [SOURce] precedes the current, frequency, function, phase, pulse, list, and voltage subsystems. This effectively makes :CURRent,:FREQuency, :FUNCtion, :PHASe, :PULse, :LIST, and :VOLTage root-level commands.

Moving Among Subsystems

In order to combine commands from different subsystems, you need to be able to restore the active path to the root. You do this with the root specifier (:). For example, you could clear the output protection and check the status of the Operation Condition register as follows:

```
OUTPut:PROTection:CLEAr
STATus:OPERation:CONDition?
```

Because the root specifier resets the command parser to the root, you can use the root specifier and do the same thing in one message:

```
OUTPut:PROTection:CLEAr;:STATus:OPERation:CONDition?
```

The following message shows how to combine commands from different subsystems as well as within the same subsystem:

```
VOLTage:LEVel 70;PROTection 80;:CURRent:LEVel 3;PROTection:STATe ON
```

Note the use of the optional header LEVel to maintain the correct path within the voltage and current subsystems and the use of the root specifier to move between subsystems.

Note: The "Enhanced Tree Walking Implementation" given in appendix A of the IEEE 488.2 standard is not implemented in the AC source.

Including Common Commands

You can combine common commands with system commands in the same message. Treat the common command as a message unit by separating it with a semicolon (the message unit separator). Common commands do not affect the active header path; you may insert them anywhere in the message.

```
CURRent:TRIGger 7.5;INITialize;*TRG
OUTPut OFF;*RCL 2;OUTPut ON
```

2.3 Using Queries

Observe the following precautions with queries:

- Set up the proper number of variables for the returned data.
- Read back all the results of a query before sending another command to the AC source. Otherwise a Query Interrupted error will occur and the unreturned data will be lost.

2.4 Coupled Commands

When commands are coupled it means that the value sent by one command is affected by the settings of the other commands. The following commands are coupled in the AC source:

- The step, pulse, and list commands that control output current
- The pulse commands that program the width, duty cycle, period, and the hold parameter

As explained later in chapter 4, the order in which data is sent by these coupled commands can be important when more than one parameter is changed.

2.5 Structure of a SCPI Message

SCPI messages consist of one or more message units ending in a message terminator. The terminator is not part of the syntax, but implicit in the way your programming language indicates the end of a line (such as a newline or end-of-line character).

2.5.1 The Message Unit

The simplest SCPI command is a single message unit consisting of a command header (or keyword) followed by a message terminator.

```
ABORt<newline>
VOLTage?<newline>
```

The message unit may include a parameter after the header. The parameter usually is numeric, but it can be a string:

```
VOLTage 20<newline>
VOLTage MAX<newline>
```

2.5.2 Combining Message Units

The following command message is briefly described here, with details in subsequent paragraphs.

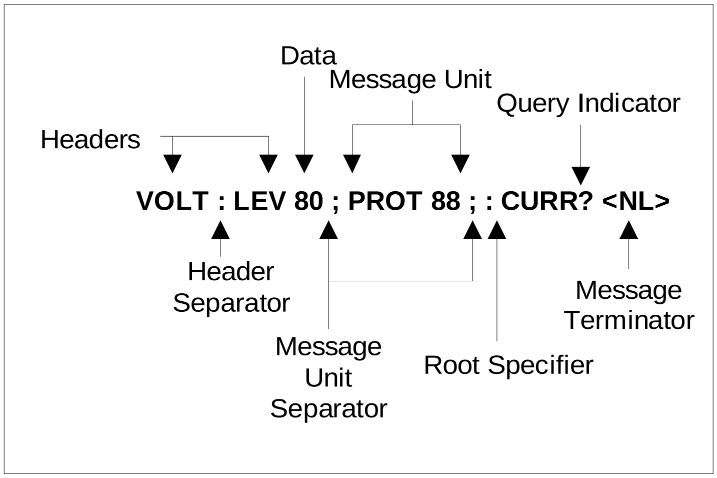


Figure 2-2: Command Message Structure

The basic parts of the above message are:

Message Component	Example
Headers	VOLT LEV PROT CURR
Header Separator	The colon in VOLT:LEV
Data	80 88
Data Separator	The space in VOLT 80 and PROT 88
Message Units	VOLT:LEV 80 PROT 88 CURR?
Message Unit Separator	The semicolons in VOLT:LEV 80; and PROT 88;
Root Specifier	The colon in PROT 88;:CURR?
Query Indicator	The question mark in CURR?
Message Terminator	The <NL> (newline) indicator. Terminators are not part of the SCPI syntax

2.5.3 Headers

Headers are instructions recognized by the AC source. Headers (which are sometimes known as "keywords") may be either in the long form or the short form.

Long Form The header is completely spelled out, such as VOLTAGE, STATUS, and DELAY.

Short Form The header has only the first three or four letters, such as VOLT, STAT, and DEL.

The SCPI interface is not sensitive to case. It will recognize any case mixture, such as TRIGGER, Trigger, TRIGger. Short form headers result in faster program execution.

Header Convention

In the command descriptions in chapter 3 of this manual, headers are emphasized with boldface type. The proper short form is shown in upper-case letters, such as DELay.

Header Separator

If a command has more than one header, you must separate them with a colon (VOLT:PROT OUTPut:RELay:POLarity).

Optional Headers

The use of some headers is optional. Optional headers are shown in brackets, such as OUTPut[:STATe] ON. As previously explained under "The Effect of Optional Headers", if you combine two or more message units into a compound message, you may need to enter the optional header.

2.5.4 Query Indicator

Following a header with a question mark turns it into a query (VOLTage?, VOLTage:PROTection?). If a query contains a parameter, place the query indicator at the end of the last header (VOLTage:PROTection? MAX).

2.5.5 Message Unit Separator

When two or more message units are combined into a compound message, separate the units with a semicolon (STATus:OPERation?;QUEStionable?).

2.5.6 Root Specifier

When it precedes the first header of a message unit, the colon becomes the root specifier. It tells the command parser that this is the root or the top node of the command tree. Note the difference between root specifiers and header separators in the following examples:

OUTPut:PROTection:DELay .1	All colons are header separators
:OUTPut:PROTection:DELay .1	Only the first colon is a root specifier
OUTPut:PROTection:DELay .1;:VOLTage 12.5	Only the third colon is a root specifier

Note: You do not have to precede root-level commands with a colon; there is an implied colon in front of every root-level command.

2.5.7 Message Terminator

A terminator informs SCPI that it has reached the end of a message. Three permitted message terminators are:

- newline (<NL>), which is ASCII decimal 10 or hex 0A.
- end or identify (<END>)
- both of the above (<NL><END>).

In the examples of this manual, there is an assumed message terminator at the end of each message. If the terminator needs to be shown, it is indicated as <NL> regardless of the actual terminator character.

2.6 SCPI Data Formats

All data programmed to or returned from the AC source is ASCII. The data may be numerical or character string.

2.6.1 Numerical Data Formats

Symbol **Data Form**

Talking Formats

<NR1> Digits with an implied decimal point assumed at the right of the least-significant digit. Examples: 273

<NR2> Digits with an explicit decimal point. Example: .0273

<NR3> Digits with an explicit decimal point and an exponent. Example: 2.73E+2

<Bool> Boolean Data. Example: 0 | 1or ON | OFF

Listening Formats

<Nrf> Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273 273. 2.73E2

<Nrf+> Expanded decimal format that includes <Nrf> and MIN MAX. Examples: 273 273. 2.73E2 MAX. MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter.

<Bool> Boolean Data. Example: 0 | 1

Class	Suffix	Unit	Multiplier
Amplitude	V	Volt	MV (millivolt)
Current	A	Ampere	MA (milliamp)
Frequency	Hz	Hertz	KHZ (kilohertz)
Time	s	second	MS (millisecond)
	Common Multipliers		
	1E3	K	kilo
	1E-3	M	milli
	1E-6	U	micro

Table 2-1: Command parameters Suffixes and Multipliers

2.6.2 Character Data

Character strings returned by query statements may take either of the following forms, depending on the length of the returned string:

<CRD> Character Response Data. Permits the return of character strings.

<AARD> Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.

<SRD> String Response Data. Returns string parameters enclosed in double quotes.

3. System Considerations

This chapter addresses some system issues concerning setting up interfaces such as GPIB, USB or Ethernet.

3.1 IEEE-488 / GPIB Interface

All CS Series power sources are equipped with an industry standard IEEE-488.2 interface (GPIB).

A GPIB controller such as a Windows PC with suitable GPIB controller card is required to use the GPIB interface.

3.1.1 Assigning the IEEE-488 Address

The AC source address cannot be set remotely. It must be set from the front panel. Once the address is set, you can assign it inside programs. The GPIB address can be set/changed from the CONFIGURATION menu screen. Press the MENU key and scroll to the CONFIGURATION menu using the Up/Down arrow keys or press the MENU key repeatedly until the CONFIGURATION screen appears. Press the ENTER key to enter the CONFIGURATION screen.

Scroll to the ADDRESS field using the Up/Down arrow keys on the front panel. The value of the ADDRESS can be set from 0 through 31. Avoid using address 0 as it is generally reserved for the GPIB bus controller. Once set, the GPIB address of the power source is retained in non-volatile memory.

For systems using the National Instruments VISA or IVI drivers, the address is specified in the resource descriptor (GPIB::1). Consult you programmer’s reference documentation on how to address a GPIB instrument using your specific GPIB controller’s function library.

3.1.2 CSGui

The provided Windows CSGui program supports the GPIB interface on CS Series models but only in combination with a National Instruments GPIB controller. The default controller ID is zero but controller ID’s from 0 thorugh 3 can be selected in the CSGui Interface screen if multiple GPIB controllers are present in the same PC. Note that the CSGui program only supports one CS power source at a time.

3.2 USB Interface

Unlike RS232C, there are no generic drivers available as a rule for us in programming environments such as LabView, LabWindows/CVI or Visual Basic. However, support for USB is included under VISA and may be used to interface to the power source using the USB interface.

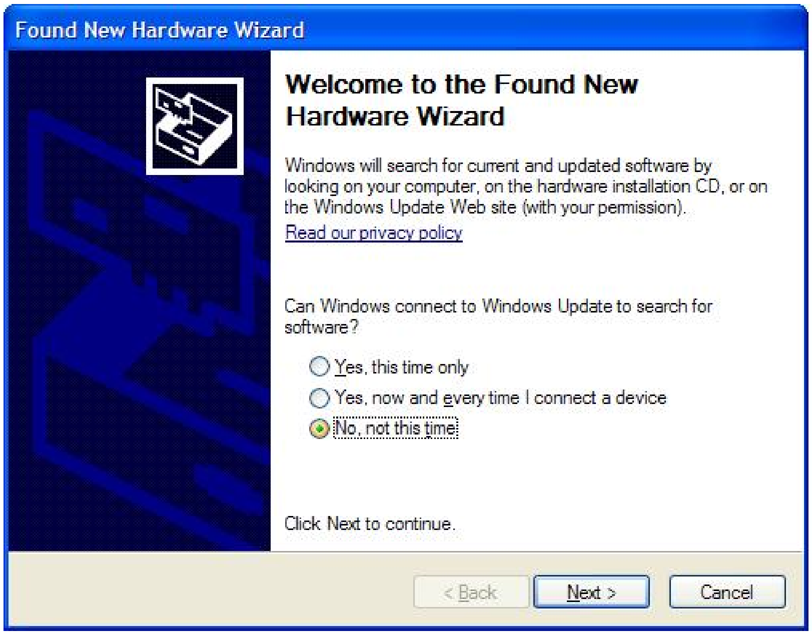
For other environments, a virtual serial port utility is provided on CD ROM CIC496 which ships with the CS Series power source. This utility will provide a virtual COM port on a PC under WindowsXP. This allows legacy programs to use the USB port port a thought is a regular serial port on the PC. The baud rate for this mode of operation is fixed at 460,800. If you plan to use this feature, the USB-Serial Adaptor installation must be run to install the virtual com port driver. This option is only supported under WindowsXP at this time.

3.2.1 USB Driver Installation

When connecting the AC source through the USB interface to Windows XP PC, the presence of a new USB device will be detected. Windows will display a dialog after a short delay prompting the user to install the USB device drivers. There are two steps to this process.

The first one installs the USB decive itself. The second step allows installation of the USB to COM virtual port driver. This driver will allow access to the AC source USB interface using a virtual COM port. Many programming environments support RS232 access but not USB. The USB-to-COM virtual port driver is distributed on the CIC496 CD ROM.

Step 1: USB Device Driver installation



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

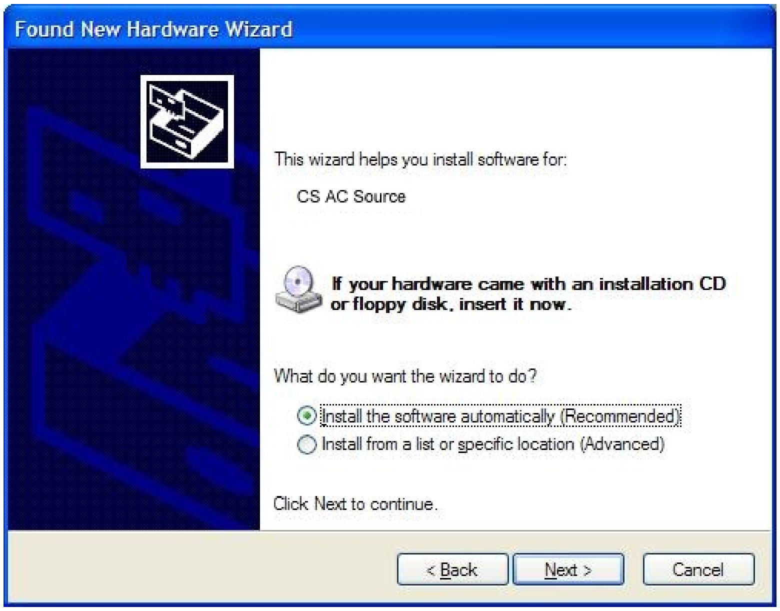
Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.

When the “Found New Hardware Wizard” dialog appears, select the “No, not this time.”option. The drivers are not available on line. Click on Next button to continue.



This wizard helps you install software for:

CS AC Source

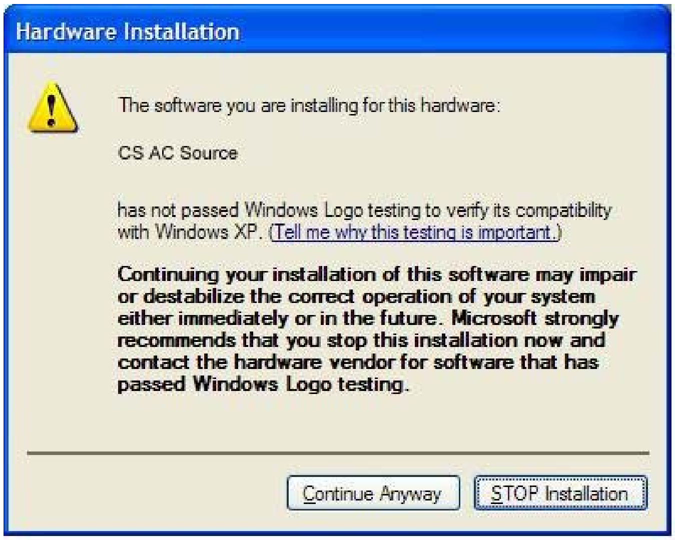
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)
○ Install from a list or specific location (Advanced)

Click Next to continue.

The next dialog will ask you to install the software for the CS AC Source. Select the “Install the software automatically (Recommended)” option and click on Next to continue.



The software you are installing for this hardware:

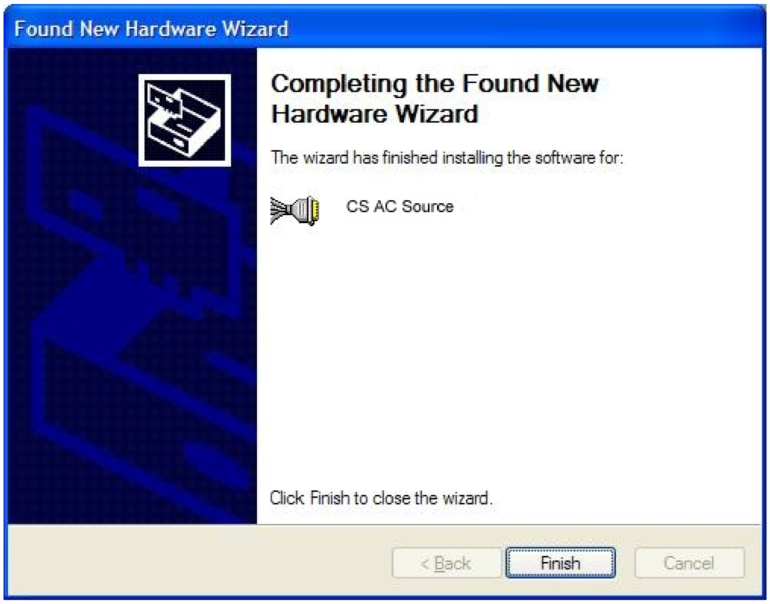
CS AC Source

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

The USB device drivers have not been WindowsXP Logo certified. Due to the limited distribution of these drivers, this is unlikely to be done. This Logo certification has no bearing on the functionality or legitimacy of this device driver so you can ignore this message. Click the “Continue Anyway” button to continue.

The installation will now proceed. This process may take several minutes to complete.



Completing the Found New Hardware Wizard

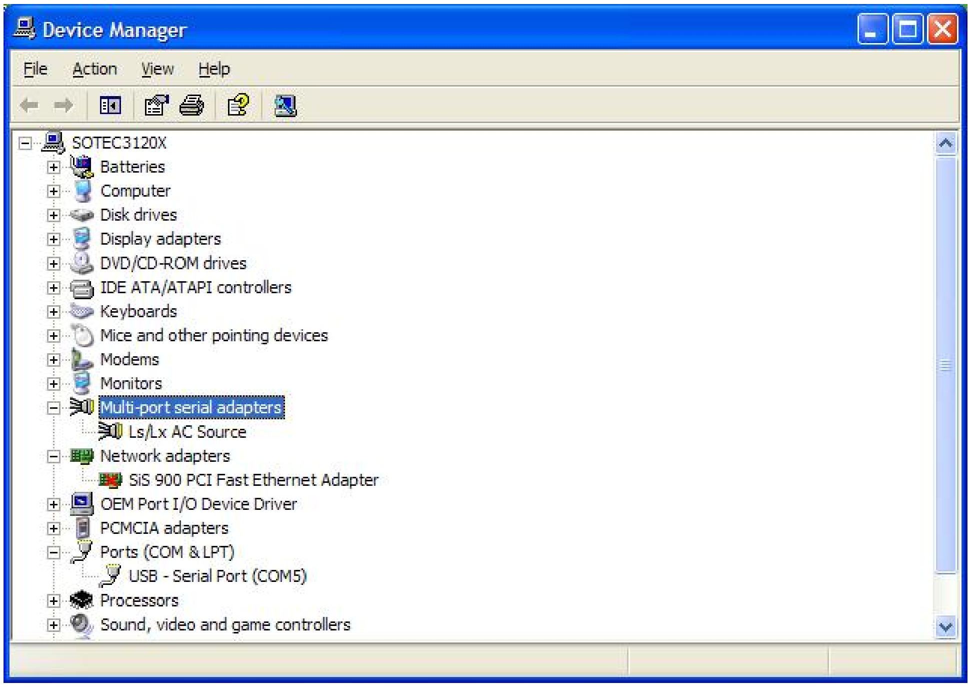
The wizard has finished installing the software for:

CS AC Source

Click Finish to close the wizard.

Once completed, the dialog box shown above will appear signaling the device drivers have been installed. The USB interface is now available to the PC's operating system. To complete the install process, click on the "Finish" button.

To verify the USB port is available, you can access the Windows System Properties screen, select the Hardware tab and open the Windows Device Manager screen. The CS Source should be listed under "Multi-port serial adapters" as shown in the image below.



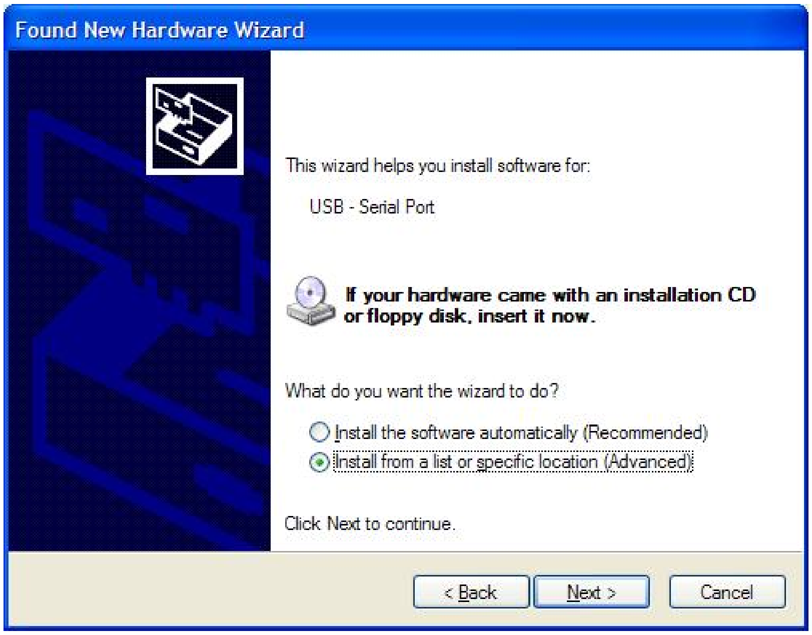
```
SOTEC3120X
  Batteries
  Computer
  Disk drives
  Display adapters
  DVD/CD-ROM drives
  IDE ATA/ATAPI controllers
  Keyboards
  Mice and other pointing devices
  Modems
  Monitors
  Multi-port serial adapters
    Ls/Lx AC Source
  Network adapters
    SiS 900 PCI Fast Ethernet Adapter
  OEM Port I/O Device Driver
  PCMCIA adapters
  Ports (COM & LPT)
    USB - Serial Port (COM5)
  Processors
  Sound, video and game controllers
```

Figure 3-1: Windows XP Device Manager - USB Port

Step 2: USB to Com Virtual Device Driver installation

The second step allows installation of the USB to COM virtual port driver. This driver will allow access to the AC source USB interface using a virtual COM port. Many programming environments support RS232 access but not USB. The use of this driver will allow you to program the CS Source through the USB port as though it was an RS232 port. The USB-to-COM virtual port driver is distributed on the CIC496 CD ROM. This second step is optional but is required to use the included CSGui Windows software however.

To skip this phase of the installation, press the Cancel button. To continue the installation, make sure the CIC496 CD Rom is available. Insert in the CD ROM drive if needed. If the auto-run screen appears, you can close it.


Found New Hardware Wizard

This wizard helps you install software for:

USB - Serial Port

If your hardware came with an installation CD or floppy disk, insert it now.

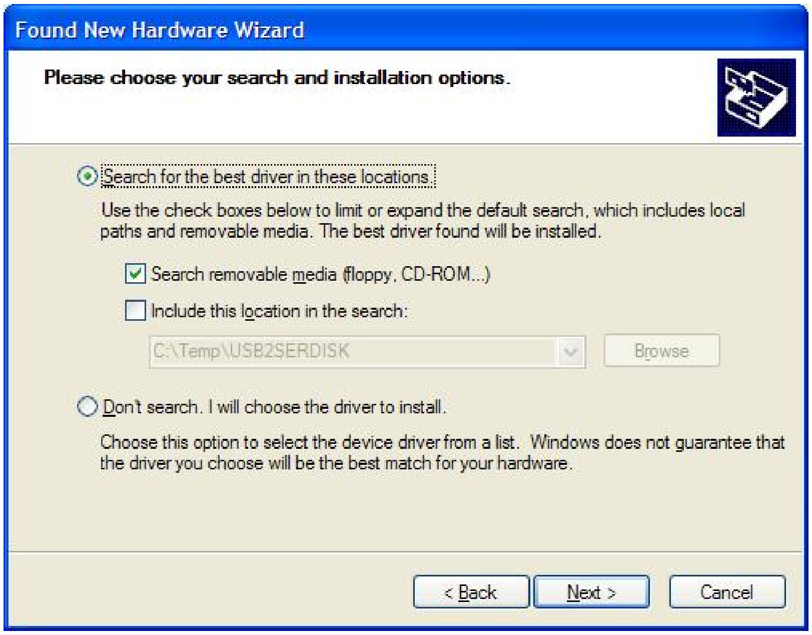
What do you want the wizard to do?

○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.

Next, select the “Install from a list or specific location (Advanced)” option and click on the Next button to continue.


Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

☑ Search removable media (floppy, CD-ROM...)

□ Include this location in the search:

C:\Temp\USB2SERDISK

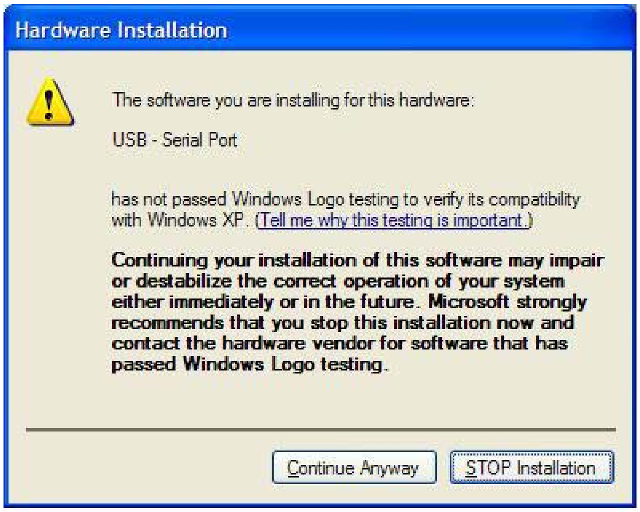
○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

When prompted for the location of the driver software, select the “Search removable media (floppy, CD)” option. Click on the Next button to continue.

The installation wizard should be able to locate virtual com driver at in sub directory “\USB_Inf” on the CIC496 CD Rom. If not, you will have to browse to this directory on the CD ROM when prompted for a specific location.

If the INF file is found, the installation will proceed automatically.



Hardware Installation

The software you are installing for this hardware:

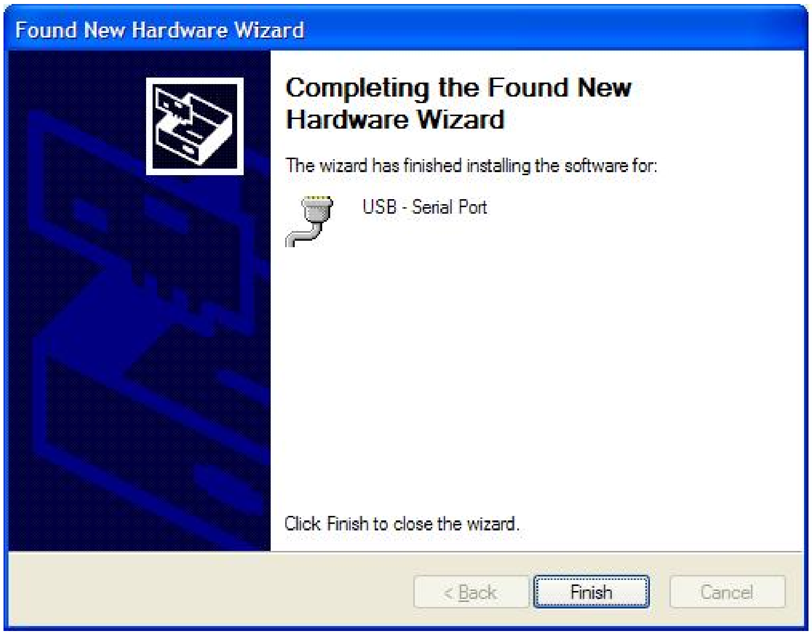
USB - Serial Port

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

The USB to Com virtual device drivers have not been WindowsXP Logo certified. Due to the limited distribution of these drivers, this is unlikely to be done. This Logo certification has no bearing on the functionality or legitimacy of this device driver so you can ignore this message. Click the “Continue Anyway” button to continue.

The installation will now proceed. This process may take several minutes to complete. Once completed, the final dialog will appear as shown.



Found New Hardware Wizard

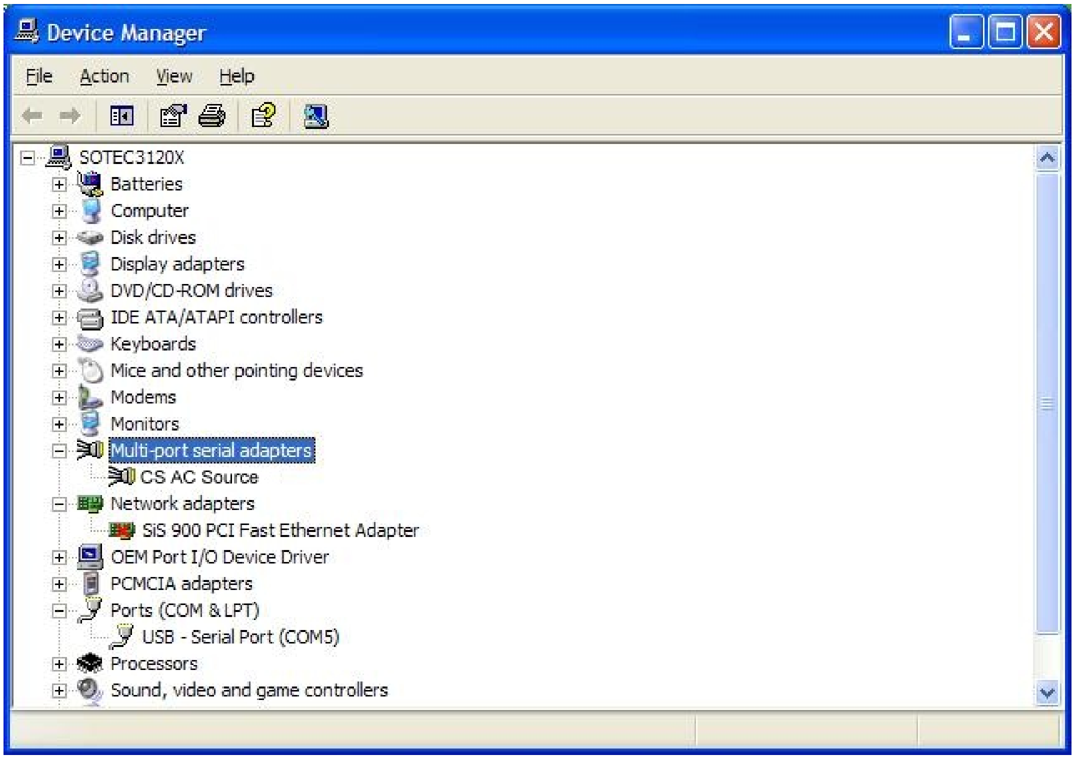
Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

USB - Serial Port

Click Finish to close the wizard.

The USB to Com virtual port is now available to the PC's operating system. To complete the install process, click on the "Finish" button. To verify the virtual com port is available, you can access the Windows System Properties screen, select the Hardware tab and open the Windows Device Manager screen. The CS Source should be listed under "Ports (COM & LPT)" as shown in the image below. The com port number is automatically assigned. Note the com port number for subsequent reference in your application software or when selected the COM port in the CSGui Interface screen.



```
SOTEC3120X
    Batteries
    Computer
    Disk drives
    Display adapters
    DVD/CD-ROM drives
    IDE ATA/ATAPI controllers
    Keyboards
    Mice and other pointing devices
    Modems
    Monitors
    Multi-port serial adapters
        CS AC Source
    Network adapters
        SiS 900 PCI Fast Ethernet Adapter
    OEM Port I/O Device Driver
    PCMCIA adapters
    Ports (COM & LPT)
        USB - Serial Port (COM5)
    Processors
    Sound, video and game controllers
```

Figure 3-2: Windows XP Device Manager – Virtual Com Port

Once completed, you can remove the CIC496 CD Rom. The USB interface to the AC source is now available for use.

3.2.2 USB Interface Use

Note that the power source will be detected automatically when turn on or plugged in once the drivers have been installed. It is recommended however to close any open USB connections to the AC source before turning it off.

To use the USB interface, you may use the CSGui Windows software supplied with the power source or develop your own application code. In either case, set the baud rate on the power source to 460,800 in the Configuration menu. From CS Front panel, press MENU key, scroll to CONFIGURATION and press ENTER key. Select BAUDRATE field and scroll to 460800.

3.3 -LAN Option

An Ethernet LAN interface option is available for the CS Series power sources. This option must be specified at the time of order. A –LAN option indicator will appear on the model number tag at the rear-panel of the power source to indicate the presence of this option. Also, a RJ45 socket will be present on the rear panel.

Using LAN lets you communicate with the instrument remotely, it is fast, simple and the LAN from your PC does not require any additional proprietary software or cards.

Note: If a USB cable is plugged into the USB interface connector of the power source, the LAN interface will be disabled. Remove any USB connection to use the LAN / Ethernet port.

3.3.1 MAC Address

Each power source with the –LAN option installed has a unique network address (MAC address). The MAC address (Media Access Conrol) is a unique hexadecimal address and is listed on a label on the rear panel of the power source. To operate the power source on a network, this MAC address needs to be assigned to a TCP/IP address which will be used to address the device on the network.

3.3.2 TCP/IP Address

The first decision you need to make is how to connect the instrument. You can connect the instrument directly to a network LAN port with a LAN cable, or you can connect it directly to the PC. When connecting the instrument directly to the PC LAN port you will need a special cable called a cross connect cable. Once connected you must establish an IP address for the instrument. An IP address consists of four groups of numbers separated by a decimal. Dynamic Host Configuration Protocol (DHCP) is typically the easiest way to configure the instrument for LAN communication. DHCP automatically assigns a dynamic IP address to a device on a network. If the power source is to be connected to a corporate network, consult your network administrator to obtain an IP address and IP Mask. You will need to enter this information on the computer used to control the power source to establish a network connection.

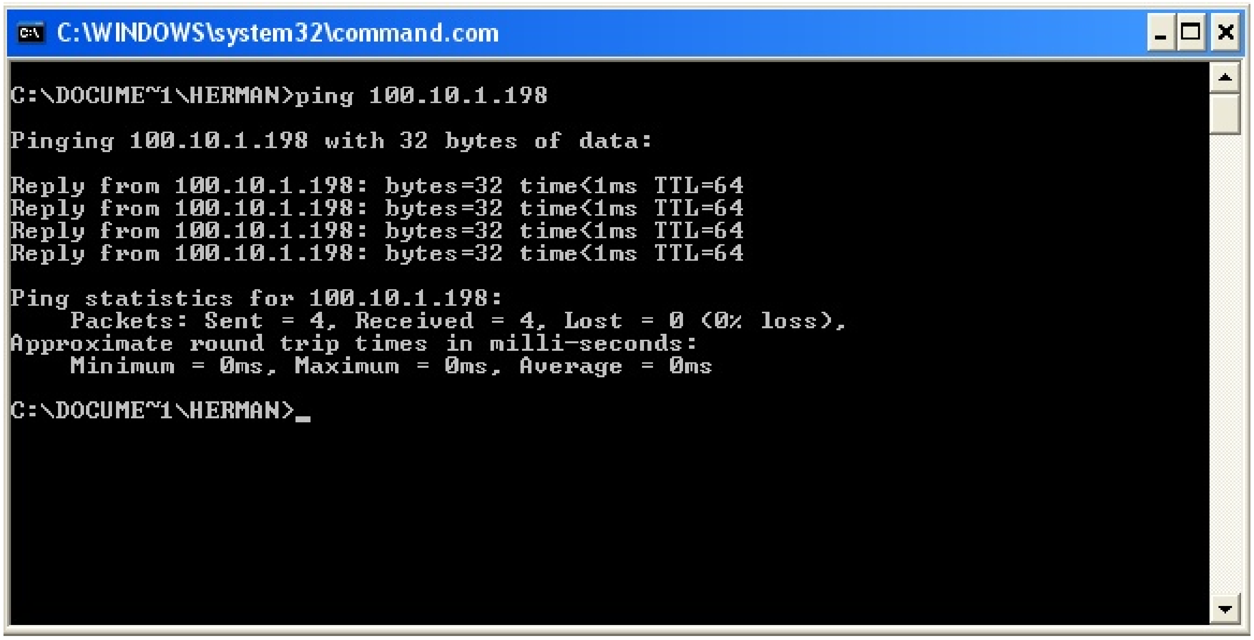
Once you have an IP address, you can test the IP address from your Windows PC. An easy way to do so is to use the ping utility under MS DOS. To do so, bring up a DOS window using the start menu:

Start>Programs>Accessories>Command Prompt)

At the command prompt type

ping <IP address>.

This will send an IP ping request to the power source. For this to work, the power source must be turned on and connected to the same network as the PC. Also, the power source interface configuration must be set to use a baud rate of 460,800. If everything is working it will look like this:



```
C:\DOCUME~1\HERMAN>ping 100.10.1.198

Pinging 100.10.1.198 with 32 bytes of data:

Reply from 100.10.1.198: bytes=32 time<1ms TTL=64
Reply from 100.10.1.198: bytes=32 time<1ms TTL=64
Reply from 100.10.1.198: bytes=32 time<1ms TTL=64
Reply from 100.10.1.198: bytes=32 time<1ms TTL=64

Ping statistics for 100.10.1.198:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\DOCUME~1\HERMAN>_
```

Figure 3-3: Pinging AC Source LAN IP address.

3.3.3 Socket Port Number

Now that a connection has been verified, you can develop your application code. If you are using one of the Microsoft environments, the Winsock protocol which is part of the Windows operating system can be used. Similar capabilities are supported on other operating systems.

To use Winsock, your will have to specificy the port number of the power source’s LAN interface. The port number determines the protocol for the communication. The CS power source uses ASCII characters and instrument SCPI commands for remote control. The IANA registered Port number for the Instrument SCPI interface is 5025.

TCP Remote port = 5025

3.4 RS232C Serial Interface

Note: If a USB cable is plugged into the USB interface connector of the power source, the RS232 interface will be disabled. Remove any USB connection to use the serial port.

The RS232C interface is factory enabled for the CS Series. The RS232C cable required to connect the CS Series AC Current Source to a PC serial port is a standard 9 pin Male to 9 pin Female straight-thorugh serial cable. A suitable 6 feet long RS232C cable is supplied with each power soruce. (CI P/N 250709). Replacement cables are available through customer service (support@calinst.com)

3.5 Instrument Drivers and Application Software

Instrument drivers for National Instruments LabWindows/CVI and LabView are generally available for download from the California Instruments' web site at www.calinst.com. Also available are ready to use interactive graphical user interface (GUI) programs for download.

4. SCPI Command Reference

4.1 Introduction

This chapter provides a complete listing of all SCPI commands supported by the CS Series Series of AC sources. Commands are grouped by function according the root level commands. Some general command related issues are:

Phases

If a command can apply to individual phases of an AC source, the entry "*Phase Selectable"* will appear in the command description.

Related Commands

Where appropriate, related commands or queries are included. These are listed because they are either directly related by function, or because reading about them will clarify or enhance your understanding of the original command or query.

This chapter is organized as follows:

- Subsystem commands, arranged by subsystem
- IEEE 488.2 common commands

4.2 Subsystem Commands

Subsystem commands are specific to AC source functions. They can be a single command or a group of commands. The groups are comprised of commands that extend one or more levels below the root. The description of common commands follows the description of the subsystem commands.

The subsystem command groups are listed in alphabetical order and the commands within each subsystem are grouped alphabetically under the subsystem. Commands followed by a question mark (?) take only the query form. When commands take both the command and query form, this is noted in the syntax descriptions.

You will find the subsystem command groups discussed on the following pages:

4.3 Calibration Subsystem

The commands in this subsystem allow you to do the following:

- Enter the calibration password
- Calibrate the current and voltage output levels, and store new calibration constants in nonvolatile memory.

Subsystem Syntax

```
CALibrate
    :MEASure
        :CURRent                Begin current measurement calibration sequence
            :SPHase             Query format returns single phase current mea coefficient.
        :VOLTage                Begin current measurement calibration sequence
    :PASSword                   Unlock calibration
    :SAVE                       Save new cal constants in non-volatile memory
    [:SOURce]
        PHASe                   Phase offset calibration
        :CURRent                Query output current cal coefficient
```

CALibrate:MEASure:CURRent

Phase Selectable

This command can only be used in calibration mode. It initiates the calibration of the AC current metering circuits. The query format returns the actual calibration coefficient. Use the INST:SEL or INST:NSEL to select the desired phase.

Command Syntax	CALibrate:MEASure:CURRent
Parameters	None
Query Syntax	CALibrate:MEASure:CURRent?
Returned Parameters	<NR3>
Examples	CAL:MEAS:CURR
Related Commands	CAL:SAVE CAL:MEAS:VOLT

CALibrate:MEASure:CURRent:SPHase

This command can only be used in calibration mode and in query form. It applies only to phase A. It returns the single-phase mode current measurement calibration coefficient. If the CS is in single phase mode, this command is equivalent to the the CAL:MEAS:CURR? query for phase 1/A except it can not be used to perform a calibration. It does not initiate a calibration. Use the CAL:MEAS:CURR? command to do this instead.

Query Syntax	CALibrate:MEASure:CURRent?
Returned Parameters	<NR3>
Examples	CAL:MEAS:CURR:SPH?
Related Commands	CAL:SAVE CAL:MEAS:VOLT

CALibrate:MEASure:VOLTage

Phase Selectable

This command can only be used in calibration mode. It initiates the calibration of the AC voltage metering circuits. The query format returns the actual calibration coefficient. Use the INST:SEL or INST:NSEL to select the desired phase.

Command Syntax	CALibrate:MEASure:VOLTage	
Parameters	None	
Query Syntax	CALibrate:MEASure:CURRent?	
Returned Parameters	<NR3>	
Examples	CAL:MEAS:VOLT	
Related Commands	CAL:SAVE	CAL:MEAS:CURR

CALibrate:PASSword

This command can only be used to unlock the calibration mode. Once unlocked, non-query calibration commands will be accepted. Query commands are always accepted.

Command Syntax	CALibrate:PASSword<NRf>
Parameters	<high voltage range> (default)
Examples	CAL:PASS 135
Related Commands	none

CALibrate:SAVE

This command can only be used in calibration mode. It saves any new calibration constants (after a current or voltage calibration procedure has been completed) in nonvolatile memory.

Command Syntax:	CALibrate:SAVE	
Parameters	None	
Examples	CAL:SAVE	
Related Commands	CAL:CURR	CAL:VOLT

CALibrate[:SOURce]:PHASe

Phase Selectable

This command can be used to set the phase calibration coefficient. Use the INST:SEL or INST:NSEL to select the desired phase. This allows the phase for voltage 2 and 3 (B and C) to be adjusted with respect to phase A. The query format returns the actual calibration coefficient

Command Syntax	CALibrate[:SOURce]:PHASe
Parameters	<NRf+>
Query Syntax	CALibrate[:SOURce]:PHASe?
Returned Parameters	<NR3>
Examples	CAL:PHAS 1.3
Related Commands	none

CALibrate[:SOURce]:CURRent

Phase Selectable

This command can be used to query the output current calibration coefficient. Only the query format is available. Use the INST:SEL or INST:NSEL to select the desired phase.

Command Syntax	CALibrate[:SOURce]:CURRent
Parameters	None
Query Syntax	CALibrate[:SOURce]:CURRent?
Returned Parameters	<NR3>
Examples	CAL:CURR?
Related Commands	none

4.4 Diagnostic Subsystem

These subsystem commands perform diagnostic functions which include reading and writing to the EEPROM, resetting the AC source and reading temperature.

Subsystem Syntax

```
DIAGnostic
    :RESet                                      Force power-on reset
    :TEMPerature
        :AMBient?                               Returns ambient temperature in °C
```

DIAGnostic:RESet

This commands forces a power-on reset.

Command Syntax	DIAGnostic:RESet
Parameters	None
Examples	DIAG:RES
Related Commands	*RST

DIAGnostic:TEMPerature:AMBient?

This query returns the temperature measured at the ambient sense thermistor in degrees C.

Query Syntax	DIAGnostic:TEMPerature:AMBient?
Parameters	None
Examples	DIAG:TEMP:AMB?
Returned Parameters	<NR3>

4.5 Instrument Subsystem

This subsystem programs the three-phase output capability of the AC Power Source.

Subsystem Syntax

INSTrument	
:COUPle ALL | NONE	Couple all phases for programming
:NSELect <n>	Select the output phase to program (1|2|3)
:SELect <output>	Select the output phase to program (OUTP1|OUTP2|OUTP3)

INSTrument:COUPle

In a three-phase power source it is convenient to set parameters of all three output phases simultaneously with one programming command. When INST:COUP ALL is programmed, sending a command to any phase will result in that command being sent to all three phases.

INSTrument:COUPle only affects the operation of subsequent commands. It does not by itself immediately affect the AC source's output. The commands that are affected by INSTrument:COUPle are those with the designation: Phase Selectable.

INSTrument:COUPle has no affect on queries. There is no way to query more than one phase with a single command. Directing queries to individual phases is done with INSTrument:NSELect.

Command Syntax	INSTrument:COUPle<coupling>
Parameters	ALL|NONE
*RST Value	ALL
Examples	INST:COUP ALL
Query Syntax	INSTrument:COUPle?
Returned Parameters	<CRD>
Related Commands	INST:NSEL

INSTrument:NSELect
INSTrument:SELect

These commands allow the selection of individual outputs in a three-phase model for subsequent commands or queries. Their operation is dependent on the setting of INSTrument:COUPle. If INST:COUP NONE is programmed, then the phase selectable commands are sent only to the particular output phase set by INSTrument:NSELect. If INST:COUP ALL is programmed, then all commands are sent to all three output phases.

INSTrument:NSELect selects the phase by its number, while INSTrument:SELect references it by name. These commands also select which output phase returns data when a query is sent.

Command Syntax	INSTrument:NSELect <NR1> INSTrument:SELect <output>
Parameters	For INST:NSEL: 1 | 2 | 3 For INST:SEL: OUTPut1 | OUTPut2 | OUTPut3
*RST Value	1 or OUTPut1
Examples	INST:NSEL 3
Query Syntax	INSTrument:NSELect?
Returned Parameters	<NR1>
Related Commands	INST:COUP

4.6 Limit Subsystem

These subsystem commands may be used to query the hardware limits (capabilities) of the AC power source. These commands are protected and can only be used in query format.

Subsystem Syntax

```
LIMit
    :CURRent                                Current limit setting
    :FREQuency
        :HIGH                               Frequency limit high
        :LOW                                Frequency limit low
    :PHASe                                  Phase mode
    :VOLTage
        :HIGH                               Voltage limit high voltage range
        :LOW                                Voltage limit low voltage range
```

LIMit:CURRent

Query form returns the configuration current limit. This value determines the maximum current available from one amplifier. Note that this is not the same as the available maximum current, which is a function of the number of CS units for multibox systems and phase mode. To query the maximum available RMS current, use the CURR? MAX command.

Query Syntax	LIMit:CURRent?
Returned Parameters	<NR3>
Examples	LIM:CURR?
Related Commands	CURR

LIMit:FREQuency:HIGH

Query form returns the maximum available output frequency. This value determines the maximum frequency available using a sinusoidal waveform. Note that this is not the same as the available maximum frequency, which is a function of the frequency harmonic content of the waveform. To query the maximum available frequency, use the FREQ? MAX command.

Query Syntax	LIMit:FREQuency:HIGH?
Returned Parameters	<NR3>
Examples	LIM:FREQ:HIGH?
Related Commands	LIM:FREQ:LOW?

LIMit:FREQuency:LOW

Query form returns the maximum available output frequency. This value determines the minimum frequency available.

Query Syntax	LIMit:FREQuency:LOW?
Returned Parameters	<NR3>
Examples	LIM:FREQ:LOW?
Related Commands	LIM:FREQ:HIGH?

LIMit:PHASe

Query form returns the phase configuration setting. This value determines the phase mode of operation according to the table below:

Setting	Operation
0	Single-phase mode.

120	Three phase mode. Determines relative phase angle between phases A, B and C (ø1, ø2 and ø3). If the MODE field is set (standard on CS Series) the AC source can operate in both 1 and 3 phase modes.
Other	Any value other than 0 or 120 indicates 2 phase configuration with phase angle between A and B set to value shown.

Query Syntax	LIMit:PHASe?
Returned Parameters	<NR3>
Examples	LIM:PHAS?
Related Commands	SYST:CONF:NOUT?

LIMit:VOLTage:HIGH

Query form returns the maximum available output voltage. This value determines the maximum AC RMS compliance voltage available using a sinusoidal waveform. Note that this is not the same as the available maximum voltage, which is a function crest factor of the voltage waveform. To query the maximum available voltage, use the VOLT? MAX command. On the CS Series, the HIGH and LOW voltage limits are set to the same value.

Query Syntax	LIMit:VOLTage:HIGH?
Returned Parameters	<NR3>
Examples	LIM:VOLT:HIGH?
Related Commands	LIM:VOLT:LOW?

LIMit:VOLTage:LOW

Query form returns the maximum available output voltage. This value determines the maximum AC RMS compliance voltage available using a sinusoidal waveform. Note that this is not the same as the available maximum voltage, which is a function crest factor of the voltage waveform. To query the maximum available voltage, use the VOLT? MAX command. On the CS Series, the HIGH and LOW voltage limits are set to the same value.

Query Syntax	LIMit:VOLTage:LOW?
Returned Parameters	<NR3>
Examples	LIM:VOLT:LOW?
Related Commands	LIM:VOLT:HIGH?

4.7 Array Measurement Subsystem

This subsystem lets you retrieve arrays containing measurements data. Only current and voltage measurements are stored in an array. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new data before returning the readings from the array. FETCh returns previously acquired data from the array.

Individual outputs of a three-phase source are specified by the setting of INSTrument:NSELect.

Subsystem Syntax

```
MEASure | FETCh
    :ARRay
        :CURRent
        [:DC]?                  Returns the digitized instantaneous current
        :HARMonic
            [:AMPLitude]?       Returns amplitudes of the first 50 harmonics
            :PHASe?             Returns phase angles of the first 50 harmonics
        :MODE                   Selects waveform data transfer format
        :NEUTral
            [:DC]?              Returns the neutral digitized instantaneous current (3-
                                phase only)
            :HARMonic
                [:AMPLitude]?   Returns neutral current harmonic amplitude
                :PHASe?         Returns neutral current harmonic phase
        :VOLTage
            [:DC]?              Returns the digitized instantaneous voltage
            :HARMonic
                [:AMPLitude]?   Returns amplitudes of the first 50 harmonics
                :PHASe?         Returns phase angles of the first 50 harmonics
```

MEASure:ARRay:CURRent?
FETCh:ARRay:CURRent?

Phase Selectable

These queries return an array containing the instantaneous output current in amperes. The data returned in arbitrary block data format as follows:

#5<block length n><b0><b1><b2><b3>.....<bn-3><bn -2><bn-1><bn>

where b0,b1,b2,b3 are four hex bytes represent IEEE single precision floating number, where b0 is the most significant byte and b3 is the least significant byte. The number of bytes returned is contained in the data block header which always starts with the "#" pound character followed by a single decimal character indicating the number of digits that make up the block length of the data. Thus, "#516384…" indicates that there are 5 digits that follow containing the number of bytes in the data block (excluding the header and length information). The actual number of bytes in this case is 16384 or 16Kbytes.

The output voltage and current are digitized whenever a measure command is given or whenever an acquisition trigger occurs. The acquisition sampling time interval is set by SENSe:SWEep:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.

This command has two optional parameters. The first one may be used to specify the number of 256 data sample blocks to transfer. Valid parameter values are from 1 through 16. The second parameter may be used to specify the offset in number of 256 data sample blocks from which to start the data transfer. Valid offset values are from 0 thorugh 15. If both parameters are omitted, all 16 blocks are transferred starting from offset 0 (first block).

Query Syntax	MEASure:ARRay:CURRent[:DC]? [<n>,<n>] FETCh:ARRay:CURRent[:DC]? [<n>,<n>]
Parameters	Optional block and offset parameters <n>,<n>. Where the first value <n> is the number of 256 sample blocks to transfer and the second value <n> is the first block (offset) to start with. Number of blocks is from 1 to 16, offset is from 0 to 15.
Examples	MEAS:ARR:CURR? FETC:ARR:CURR? 4,0
Returned Parameters	4096 NR3 values
Related Commands	INST:NSEL SENS:SWE

MEASure:ARRay:CURRent:HARMonic?
FETCh:ARRay:CURRent:HARMonic?

Phase Selectable

These queries return an array of harmonic amplitudes of output current in rms amperes. The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure:ARRay:CURRent:HARMonic[:AMPLitude]? FETCh:ARRay:CURRent:HARMonic[:AMPLitude]?
Parameters	None
Examples	MEAS:ARR:CURR:HARM? FETC:ARR:CURR:HARM?
Returned Parameters	51 NR3 values
Related Commands	INST:NSEL

MEASure:ARRay:CURRent:HARMonic:PHASe?
FETCh:ARRay:CURRent:HARMonic:PHASe?

Phase Selectable

These queries return an array of harmonic phases of output current in degrees, referenced to the positive zero crossing of the fundamental component. The first value returned is the dc component (always returned as 0 degrees phase), the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure:ARRay:CURRent:HARMonic:PHASe?<NRf> FETCh:ARRay:CURRent:HARMonic:PHASe?<NRf>
Parameters	None
Examples	MEAS:ARR:CURR:HARM:PHAS? FETC:ARR:CURR:HARM:PHAS?
Returned Parameters	51 NR3 values
Related Commands	INST:NSEL

MEASure:ARRay:CURRent:NEUTral?
FETCh:ARRay:CURRent:NEUTral?

These queries return an array containing the instantaneous output current of the neutral output terminal in amperes. The output voltage and current are digitized whenever a measure command is given or whenever an acquisition trigger occurs. The acquisition sampling time interval is set by

SENSe:SWEep:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.

Query Syntax	MEASure:ARRay:CURRent:NEUTral[:DC]? FETCh:ARRay:CURRent:NEUTral[:DC]?
Parameters	None
Examples	MEAS:ARR:CURR:NEUT? FETC:ARR:CURR:NEUT?
Returned Parameters	4096 NR3 values
Related Commands	INST:NSEL SENS:SWE

MEASure:ARRay:CURRent:NEUTral:HARMonic?
FETCh:ARRay:CURRent:NEUTral:HARMonic?

These queries return an array of harmonic amplitudes of output current of the neutral output terminal in rms amperes.

The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure:ARRay:CURRent:NEUTral:HARMonic[:AMPLitude]? FETCh:ARRay:CURRent:NEUTral:HARMonic[:AMPLitude]?
Parameters	None
Examples	MEAS:ARR:CURR:NEUT:HARM? FETC:ARR:CURR:NEUT:HARM?
Returned Parameters	51 NR3 values
Related Commands	INST:NSEL

MEASure:ARRay:CURRent:NEUTral:HARMonic:PHASe?
FETCh:ARRay:CURRent:NEUTral:HARMonic:PHASe?

These queries return an array of harmonic phases of output current of the neutral output terminal in degrees, referenced to the positive zero crossing of the fundamental component. The first value returned is the dc component (always returned as 0 degrees phase); the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure:ARRay:CURRent:NEUTral:HARMonic:PHASe? FETCh:ARRay:CURRent:NEUTral:HARMonic:PHASe?
Parameters	None
Example	MEAS:ARR:CURR:NEUT:HARM:PHAS? FETC:ARR:CURR:NEUT:HARM:PHAS?
Returned Parameters	51 NR3 values
Related Commands	INST:NSEL

MEASure:ARRay:MODe

This command selects the waveform array data format to be used. The default mode is binary (BIN) which uses an IEEE floating point data format in which each data sample is transferred as a 4 byte floating point binary data word. Alternatively, an ASCII format may be selected (ASCii) in which each data sample is sent as 8 ASCII Hex values representing the 4 byte IEEE floating point data. Note that the transfer mode only applies to MEAS:ARR:VOLT and MEAS:ARR:CURR queries. All other measurement queries always return ASCII data. Note that at power on, the default mode is always set to binary (BIN).

Syntax	MEASure:ARRay:MODe	
	Parameters	BIN | ASCii
Examples	MEAS:ARR:MOD ASC	
Related Commands	MEAS:ARR:VOLT	MEAS:ARR:CURR

Note: The MEAS:ARR:MOD command is provided to allow waveform data transfers in ASCII on DBCS versions of MS Windows. Examples of DBCS versions are Chinese, Korean, Japanese etc. On most Windows versions, the binary mode can be used as it reduces the amount of data transferred and thus provides better throughput.

The ASCII mode will double the number of characters transferred so provisions for a larger receive buffer on the PC may have to be made. On the Lx/Ls, the full acquisition data size that can be sent with one command in BIN mode is 16KB, in ASC mode 32KB.

The binary data must be converted to a single precision floating point notation. Sample VB6 code is shown on the next page.

Conversion function sample VB6. Converting waveform data from either transfer mode to a single precision value can be accomplished using the following sample routine:

```
Public Function StringToIEEEFloat(ByVal sData As String, ByVal bAsciiMode As Boolean) As
Single
'=============================================================
'bAsciiMode flag is used if data is received as 8 ascii chars
'representing Hex 0-9,A-F. If bAsciiMode flag is false, then
'data is process as 4 char representing a byte each. Ascii
'mode is needed for DCBS windows
'=============================================================
  Dim i         As Integer
  Dim j         As Integer
  Dim iChar     As Integer
  Dim expo      As Long
  Dim mantisse  As Long
  Dim expo_val  As Variant
  Dim mant_f    As Single
  Dim c(3)      As Long    'Must use 32 bit integers to allow for
                      'intermediate result of 24 bit shift
  Dim sign      As Boolean
'=============================================================
Const MANT_MAX = &H7FFFFF
Const EXPO_MAX = 2 ^ 126
'=============================================================

On Error GoTo FloatConvError
  If bAsciiMode Then
    'Retrieve ASC values from eight hex byte input data
    sData = UCase(sData)
    For i = 0 To 3
      c(i) = 0
      For j = 0 To 1
        iChar = AscB(Mid$(sData, i * 2 + j + 1, 1)) - 48
        If iChar > 9 Then iChar = iChar - 7
        c(i) = c(i) * 16 * j + iChar
      Next j
    Next i
  Else
    'Retrieve ASC values from four byte input data
    'Note: Don't use ASCB or ASCW functions as results will differ
    'based on character sets, even on non DCBS Windows
    'Retrieve ASC values from four byte input data
    For i = 0 To 3
      c(i) = Asc(Mid$(sData, i + 1, 1))
    Next i
  End If
  'Get sign bit
  sign = ((c(0) And &H80) = &H80)
  'Get exponent value less sign bit
  expo = (c(0) And &H7F) * 2
  'Pick up exponent sign
  If (c(1) And &H80) = &H80 Then expo = expo Or 1
  'get data less exponent sign bit
  c(1) = c(1) And &H7F
  mantisse = c(1) * &H10000 + c(2) * &H100 + c(3)
  mant_f = mantisse / MANT_MAX
  'Process exponent
  If (expo <> 0) And (expo <> &HFF) Then
    expo = expo - 127
    mant_f = mant_f + 1
    expo_val = 2 ^ Abs(expo)
    If (expo > 0) Then mant_f = mant_f * expo_val
    If (expo < 0) Then mant_f = mant_f / expo_val
  Else
    If (mant_f <> 0) Then
      If expo = 0 Then
        mant_f = mant_f / EXPO_MAX
      Else
        mant_f = mant_f * EXPO_MAX
      End If
    End If
  End If
  'Append number sign and return value
  If sign Then mant_f = -mant_f
  StringToIEEEFloat = mant_f
Exit Function
'==============================================================

FloatConvError:
  'Conversion errors are truncated to zero
  StringToIEEEFloat = 0
  Exit Function

End Function
```

MEASure:ARRay:VOLTage?
FETCh:ARRay:VOLTage?

Phase Selectable

These queries return an array containing the instantaneous output voltage in volts. The data returned in arbitrary block data format as follows:

#5<block length n><b0><b1><b2><b3>.....<bn-3><bn -2><bn-1><bn>

where b0,b1,b2,b3 are four hex bytes represent IEEE single precision floating number, where b0 is the most significant byte and b3 is the least significant byte. The number of bytes returned is contained in the data block header which always starts with the “#” pound character followed by a single decimal character indicating the number of digits that make up the block length of the data. Thus, “#516384…” indicates that there are 5 digits that follow containing the number of bytes in the data block (excluding the header and length information). The actual number of bytes in this case is 16384 or 16Kbytes.

The output voltage and current are digitized whenever a measure command is given or whenever an acquire trigger occurs. If digitization is caused by a measure command, the time interval between samples is determined by the output frequency. For frequencies greater than 45 Hz, the time interval is 10.4 microseconds. If digitization is caused by an acquire trigger, the time interval is set by SENSe:SWEep:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.

This command has two optional parameters. The first one may be used to specify the number of 256 data sample blocks to transfer. Valid parameter values are from 1 through 16. The second parameter may be used to specify the offset in number of 256 data sample blocks from which to start the data transfer. Valid offset values are from 0 thorugh 15. If both parameters are omitted, all 16 blocks are transferred starting from offset 0 (first block).

Query Syntax	MEASure:ARRay:VOLTage[:DC]? [<n>, <n>] FETCh:ARRay:VOLTage[:DC]? [<n>, <n>]
Parameters	Optional block and offset parameters <n>,<n>. Where the first value <n> is the number of 256 sample blocks to transfer and the second value <n> is the first block (offset) to start with. Number of blocks is from 1 to 16, offset is from 0 to 15.
Examples	MEAS:ARR:VOLT? FETC:ARR:VOLT?
Returned Parameters	4096 or less NR3 values
Related Commands	INST:NSEL SENS:SWE

MEASure:ARRay:VOLTage:HARMonic?
FETCh:ARRay:VOLTage:HARMonic?

Phase Selectable

These queries return an array of harmonic amplitudes of output voltage in rms volts. The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure:ARRay:VOLTage:HARMonic[:AMPLitude]? FETCh:ARRay:VOLTage:HARMonic[:AMPLitude]?
Parameters	None
Examples	MEAS:ARR:VOLT:HARM? FETC:ARR:VOLT:HARM?
Returned Parameters	51 NR3 values
Related Commands	INST:NSEL

4.8 Current Measurement Subsystem

This subsystem programs the current measurement capability of the CS Series. Two measurement commands are available: MEASure and FETCh.

MEASure triggers the acquisition of new measurement data before returning a reading.

FETCh returns a reading computed from previously acquired data.

Individual outputs of a three-phase source are specified by the setting of INSTrument:NSELect.

Subsystem Syntax

```
MEASure | FETCh
    [:SCALar]
    :CURRent
            [:DC]?                  Returns DC component of the current
            :AC?                    Returns AC rms current
            :ACDC?                  Returns AC+DC rms current
            :HARMonic
                [:AMPLitude]? <n>   Returns amplitude of the Nth harmonic of current
                :PHASe? <n>         Returns phase of the Nth harmonic of current
                :THD?               Returns % of total harmonic distortion of current
            :NEUTral
                [:DC]?              Returns neutral DC current (3-phase only)
                :AC?                Returns neutral AC rms current (3-phase only)
                :ACDC?              Returns neutral AC+DC rms current (3-phase only)
                :HARMonic
                    [:AMPLitude]? <n> Returns neutral current harmonic amplitude
                    :PHASe? <n>     Returns neutral current harmonic phase
    :THD:MODE RMSQ | FUND           Sets THD calculation to either RMS or Fundamental
                                    mode.
```

MEASure:CURRent?
FETCh:CURRent?

Phase Selectable

These queries return the DC component of the output current being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:CURRent[:DC]? FETCh[:SCALar]:CURRent[:DC]?
Parameters	None
Examples	MEAS:CURR? FETC:CURR?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:AC?
FETCh:CURRent:AC?

Phase Selectable

These queries return the AC component rms current being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:CURRent:AC? FETCh[:SCALar]:CURRent:AC?
Parameters	None
Examples	MEAS:CURR:AC? FETC:CURR:AC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:ACDC?
FETCh:CURRent:ACDC?

Phase Selectable

These queries return the AC and DC components of the rms current being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:CURRent:ACDC? FETCh[:SCALar]:CURRent:ACDC?
Parameters	None
Examples	MEAS:CURR:ACDC? FETC:CURR:ACDC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:HARMonic?
FETCh:CURRent:HARMonic?

Phase Selectable

These queries return the rms amplitude of the Nth harmonic of output current. The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:CURRent:HARMonic[:AMPLitude]?<NRf> FETCh[:SCALar]:CURRent:HARMonic[:AMPLitude]?<NRf>
Parameters	0 to 50
Examples	MEAS:CURR:HARM? 3 FETC:CURR:HARM? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:HARMonic:PHASe?
FETCh:CURRent:HARMonic:PHASe?

Phase Selectable

These queries return the phase angle of the Nth harmonic of output current, referenced to the positive zero crossing of the fundamental component. The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:CURRent:HARMonic:PHASe?<NRf> FETCh[:SCALar]:CURRent:HARMonic:PHASe?<NRf>
Parameters	0 to 50
Examples	MEAS:CURR:HARM:PHAS? 3 FETC:CURR:HARM:PHAS? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:HARMonic:THD?
FETCh:CURRent:HARMonic:THD?

Phase Selectable

These queries return the percentage of total harmonic distortion and noise in the output current.

Query Syntax	MEASure[:SCALar]:CURRent:HARMonic:THD? FETCh[:SCALar]:CURRent:HARMonic:THD?
Parameters	None
Examples	MEAS:CURR:HARM:THD? FETC:CURR:HARM:THD?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:NEUTral?
FETCh:CURRent:NEUTral?

These queries return the DC current in the neutral output terminal of a three-phase AC source.

Query Syntax	MEASure[:SCALar]:CURRent:NEUTral[:DC]? FETCh[:SCALar]:CURRent:NEUTral[:DC]?
Parameters	None
Examples	MEAS:CURR:NEUT? FETC:CURR:NEUT?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:NEUTral:AC?
FETCh:CURRent:NEUTral:AC?

These queries return the AC rms current in the neutral output terminal of a three-phase AC source.

Query Syntax	MEASure[:SCALar]:CURRent:NEUTral:AC? FETCh[:SCALar]:CURRent:NEUTral:AC?
Parameters	None
Examples	MEAS:CURR:NEUT:AC? FETC:CURR:NEUT:AC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:NEUTral:ACDC?
FETCh:CURRent:NEUTral:ACDC?

These queries return the AC+DC rms current in the neutral output terminal of a three-phase AC source.

Query Syntax	MEASure[:SCALar]:CURRent:NEUTral:ACDC? FETCh[:SCALar]:CURRent:NEUTral:ACDC?
Parameters	None
Examples	MEAS:CURR:NEUT:ACDC? FETC:CURR:NEUT:ACDC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:NEUTral:HARMonic?
FETCh:CURRent:NEUTral:HARMonic?

These queries return the rms amplitude of the Nth harmonic of current in the neutral output terminal of a three-phase AC source. The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:CURRent:NEUTral:HARMonic [:AMPLitude]?<NRf> FETCh[:SCALar]:CURRent:NEUTral:HARMonic [:AMPLitude]?<NRf>
Parameters	0 to 50
Examples	MEAS:CURR:NEUT:HARM? 3 FETC:CURR:NEUT:HARM? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:CURRent:NEUTral:HARMonic:PHASe?
FETCh:CURRent:NEUTral:HARMonic:PHASe?

These queries return the phase angle of the Nth harmonic of current in the neutral output terminal of a three-phase, referenced to the positive zero crossing of the fundamental component.

The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:CURRent:NEUTral:HARMonic :PHASe?<NRf> FETCh[:SCALar]:CURRent:NEUTral:HARMonic :PHASe?<NRf>
Parameters	0 to 50
Examples	MEAS:CURR:NEUT:HARM:PHAS? 3 FETC:CURR:NEUT:HARM:PHAS? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:THDistortion:MODE

This command sets the calculation method for THD measurements. The distortion calculation is based on the H2 through H50 with the RMS voltage or current in the denominator. Note that some definitions of THD use the fundamental component (H1) of the voltage as the denominator.

CS units can be programmed to use the fundamental component as the denominator instead of the RMS value. This mode can only be programmed over the bus by sending the “MEAS:THD:MODE FUND” command. At power up or after a reset command, the mode will revert back to the RMS mode. This mode setting is not saved in any of the set up registers.

Syntax	MEASure:THDistortion:MODE	
Parameters	RMSQuare | FUNDamental	
Examples	MEAS:THD:MODE FUND	
Query Syntax	MEAS:THD:MODE?	
Returned Parameters	<CRD>	
Related Commands	MEAS:VOLT:HARM:THD?	MEAS:CURR:HARM:THD?

4.9 Frequency Measurement Subsystem

This subsystem programs the frequency measurement capability of the CS Series AC Current source. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new measurement data before returning a reading. FETCh returns a reading computed from previously acquired data.

Subsystem Syntax

```
MEASure | FETCh
    [:SCALar]
        :FREQuency?                    Returns the output frequency
```

MEASure:FREQuency?
FETCh:FREQuency?

This query returns the output frequency in Hertz.

Query Syntax	MEASure[:SCALar]:FREQuency? FETCh[:SCALar]:FREQuency?
Parameters	None
Examples	MEAS:FREQ? FETC:FREQ?
Returned Parameters	<NR3>

4.10 Power Measurement Subsystem

This subsystem programs the power measurement capability of the CS Series AC Current source. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new measurement data before returning a reading. FETCh returns a reading computed from previously acquired data.

Individual outputs of a three-phase source are specified by the setting of INSTrument:NSELect.

Subsystem Syntax

```
MEASure | FETCh
    [:SCALar]
        :POWer
            [:DC]?                      Returns the DC component of power
            :AC
                [:REAL]?                Returns real power
                :APParent?              Returns VA
                :REACtive?              Returns VAR
                :PFACtor?               Returns power factor
                :TOTal?                 Returns real 3-phase total power
```

MEASure:POWer?
FETCh:POWer?

Phase Selectable

These queries return the DC component of the power being sourced at the output terminals in watts.

Query Syntax	MEASure[:SCALar]:POWer[:DC]? FETCh[:SCALar]:POWer[:DC]?	
Parameters	None	
Examples	MEAS:POW?	FETC:POW?
Returned Parameters	<NR3>	
Related Commands	INST:NSEL	

MEASure:POWer:AC?
FETCh:POWer:AC?

Phase Selectable

These queries return the in-phase component of power being sourced at the output terminals in watts.

Query Syntax	MEASure[:SCALar]:POWer:AC[:REAL]? FETCh[:SCALar]:POWer:AC[:REAL]?	
Parameters	None	
Examples	MEAS:POW:AC?	FETC:POW:AC?
Returned Parameters	<NR3>	
Related Commands	INST:NSEL	

MEASure:POWer:AC:APParent?
FETCh:POWer:AC:APParent?

Phase Selectable

These queries return the apparent power being sourced at the output terminals in volt-amperes.

Query Syntax	MEASure[:SCALar]:POWer:AC:APParent? FETCh[:SCALar]:POWer:AC:APParent?
Parameters	None
Examples	MEAS:POW:AC:APP? FETC:POW:AC:APP?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:POWer:AC:REACtive?
FETCh:POWer:AC:REACtive?

Phase Selectable

These queries return the reactive power being sourced at the output terminals in volt-amperes reactive. Reactive power is computed as:

VAR = sqrt(square(apparent power) - square(real power))

Query Syntax	MEASure[:SCALar]:POWer:AC:REACtive? FETCh[:SCALar]:POWer:AC:REACtive?
Parameters	None
Examples	MEAS:POW:AC:REAC?FETC:POW:AC:REAC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:POWer:AC:PFACtor?
FETCh:POWer:AC:PFACtor?

Phase Selectable

These queries return the output power factor. The power factor is computed as:

pfactor = real power/apparent power

Query Syntax	MEASure[:SCALar]:POWer:AC:PFACtor? FETCh[:SCALar]:POWer:AC:PFACtor?
Parameters	None
Examples	MEAS:POW:AC:PFAC? FETC:POW:AC:PFAC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:POWer:AC:TOTal?
FETCh:POWer:AC:TOTal?

These queries return the total power being sourced at the output terminals of a three-phase AC source.

Query Syntax	MEASure[:SCALar]:POWer:AC:TOTal? FETCh[:SCALar]:POWer:AC:TOTal?
Parameters	None
Examples	MEAS:POW:AC:TOT? FETC:POW:AC:TOT?
Returned Parameters	<NR3>

4.11 Voltage Measurement Subsystem

This subsystem programs the voltage measurement capability of the CS Series AC Current source. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new measurement data before returning a reading. FETCh returns a reading computed from previously acquired data.

Individual outputs of a three-phase source are specified by the setting of INSTrument:NSELect.

Subsystem Syntax

```
MEASure | FETCh
    [:SCALar]
        :VOLTage
            [:DC]?                      Returns the DC component of the voltage
            :AC?                        Returns AC rms voltage
            :ACDC?                      Returns AC+DC rms voltage
            :AMPLitude
                :MAX?                   Returns non-recurring peak current
                :RESet                  Clear the non-recurring peak current.
            :CREStfactor?               Returns current crestfactor
            :HARMonic
                [:AMPLitude]? <n>       Returns amplitude of the Nth harmonic of voltage
                :PHASe? <n>             Returns phase of the Nth harmonic of voltage
                :THD?                   Returns % of total harmonic distortion of voltage
    :THD:MODE RMS | FUND                Sets THD calculation to either RMS or Fundamental
                                        mode.
```

MEASure:VOLTage?
FETCh:VOLTage?

Phase Selectable

These queries return the DC component of the output voltage being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:VOLTage[:DC]? FETCh[:SCALar]:VOLTage[:DC]?
Parameters	None
Examples	MEAS:VOLT? FETC:VOLT?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:AC?
FETCh:VOLTage:AC?

Phase Selectable

These queries return the AC rms voltage being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:VOLTage:AC? FETCh[:SCALar]:VOLTage:AC?
Parameters	None
Examples	MEAS:VOLT:AC? FETC:VOLT:AC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:ACDC?
FETCh:VOLTage:ACDC?

Phase Selectable

These queries return the AC or DC rms voltage being sourced at the output terminals.

Query Syntax	MEASure[:SCALar]:VOLTage:ACDC? FETCh[:SCALar]:VOLTage:ACDC?
Parameters	None
Examples	MEAS:VOLT:ACDC? FETC:VOLT:ACDC?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:AMPLitude:MAXimum?
FETCh:VOLTage:AMPLitude:MAXimum?

Phase Selectable

These queries return the absolute value of the peak voltage as sampled over one measurement acquisition of 4096 data points. Note that the MEAS format returns the non-recurring peak current (as in a peak hold reading). This value can be cleared with the MEAS:CURR:AMP:RES command. The FETC format can be used to obtain the recurring or repetitive peak current this requires the acquisition to be triggered first by either an INIT:ACQ or a MEAS command for another paramter e.g. current, followed by the FETC:CURR:AMPL:MAX?

Query Syntax	MEASure[:SCALar]:VOLTage:AMPLitude:MAXimum? FETCh[:SCALar]:VOLTage:AMPLitude:MAXimum?
Parameters	None
Examples	MEAS:VOLT:AMPL:MAX? FETC:VOLT:AMPL:MAX?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:AMPLitude:RESet
FETCh:VOLTage:AMPLitude:RESet

Phase Selectable

These command resets the peak voltage hold value returned with the MEAS:CURR:AMPL:MAX? query.

Syntax	MEASure[:SCALar]:VOLTage:AMPLitude:RESet FETCh[:SCALar]:VOLTage:AMPLitude:RESet
Parameters	None
Examples	MEAS:VOLT:AMPL:RES
Returned Parameters	None
Related Commands	MEAS:VOLT:AMPL:MAX? FETC:VOLT:AMPL:MAX?

MEASure:VOLTage:CREStfactor?
FETCh:VOLTage:CREStfactor?

Phase Selectable

These queries return the output voltage crest factor. This is the ratio of peak output voltage to rms output voltage.

Query Syntax	MEASure[:SCALar]:VOLTage:CRESTfactor? FETCh[:SCALar]:VOLTage:CRESTfactor?
Parameters	None
Examples	MEAS:VOLT:CRES? FETC:VOLT:CRES?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:HARMonic?
FETCh:VOLTage:HARMonic?

Phase Selectable

These queries return the rms amplitude of the Nth harmonic of output voltage. The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:VOLTage:HARMonic[:AMPLitude]?<NRf> FETCh[:SCALar]:VOLTage:HARMonic[:AMPLitude]?<NRf>
Parameters	0 to 50
Examples	MEAS:VOLT:HARM? 3 FETC:VOLT:HARM? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:HARMonic:PHASe?
FETCh:VOLTage:HARMonic:PHASe?

Phase Selectable

These queries return the phase angle of the Nth harmonic of output voltage, referenced to the positive zero crossing of the fundamental component. The parameter is the desired harmonic number. Queries sent with a value of 0 return the DC component. A value of 1 returns the fundamental output frequency. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 16 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 16 kHz are returned as 0.

Query Syntax	MEASure[:SCALar]:VOLTage:HARMonic:PHASe?<NRf> FETCh[:SCALar]:VOLTage:HARMonic:PHASe?<NRf>
Parameters	0 to 50
Examples	MEAS:VOLT:HARM:PHAS? 3 FETC:VOLT:HARM:PHAS? 1
Returned Parameters	<NR3>
Related Commands	INST:NSEL

MEASure:VOLTage:HARMonic:THD?
FETCh:VOLTage:HARMonic:THD?

Phase Selectable

These queries return the percentage of total harmonic distortion and noise in the output voltage.

Query Syntax	MEASure[:SCALar]:VOLTage:HARMonic:THD? FETCh[:SCALar]:VOLTage:HARMonic:THD?	
Parameters	None	
Examples	MEAS:VOLT:HARM:THD?	FETC:VOLT:HARM:THD?
Returned Parameters	<NR3>	
Related Commands	INST:NSEL	

MEASure:THDistortion:MODE

This command sets the calculation method for THD measurements. The distortion calculation is based on the H2 through H50 with the RMS voltage or current in the denominator. Note that some definitions of THD use the fundamental component (H1) of the voltage or as the denominator.

CS units can be programmed to use the fundamental component as the denominator instead of the RMS value. This mode can only be programmed over the bus by sending the "MEAS:THD:MODE FUND" command. At power up or after a reset command, the mode will revert back to the RMS mode. This mode setting is not saved in any of the set up registers.

Syntax	MEASure:THDistortion:MODE	
Parameters	RMSQuare | FUNDamental	
Examples	MEAS:THD:MODE FUND	
Query Syntax	MEAS:THD:MODE?	
Returned Parameters	<CRD>	
Related Commands	MEAS:VOLT:HARM:THD?	MEAS:CURR:HARM:THD?

4.12 Output Subsystem

This subsystem controls the main outputs, the signal outputs, the power-on state, and the output protection function of the CS Series.

Subsystem Syntax

```
OUTPut
    [:STATe] <bool>               Enable/disable output voltage, current, power, etc.
    :DFI
        [:STATE] <bool>           Enable/disable DFI output
        :SOURce <source>          Selects an event source (QUES|OPER|ESB|RQS|OFF)
    :PON
        :STATe RST | RCL0         Set power-on state to *RST or *RCL0
    :PROTection
        :CLEar                    Reset latched protection
        :DELay <n>                Delay after programming/before protection
    :RI
        :MODE <mode>              set remote inhibit input (LATC|LIVE|OFF)
    :TTLTrg
        :MODE TRIG | FSTR         Sets or disabled Function strobe mode.
        [:STATE] <bool>           Enable/disable trigger out drive
        :SOURce <source>          Selects a TTLTrg source (BOT|EOT|LIST)
```

OUTPut

This command enables or disables the AC source output. The state of a disabled output is an output voltage amplitude set to 0 volts, with output relays opened. When opening the output relay, the output is set to 0 volt first, then the output relay is opened. A user settable delay may be inserted before the output relay is opened. See the “PONSetup:RELay” command for details. Your application program should allow for this delay. (default is 0.1 sec or 100 msec).

The query form returns the output state.

Command Syntax	OUTPut[:STATe]<bool>		
Parameters	0 | OFF | 1 | ON		
*RST Value	OFF		
Examples	OUTP 1	OUTP:STAT ON	
Query Syntax	OUTPut[:STATe]?		
Returned Parameters	0 | 1		
Related Commands	*RCL	*SAV	PONS:REL

OUTPut:DFI

This command enables or disables the discrete fault indicator (DFI) signal to the CS Series AC Current source.

Command Syntax	OUTPut:DFI[:STATe]<bool>	
Parameters	0|1|OFF|ON	
*RST Value	OFF	
Examples	OUTP:DFI 1	OUTP:DFI OFF
Query Syntax	OUTPut:DFI[:STATe]?	
Returned Parameters	0 | 1	
Related Commands	OUTP:DFI:SOUR	

OUTPut:DFI:SOURce

This command selects the source for DFI events. The choices are:

QUEStionable	Questionable summary bit
OPERation	Operation summary bit
ESB	Standard Event summary bit
RQS	Request Service summary bit
OFF	Never true

Command Syntax	OUTP:DFI:SOUR<source>
Parameters	QUES | OPER | ESP | RQS | OFF
*RST Value	OFF
Examples	OUTP:DFI:SOUR OPER
Query Syntax	OUTPut:DFI:SOUR?
Returned Parameters	<CRD>
Related Commands	OUTP:DFI

OUTPut:PON:STATe

This command selects the power-on state of the AC source. The following states can be selected:

RST	Sets the power-on state to *RST. Refer to the *RST command as described later in this chapter for more information.
RCL0	Sets the power-on state to *RCL 0. Refer to the *RCL command as described later in this chapter for more information.

Command Syntax	OUTPut:PON:STATE <state>
Parameters	RST | RCL0
Examples	OUTP:PON:STAT RST
Query Syntax	OUTPut:PON:STATe?
Returned Parameters	<CRD>
Related Commands	*RST *RCL

OUTPut:PROTection:CLEar

This command clears the latch that disables the output when an overvoltage (OV), overcurrent (OC), overtemperature (OT), or remote inhibit (RI) fault condition is detected. All conditions that generated the fault must be removed before the latch can be cleared. The output is then restored to the state it was in before the fault condition occurred.

Command Syntax	OUTPut:PROTection:CLEar
Parameters	None
Examples	OUTP:PROT:CLE
Related Commands	OUTP:PROT:DEL *RCL *SAV

OUTPut:PROTection:DELay

This command sets the delay time between the programming of an output change that produces a CL or UNREG status condition and the recording of that condition by the Status Operation Condition register. The delay prevents momentary changes in status that can occur during programming from being registered as events by the status subsystem. In most cases these temporary conditions are not considered an event, and to record them as such would be a nuisance.

Command Syntax	OUTPut:PROTection:DELay<NRf>
Parameters	0 to 32 | MINimum | MAXimum
Unit	S (seconds)
*RST Value	100 milliseconds
Examples	OUTP:PROT:DEL 75E-1
Query Syntax	OUTPut:PROTection:DELay?
Returned Parameters	<NR3>
Related Commands	OUTP:PROT:CLE *RCL *SAV

OUTPut:RI:MODE

This command selects the mode of operation of the Remote Inhibit protection. The following modes can be selected:

LATChing A TTL low at the RI input latches the output in the protection shutdown state, which can only be cleared by OUTPut:PROTection:CLEar.

LIVE The output state follows the state of the RI input. A TTL low at the RI input turns the output off; a TTL high turns the output on.

OFF The instrument ignores the RI input.

The RI output state is saved as part of an instrument setup using the *SAV command. It can be made part of the power on setting if needed. The default state is LIVE.

Command Syntax	OUTPut:RI:MODE <mode>
Parameters	LATChing | LIVE | OFF
*RST Value	OFF
Examples	OUTP:RI:MODE LIVE
Query Syntax	OUTPut:RI:MODE?
Returned Parameters	<CRD>
Related Commands	OUTP:PROT:CLE

OUTPut:TTLTrg:MODE

This command sets the operation of the Trigger Out signal to either Function Strobe or Trigger mode. Factory default is Trigger state which means the OUTP:TTLT:STAT command is required to generate outputs. In Function Strobe mode, an output pulse is generated automatically any time an output parameter such as current, frequency or phase is programmed. The AC source Trigger Out signal is available at a SMA connector on the rear of the CS Series AC Current source units.

Command Syntax	OUTPut:TTLTrg:MODE TRIG | FSTR
Parameters	TRIG | FSTR
*RST Value	TRIG
Examples	OUTP:TTLT:MODE FSTR
Query Syntax	OUTPut:TTLTrg:MODE?
Returned Parameters	<CRD>
Related Commands	OUTP:TTLT:STAT

OUTPut:TTLTrg[:STATe]

This command enables or disables the AC source Trigger Out1 signal, which is available at a SMA connector on the rear of the CS Series AC Current source units.

Command Syntax	OUTPut:TTLTrg[:STATe]<bool>
Parameters	0|1|OFF|ON
*RST Value	OFF
Examples	OUTP:TTLT 1 OUTP:TTLT OFF
Query Syntax	OUTPut:TTLTrg[:STATe]?
Returned Parameters	0 | 1
Related Commands	OUTP:TTLT:SOUR

OUTPut:TTLTrg:SOURce

This command selects the signal source for the Trig Out signal as follows:

- BOT Beginning of transient output
- EOT End of transient output
- LIST Specified by the TTLTrg list

When an event becomes true at the selected TTLTrg source, a pulse is sent to the SMA connector on the rear of the AC source.

Command Syntax	OUTPut:TTLTrg:SOURce<source>
Parameters	BOT | EOT | LIST
*RST Value	BOT
Examples	OUTP:TTLT:SOUR LIST
Query Syntax	OUTPut:TTLTrg:SOURce?
Returned Parameters	<CRD>
Related Commands	OUTP:TTLT

4.13 Power On Subsystem

This subsystem controls the specific configuration settings at power on. Most power on settings are determined by the power on register recall state using the OUTPut:PON:STATe command. Some aspects are not part of a register however and must be controlled using the PONS commands.

Subsystem Syntax

PONSetup	
:CLOCk	STD | MAST | AUX
:PEAK:CURRent[:PROTection]	Disables peak current protection.
:RELay:HOLD	Sets output off relay open delay in seconds.

PONSetup:CLOCk

This command is used to set the clock and lock mode at power up. It is factory set and should not be changed unless the configuration has been modified in the field. Units with the –LKM option are fixed to MAST mode. Units with the –LKS option can be set to either STANdalone or AUX. When set to AUX, the –LKS unit will power up in external clock mode. When set to STANDalone, the –LKS unit will power up in internal clock mode. The –LKM unit always powers up in internal clock mode. It’s clock state cannot be changed.

Command Syntax	PONSetup:CLOCk
Parameters	STANdalone | MASTer | AUXiliary
Examples	PONS:CLOC
Query Syntax	PONS:CLOC?
Returned Parameters	<CRD>
Related Commands	None

PONSetup:PEAK:CURRent[:PROTection]

This command can be used to disable the peak current shutdown mode. It is factory disabled and should be left disabled for most situations.

Command Syntax	PONSetup:PEAK:CURRent[:PROTection]
Parameters	0 | 1 | OFF | ON
Examples	PONS:PEAK:CURR 1
Query Syntax	PONS:PEAK:CURR?
Returned Parameters	0 | 1
Related Commands	OUTP:PROT:DEL

PONSetup:RELay[:HOLD]

This command is used to set a delay time between programming down the output current to 0 amps and opening the output relay. This provides some time for any inductive loads connected at the output of the power source to discharge into the amplifiers before opening the output relays (OUTP 0 command or front panel On/Off).. Without this delay, inductive EUT's may kick up a high flyback voltage. The same delay time is also used to program down and hold the voltage to zero volt before switching the voltage range relays when performing a voltage range change. (VOLT:RANG command or front panel RANGE change).

The default delay is set to 0.1 or 100 msec. This delay can be set to a value from 0.000 to 1.000 seconds. Once set, it is recalled at power up. Note that this delay affects the time it takes to execute the OUTP 0 and the VOLT:RANG commands. If a delay is set, the bus will be held by this amount of time while the power source executes either of these commands.

Note that this delay time is approximate only and may vary somewhat from unit to unit.

Command Syntax	PONSetup:RELay[:HOLD]	
Parameters	0.000 through 1.000	
Examples	PONS:REL 0.2	
Query Syntax	PONS:REL?	
Returned Parameters	<NR3>	
Related Commands	OUTP 1	VOLT:RANG

4.14 Sense Subsystem - Sweep

This subsystem controls the measurement current range and the data acquire sequence of the AC source.

Subsystem Syntax

```
SENSe
    [:COUPle]   AC | DC                        ADC coupling mode.
    :SWEep
        :OFFSet                                Define trigger delay in time relative to start of the
                                               digitizer data record
            :POINts <n>                        Define trigger points relative to the start of the
                                               digitizer data record
        :TINTerval <n>                         Sets the digitizer sample spacing
```

SENSe[:COUPle]

This command sets the coupling mode for the ADC of the measurement system. Available coupling modes are AC or DC. Factory default is DC coupled.

Command Syntax	SENSe[:COUPle] <CRD>
Parameters	AC | DC
*RST Value	DC
Examples	SENS:COUP DC
Query Syntax	SENS:COUP?
Returned Parameters	<CRD>
Related Commands	SENS:SWE:TINT

SENSe:SWEep:OFFSet

This command defines the trigger point expressed in seconds relative to the start of the returned data record when an acquire trigger is used. The values can range from MIN to MAX depending on the phase mode and the selected sample interval. When the values are negative, the values in the beginning of the data record represent samples taken prior to the trigger.

Command Syntax	SENSe:SWEep:OFFSet <NRf+>	
Parameters	<NRf> | MINimum | MAXimum	
*RST Value	0	
Examples	SENS:SWE:OFFS -12E-3	
Query Syntax	SENSe:SWEep:OFFSet?	
Returned Parameters	<NR3>	
Related Commands	SENS:SWE:TINT	MEAS:ARR

SENSe:SWEep:OFFSet:POINts

This command defines the trigger point expressed in sample points relative to the start of the returned data record when an acquire trigger is used. The values can range from -4095 to 2E9. When the values are negative, the values in the beginning of the data record represent samples taken prior to the trigger.

Command Syntax	SENSe:SWEep:OFFSet:POINts<NRf+>	
Parameters	4096 through 2E9 | MINimum | MAXimum	
*RST Value	0	
Examples	SENS:SWE:OFFS:POIN -2047	
Query Syntax	SENSe:SWEep:OFFSet:POINts?	
Returned Parameters	<NR3>	
Related Commands	SENS:SWE:TINT	MEAS:ARR

SENSe:SWEep:TINTerval

This command defines the time period between samples. The sample period can be programmed from 10.42 to 104.2 microseconds in 10 microsecond increments in single-phase mode and from 31.25 to 312.5 microseconds in three-phase mode.

All the MEASure commands use the ACQuire trigger sequence implicitly. These commands always set the sample period to 21 microseconds.

Command Syntax	SENSe:SWEep:TINTerval<NRf+>	
Parameters	10.42 through 104.2 (microseconds)	
*RST Value	10.42 us	
Examples	SENS:SWE:TINT 100	
Query Syntax	SENSe:SWEep:TINTerval?	
Returned Parameters	<NR3>	
Related Commands	SENS:SWE:OFFS:POIN	MEAS:ARR

4.15 Source Subsystem - Voltage

This subsystem programs the compliance voltage limit setting of the AC current source.

Subsystem Syntax

```
[SOURce:]
    VOLTage
        [:LEVel]
            [:IMMediate]
                [:AMPLitude] <n>              Sets the rms voltage limit
        :PROTection
            :DELay                            Voltage limit fault delay
            :STATe <bool>                     Enable/Disable rms voltage limit protection
        :SENSe
            [:SOURce] INTernal | EXTernal     Sets voltage sense source
```

VOLTage

Phase Selectable

This command sets the rms voltage limit of the specified output phase. If the output voltage exceeds this limit, the output current amplitude is reduced until the rms voltage is within the limit. The CL bit of the questionable status register indicates that the voltage limit control loop is active. If the voltage protection state is programmed on, the output latches into a disabled state when voltage limiting occurs.

To determine the maximum available voltage, use the “volt? max” query

Command Syntax	[SOURce:]VOLTage[:LEVel] [:IMMediate][:AMPLitude]<NRf+>
Parameters	0 to max. available voltage
Unit:	V (rms voltage)
*RST	1
Examples	VOLT 5.0 VOLT:LEV .5.0
Query Syntax	[SOURce:]VOLTage[:LEVel] [:IMMediate][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	VOLT:PROT:STAT

VOLTage:PROTection:DELay

This command holds off the over voltage trip of the output current for the time specificied. Default value at *RST is 0.1 sec. The range is from 0.1 to 5.000 secs and can be queries with the VOLT:PROT:DEL? MIN and VOLT:PROT:DEL? MAX commands.

Command Syntax	[SOURce:]VOLTage:PROTection:DELay <NRf+>
Parameters	0.100 to 5.000 | MINimum | MAXimum
Unit:	S (seconds)
*RST Value	0.100
Examples	VOLT:PROT:DEL 1.5
Query Syntax	[SOURce:]VOLTage:PROTection:DELay? VOLTage:PROTection:DELay? Min VOLTage:PROTection:DELay? Max
Returned Parameters	<NR3>
Related Commands	VOLT:PROT:STAT

VOLTage:PROTection:STATe

This command enables or disables the AC source over voltage(OV) protection function. If the over voltage protection function is enabled and the AC source exceeds the programmed level, then the output is disabled and the Questionable Condition status register OV bit is set (see Chapter 7). An over voltage condition can be cleared with OUTPut:PROTection:CLEar after the cause of the condition is removed.

Use OUTP:PROT:DEL to prevent momentary votlage limit conditions caused by programmed output changes from tripping the over votlage protection. Use VOLT:PROT:DEL to hold off tripping the output due to temporary overload conditions.

Command Syntax	[SOURce:]VOLTage:PROTection:STATe<bool>	
Parameters	0 | 1 | OFF | ON	
*RST Value	OFF	
Examples	VOLT:PROT:STAT 0	VOLT:PROT:STAT OFF
Query Syntax	[SOURce:]VOLTage:PROTection:STATe?	
Returned Parameters	0 | 1	
Related Commands	OUTP:PROT:CLE	VOLT:PROT:DEL

VOLTage:SENSe:SOURce

These commands select the source from which the output voltage is sensed. The following voltage sense sources can be selected:

INTernal	This senses the voltage at the output of the power amplifier on the inboard side of the output disconnect relay.
EXTernal	This senses the output voltage at the user's sense terminals, which allows remote voltage sensing at the load.

Command Syntax	[SOURce:]VOLTage:SENSe:SOURce<source> [SOURce:]VOLTage:ALC:SOURce<source>
Parameters	INTernal | EXTernal
*RST Value	INTernal
Examples	VOLT:SENS:SOUR INT
Query Syntax	[SOURce:]VOLTage:SENSe:SOURce?
Returned Parameters	<CRD>
Related Commands	VOLT:ALC:SOUR EXT

Note: The VOLT:ALC:SOUR command is an alias for the VOLT:SENS:SOUR command. Both perform the same function.

4.16 Source Subsystem - Frequency

This subsystem programs the output frequency of the AC source.

Subsystem Syntax

```
[SOURce:]
    FREQuency
        [:CW | :IMMediate] <n>          Sets the frequency
        :MODE <mode>                    Sets frequency mode
                                        (FIX|STEP|PULS|LIST|SENS|EXT)
        :SLEW
           [:IMMediate] <n> | INFinity  Sets the frequency slew rate
           :MODE <mode>                 Sets frequency slew mode (FIX|STEP|PULS|LIST)
           :TRIGgered <n> | INFinity    Sets the triggered frequency slew rate
        :TRIGgered <n>                  Sets the triggered frequency
```

FREQuency

This command sets the frequency of the output waveform.

Command Syntax	[SOURce:]FREQuency[:CW|:IMMediate]<NRf+>
Parameters	45 to 2000
Unit	HZ (Hertz)
*RST Value	60 Hz
Examples	FREQ 50
Query Syntax	[SOURce:]FREQuency?
Returned Parameters	<NR3>
Related Commands	FREQ:MODE FREQ:SLEW

FREQuency:MODE

This command determines how the output frequency is controlled. Available modes are:

FIXed — The output frequency is unaffected by a triggered output transient. The clock source is the internal controller timebase.

STEP — The output frequency is programmed to the value set by FREQuency:TRIGgered when a triggered transient occurs.

PULSe — The output frequency is changed to the value set by FREQuency:TRIGgered for a duration determined by the pulse commands.

LIST — The output frequency is controlled by the frequency list when a triggered transient occurs.

SENSe — External sync or line sync clock mode. The frequency is synchronized to the external sync or line sync frequency.

EXTernal — External clock mode. This mode is used on a CS unit with the –LKS option to sync to a master CS unit.

Command Syntax	[SOURce:]FREQuency:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST | SENSe | EXTernal
*RST Value	FIXed
Examples	FREQ:MODE FIX
Query Syntax	[SOURce:]FREQuency:MODE?
Returned Parameters	<CRD>
Related Commands	FREQ FREQ:TRIG

FREQuency:SLEW

This command sets the rate at which frequency changes for all programmed changes in output frequency. Instantaneous frequency changes can be obtained by sending MAXimum or INFinity. The SCPI keyword INFinity is represented by the number 9.9E37.

Command Syntax	[SOURce:]FREQuency:SLEW[:IMMediate]<NRf+> |INFinity
Parameters	1E-3 to 9.9E37 | INFinity |MINimum | MAXimum
Unit	HZ (Hertz per second)
*RST Value	MAXimum
Examples	FREQ:SLEW:IMM 75 FREQ:SLEW MAX
Query Syntax	[SOURce:]FREQuency:SLEW?
Returned Parameters	<NR3>
Related Commands	FREQ:SLEW:MODE FREQ

FREQuency:SLEW:MODE

This command determines how the frequency slew rate is controlled during a triggered output transient. The choices are:

FIXed The frequency slew rate is unaffected by a triggered output transient.

STEP The frequency slew rate is programmed to the value set by FREQuency:TRIGgered when a triggered transient occurs.

PULSe The frequency slew rate is changed to the value set by FREQuency:TRIGgered for a duration determined by the pulse commands.

LIST The frequency slew rate is controlled by the frequency list when a triggered transient occurs.

Command Syntax	[SOURce:]FREQuency:SLEW:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST
*RST Value	FIXed
Examples	FREQ:SLEW:MODE FIX
Query Syntax	[SOURce:]FREQuency:SLEW:MODE?
Returned Parameters	<CRD>
Related Commands	FREQ FREQ:SLEW:TRIG

FREQency:SLEW:TRIGgered

This command sets the rate at which frequency changes during a triggered output transient. Instantaneous frequency changes can be obtained by sending MAXimum or INFinity. The SCPI keyword INFinity is represented by the number 9.9E37.

Command Syntax	[SOURce:]FREQuency:SLEW:TRIGgered<NRf+> |INFinity
Parameters	1E-3 to 9.9E37 | INFinity |MINimum | MAXimum
Unit	HZ (Hertz per second)
*RST Value	MAXimum
Examples	FREQ:SLEW:TRIG 75 FREQ:SLEW:TRIG MAX
Query Syntax	[SOURce:]FREQuency:SLEW:TRIG?
Returned Parameters	<NR3>
Related Commands	FREQ:SLEW:MODE FREQ

FREQuency:TRIGgered

This command programs the frequency that the output will be set to during a triggered step or pulse transient.

Command Syntax	[SOURce:]FREQuency:TRIGgered<NRf+>
Parameters	Refer to specifications table in User Manual
Unit	HZ (Hertz)
*RST Value	60 Hz
Example	FREQ:TRIG 50
Query Syntax	[SOURce:]FREQuency:TRIGgered?
Returned Parameters	<NR3>
Related Commands	FREQ FREQ:MODE

4.17 Source Subsystem - Function

This subsystem programs the output function of the AC source.

Subsystem Syntax

```
[SOURce:]
    FUNCtion
        [:SHAPe]
            [:IMMediate] <shape>      Sets the periodic waveform shape
                                      (SIN|SQU|CSIN|<user-defined>)
            :MODE <mode>              Sets the waveform shape mode (FIX|STEP|PULS|LIST)
            :TRIGgered <shape>        Sets the triggered transient shape
                                      (SIN|SQU|CSIN|<user-defined>)
            :CSINusoid <n> [THD]      Sets the % of peak at which the clipped sine clips (or %
                                      THD)
```

FUNCtion

This command selects the shape of the output current waveform as follows:

SINusoid A sinewave is output

SQUare A squarewave is output

CSINusoid The output is a clipped sine waveform. Both positive and negative peak amplitudes are clipped at a value determined by the SOURce:FUNCtion:SHAPe:CSINusoid setting.

<user_defined> The output shape is described by one of the user-defined waveform tables.

The maximum peak current that the AC source can output in 3 phase mode is 20.95 A peak. This includes any combination of current and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum current that can be programmed is 14.81 A rms.

Before programming a different waveform shape, the output current should be programmed to zero amps. After the shape is changed, the current maybe programmed to the desired value.

Command Syntax	[SOURce:]FUNCtion[:SHAPe][:IMMediate]<shape>
Parameters	SINusoid|SQUare|CSINusoid|<waveform_name>
*RST Value	SINusoid
Examples	FUNC SIN FUNC USERNAME
Query Syntax	[SOURce:]FUNCtion[:SHAPe]?
Returned Parameters	<CRD>
Related Commands	FUNC:MODE FUNC:TRIGVOLT

FUNCtion:MODE

This command determines how the waveform shape is controlled during a triggered output transient. The choices are:

FIXed The waveform shape is unaffected by a triggered output transient.

STEP The waveform shape is programmed to the value set by FUNCtion:TRIGgered when a triggered transient occurs.

PULSe The waveform shape is changed to the value set by FUNCtion:TRIGgered for a duration determined by the pulse commands.

LIST	The waveform shape is controlled by the waveform shape list when a triggered transient occurs.

Command Syntax	[SOURce:]FUNCtion[:SHAPe]:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST
*RST Value	FIXed
Examples	FUNC:MODE FIX
Query Syntax	[SOURce:]FUNCtion[:SHAPe]:MODE?
Returned Parameters	<CRD>
Related Commands	FUNC FUNC:TRIG

FUNCtion:TRIGgered

This command selects the shape of the output current waveform when a triggered step or pulse transient occurs. The parameters are:

SINusoid	A sinewave is output
SQUare	A squarewave is output
CSINusoid	The output is a clipped sine waveform. Both positive and negative peak amplitudes are clipped at a value determined by SOURce:FUNCtion:SHAPe:CSINusoid.
<waveform_name>	The output shape is described by one of the user-defined waveform tables.

The maximum peak current that the AC source can output in 3 phase mode is 20.95 A peak. This includes any combination of voltage and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum current that can be programmed is 14.81 A rms.

Command Syntax	[SOURce:]FUNCtion[:SHAPe]:TRIGgered<shape>
Parameters	SINusoid|SQUare|CSINusoid|<waveform_name>
*RST Value	SINusoid
Examples	FUNC:TRIG SIN FUNC:TRIG TABLE1
Query Syntax	[SOURce:]FUNCtion[:SHAPe]:TRIGgered?
Returned Parameters	<CRD>
Related Commands	FUNC FUNC:MODEVOLT

FUNCtion:CSINusoid

This command sets the clipping level when a clipped sine output waveform is selected. The clipping characteristics can be specified in two ways:

- The clipping level is expressed as a percentage of the peak amplitude at which clipping occurs. The range is 0 to 100 percent. These are the default units when the optional THD suffix is not sent.
- The clipping level is expressed at the percentage of total harmonic distortion in the output voltage. The range is 0 to 43 percent. The optional THD suffix is sent to program in these units.

Command Syntax	[SOURce:]FUNCtion[:SHAPe]:CSINusoid<NRf>[THD]
Parameters	0 to 100% | 0 to 43% THD
*RST Value	100% | 0% THD (no clipping)
Examples	FUNC:CSIN 80 FUNC:CSIN 10 THD
Query Syntax	[SOURce:]FUNCtion[:SHAPe]:CSINusoid?
Returned Parameters	<NR3>
Related Commands	FUNC:MODE

4.18 Source Subsystem - List

This subsystem controls the generation of complex sequences of output changes with rapid, precise timing and synchronized with internal or external signals. Each subsystem command for which lists can be generated has an associated list of values that specify the output at each list step. LIST:COUNt determines how many times the AC source sequences through a list before that list is completed. LIST:DWELl specifies the time interval that each value (point) of a list is to remain in effect. LIST:STEP detemines if a trigger causes a list to advance only to its next point or to sequence through all of its points.

All active subsystems that have their modes set to LIST must have the same number of points (up to 100), or an error is generated when the first list point is triggered. The only exception is a list consisting of only one point. Such a list is treated as if it had the same number of points as the other lists, with all of the implied points having the same value as the one specified point. All list point data is stored in nonvolatile memory.

MODE commands such as CURRent:MODE:LIST are used to activate lists for specific functions (However, the LIST:DWELl command is active whenever any function is set to list mode. Therefore, LIST:DWELl must always be set either to one point, or to the same number of points as the active list.

Subsystem Syntax

```
[SOURce:]
    LIST
        :COUNt <n> | INFinity        Sets the list repeat count
        :CURRent
            [:LEVel] <n>{,<n>}       Sets the current list
                :POINts?             Returns the number of current level points
            :SLEW <n>{,<n>}          Sets the current slew list
                :POINts?             Returns the number of current slew points
        :DWELl <n>{,<n>}             Sets the list of dwell times
            :POINts?                 Returns the number of dwell list points
        :FREQuency
            [:LEVel] <n>{,<n>}       Sets the frequency list
                :POINts?             Returns the number of frequency points
            :SLEW <n>{,<n>}          Sets the frequency slew list
                :POINts?             Returns the number of frequency slew points
        :PHASe <n>{,<n>}             Sets the phase list
            :POINts?                 Returns the number of phase list points
        :STEP ONCE | AUTO            Defines whether list is dwell- or trigger-paced
        :TTLTrg <bool>{,<bool>}      Defines the output marker list
            :POINts?                 Returns the number of output marker list points
```

LIST:COUNt

This command sets the number of times that the list is executed before it is completed. The command accepts parameters in the range 1 through 2E8. Use MAX to set the list to maximum.

Command Syntax	[SOURce:]LIST:COUNt<NRf+> | MAX
Parameters	1 to 2E8 | MINimum | MAXimum
*RST Value	1
Examples	LIST:COUN 3 LIST:COUN INF
Query Syntax	[SOURce:]LIST:COUNt?
Returned Parameters	<NR3>
Related Commands	LIST:CURRLIST:FREQ
	LIST:TTLTLIST:VOLT

LIST:CURRent

This command specifies the output current points in a list. The current points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which the list will be output when a list transient is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:CURRent[:LEVel] <NRf+>{,<NRf+>}
Parameters	0.000 to 14.8
Unit	A (rms current)
Examples	LIST:CURR 0.1,0.5,0.3
	LIST:CURR MAX,0.5,MIN
Query Syntax	[SOURce:]LIST:CURRent[:LEVel]?
Returned Parameters	<NR3>
Related Commands	LIST:CURR:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:CURRent:POINts?

This query returns the number of points specified in LIST:CURR. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST:CURRent:POINts?
Returned Parameters	<NR1>
Example	LIST:CURR:POIN?
Related Commands	LIST:CURR

LIST: CURRent:SLEW

This command specifies the output offset slew points in a list. The slew points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which the list will be output when a list transient is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:CURRent:SLEW <NRf+>{,<NRf+>}
Parameters	1E-4 to 9.9E37 | INFinity | MINimum | MAXimum
Unit	A/S (volts per second)
Example	LIST:CURR:SLEW 10, 1E2, INF
Query Syntax	[SOURce:]LIST:CURRent:SLEW?
Returned Parameters	<NR3>
Related Commands	LIST:CURR:SLEW:POIN? LIST:COUN LIST:DWEL
	LIST:STEP

LIST: CURRent:SLEW:POINts?

This query returns the number of points specified in LIST:CURRent:SLEW. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST: CURRent:SLEW:POINts?	
Returned Parameters	<NR1>	
Example	LIST:CURR:SLEW:POIN?	LIST:CURR:SLEW:POIN? MAX
Related Commands	LIST:CURR:SLEW	

LIST:DWELl

This command sets the sequence of list dwell times. Each value represents the time in seconds that the output will remain at the particular list step point before completing the step. At the end of the dwell time, the output of the AC source depends upon the following conditions:

- If LIST:STEP AUTO has been programmed, the output automatically changes to the next point in the list.
- If LIST:STEP ONCE has been programmed, the output remains at the present level until a trigger sequences the next point in the list.

The order in which the points are entered determines the sequence in which they are output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:DWELl<NRf+>{,<NRf+>}
Parameters	3-phase mode: 0 to 1.07533E6|MINimum|MAXimum 1-phase mode: 0 to 4.30133E5|MINimum|MAXimum
Unit	S (seconds)
Examples	LIST:DWEL .5,.5,1.5
Query Syntax	[SOURce:]LIST:DWEL?
Returned Parameters	<NR3>
Related Commands	LIST:FREQ LIST:TTLT LIST:VOLT

LIST:DWELl:POINts?

This query returns the number of points specified in LIST:DWELl. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST:DWELl:POINts?
Returned Parameters	<NR1>
Example	LIST:DWEL:POIN?
Related Commands	LIST:DWEL

LIST:FREQuency

This command sets the sequence of frequency list points. The frequency points are given in the command parameters, which are separated by commas.

The order in which the points are entered determines the sequence in which they are output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:FREQuency[:LEVel]<NRf+>{,<NRf+>}
Parameters	45 to 5000
Unit	HZ (Hertz)
Examples	LIST:FREQ 60,65,70
Query Syntax	[SOURce:]LIST:FREQ?
Returned Parameters	<NR3>
Related Commands	LIST:FREQ:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:FREQ:SLEW

LIST:FREQuency:POINts?

This query returns the number of points specified in LIST:FREQuency. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST:FREQ[:LEVel]:POINts?
Returned Parameters	<NR1>
Example	LIST:FREQ:POIN?
Related Commands	LIST:FREQ

LIST:FREQuency:SLEW

This command sets the sequence of frequency slew list points. The frequency points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which they are output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:FREQuency:SLEW<NRf+>{,<NRf+>}
Parameters	0 to 9.9E31 | INFinity
Unit	HZ (Hertz) per second
Examples	LIST:FREQ:SLEW 10, 1E2, INF
Query Syntax	[SOURce:]LIST:FREQ:SLEW?
Returned Parameters	<NR3>
Related Commands	LIST:FREQ:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:FREQ

LIST:FREQuency:SLEW:POINts?

This query returns the number of points specified in LIST:FREQuency:SLEW. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST:FREQ:SLEW:POINts?
Returned Parameters	<NR1>
Example	LIST:FREQ:SLEW:POIN?
Related Commands	LIST:FREQ:SLEW

LIST:PHASe

Phase Selectable

This phase selectable command sets the sequence of phase list points. The phase points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which they are output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:PHASe<NRf+>{,<NRf+>}
Parameters	360 through +360
Examples	LIST:PHAS 90,120,135
Query Syntax	[SOURce:]LIST:PHAS?
Returned Parameters	<NR3>
Related Commands	LIST:PHAS:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:PHASe:POINts?

This query returns the number of points specified in LIST:PHASe. Note that it returns only the total number of points, not the point values.

Query Syntax	SOURce:]LIST:PHASe:POINts?
Returned Parameters	NR3>
Example	IST:PHAS:POIN?
Related Commands	IST:FREQ LIST:DWEL

LIST:STEP

This command specifies how the list sequencing responds to triggers.

ONCE causes the list to advance only one point after each trigger. Triggers that arrive during a dwell delay are ignored.

AUTO causes the entire list to be output sequentially after the starting trigger, paced by its dwell delays. As each dwell delay elapses, the next point is immediately output.

Command Syntax	[SOURce:]LIST:STEP<step>
Parameters	ONCE | AUTO
*RST Value	AUTO
Examples	LIST:STEP ONCE
Query Syntax	[SOURce:]LIST:STEP?
Returned Parameters	<CRD>
Related Commands	LIST:COUN LIST:DWEL

LIST:TTLTrg

This command sets the sequence of Trigger Out list points. Each point which is set ON will cause a pulse to be output at Trigger Out when that list step is reached. Those entries which are set OFF will not generate Trigger Out pulses. The order in which the list points are given determines the sequence in which Trigger Out pulses will be output when a list transient is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

Command Syntax	[SOURce:]LIST:TTLTrg<bool>{,<bool>}
Parameters	0 | 1 | OFF | ON
Examples	LIST:TTLT 1,0,1 LIST:TTLT ON,OFF,ON
Query Syntax	LIST:TTLT?
Returned Parameters	0 | 1
Related Commands	LIST:TTLT:POIN? LIST:COUN LIST:DWEL LIST:STEP OUTP:TTLT:STAT OUTP:TTLT:SOUR

LIST:TTLTrg:POINts?

This query returns the number of points specified in LIST:TTLT. Note that it returns only the total number of points, not the point values.

Query Syntax	[SOURce:]LIST:TTLTrg:POINts?
Returned Parameters	<NR1>
Example	LIST:TTLT:POIN?
Related Commands	LIST:TTLT

4.19 Source Subsystem - Phase

This subsystem programs the output phases of the AC source. When phase commands are used to program single-phase units, the only discernible effect in using the phase commands is to cause an instantaneous shift in the output waveform phase.

Subsystem Syntax

```
[SOURce:]
    PHASe
        [:IMMediate] <n>                Sets the output phase
        :MODE <mode>                    Sets the phase mode (FIX|STEP|PULS|LIST)
        :TRIGgered <n>                  Sets the triggered phase (step or pulse mode only)
```

PHASe

Phase Selectable

This command sets the phase of the output voltage waveform relative to an internal reference. The phase angle is programmed in degrees. Positive phase angles are used to program the leading phase, and negative phase angles are used to program the lagging phase.

The PHASe command is not influenced by INSTrument:COUPle ALL. It applies only to the current output phase selected by INSTrument:NSELect.

Command Syntax	[SOURce:]PHASe[IMMediate]<NRf+>
Parameters	-360º through +360º
*RST Value	phase ø1 = 0°, phase ø2 = 240°, phase ø3 = 120°
Examples	PHAS 45 PHASE MAX
Query Syntax	[SOURce:]PHASe?
Returned Parameters	<NR3>
Related Commands	PHAS:MODE PHAS:TRIG

PHASe:MODE

Phase Selectable

This command determines how the output phase is controlled during a triggered output transient. The choices are:

FIXed	The output phase is unaffected by a triggered output transient.
STEP	The output phase is programmed to the value set by PHASe:TRIGgered when a triggered transient occurs.
PULSe	The output phase is changed to the value set by PHASe:TRIGgered for a duration determined by the pulse commands.
LIST	The waveform shape is controlled by the phase list when a triggered transient occurs.

Command Syntax	[SOURce:]PHASe:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST
*RST Value	FIX
Examples	PHAS:MODE LIST PHAS:MODE FIX
Query Syntax	[SOURce:]PHASe:MODE?
Returned Parameters	<CRD>
Related Commands	PHAS:TRIG PHAS

PHASe:TRIGgered

Phase Selectable

This command sets the output phase when a triggered step or pulse transient occurs. The phase of the output current waveform is expressed relative to an internal reference. The phase angle is programmed in degrees. Positive phase angles are used to program the leading phase, and negative phase angles are used to program the lagging phase.

The PHASe command is not influenced by INSTrument:COUPle ALL. It applies only to the current output phase selected by INSTrument:NSELect.

Command Syntax	[SOURce:]PHASe:TRIGgered<NRf+>
Parameters	-360° through +360°
*RST Value	triggered phase ø1 = 0°, triggered phase ø2 = 120°, triggered phase ø3 = 240°
Examples	PHAS:TRIG 120 PHASE MAX
Query Syntax	[SOURce:]PHASe:TRIGgered?
Returned Parameters	<NR3>
Related Commands	PHAS:MODE PHAS

4.20 Source Subsystem - Pulse

This subsystem controls the generation of output pulses. The PULSe:DCYCle, PULSe:HOLD, PULSe:PERiod, and PULSe:WIDTh commands are coupled, which means that the values programmed by any one of these commands can be affected by the settings of the others. Refer to the tables under PULSe:HOLD for an explanation of how these commands affect each other.

Subsystem Syntax

```
[SOURce:]
    PULSe
        :COUNt <n> | INFinity         Selects transient pulse count
        :DCYCle <n>                   Selects pulse duty cycle
        :HOLD WIDTh |DCYCle           Selects parameter that is held constant
        :PERiod <n>                   Selects pulse period when the count is greater than 1
        :WIDTh <n>                    Selects width of the pulses
```

PULSe:COUNt

This command sets the number of pulses that are output when a triggered output transient occurs. The command accepts parameters in the range 1 through 2E8. If INFinity or MAXimum is sent, the output pulse repeats indefinitely.

Command Syntax	[SOURce:]PULSe:COUNt<NRf+> | INFinity
Parameters	1 to 2E8 | MINimum | MAXimum | INFinity
*RST Value	1
Examples	PULS:COUN 3 PULS:COUN MIN PULS:COUN INF
Query Syntax	[SOURce:]PULS:COUNt?
Returned Parameters	<NR3>
Related Commands	PULS:DCYC PULS:HOLD PULS:PER PULS:WIDT

PULSe:DCYCle

This command sets the duty cycle of the triggered output pulse. The duty cycle units are specified in percent.

Command Syntax	[SOURce:]PULSe:DCYCle<NRf+>
Parameters	0 to 100%|MINimum|MAXimum
*RST Value	50%
Examples	PULS:DCYC 75 PULS:DCYC MAX
Query Syntax	[SOURce:]PULSe:DCYCle?
Returned Parameters	<NR3>
Related Commands	PULS:COUN PULS:HOLD PULS:PER PULS:WIDT

PULSe:HOLD

This command specifies whether the pulse width or the duty cycle is to be held constant when the pulse period changes. The following tables describe how the duty cycle, period, and width are affected when one, two, or all three parameters are set in a single program message.

Command Syntax	[SOURce:]PULSe:HOLD<parameter>
Parameters	WIDTh|DCYCle
*RST Value	WIDTh
Examples	PULS:HOLD DCYC
Query Syntax	[SOURce:]PULSe:HOLD?
Returned Parameters	<CRD>
Related Commands	PULS:COUN PULS:DCYC PULS:PER PULS:WIDT

Parameter Set			Action
DCYCle	PERiod	WIDTh	
		√	Sets WIDTh. If WIDTh < PERiod, recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle.
	√		Sets PERiod. If WIDTh < PERiod, recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle.
	√	√	Sets WIDTh. If WIDTh < PERiod, sets the PERiod and recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle
√			Sets DCYCle and recalculates PERiod
√		√	Sets DCYCle and WIDTh and recalculates PERiod
√	√		Sets DCYCle and PERiod and recalculates WIDTh
√	√	√	Sets WIDTh. If WIDTh < PERiod, sets the PERiod and recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle

Table 4-1: PULSe:HOLD = WIDTh parameters

Parameter Set			Action
DCYCle	PERiod	WIDTh	
		√	Sets WIDTh and recalculates the PERiod
	√		Sets PERiod and recalculates the WIDTh
	√	√	Sets WIDTh. If WIDTh < PERiod, sets the PERiod and recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle
√			Sets DCYCle and recalculates PERiod
√		√	Sets DCYCle and WIDTh and recalculates PERiod
√	√		Sets DCYCle and PERiod and recalculates WIDTh
√	√	√	Sets WIDTh. If WIDTh < PERiod, sets the PERiod and recalculates DCYCle; otherwise, recalculates the PERiod and DCYCle

Table 4-2: PULSe:HOLD = DCYCle parameters

PULSe:PERiod

This command sets the period of a triggered output transient. The command parameters are mode-dependent.

Command Syntax	[SOURce:]PULSe:PERiod<NRf+>
Parameters	3-phase mode: 0 to 1.07533E6 | MINimum | MAXimum
	1-phase mode: 0 to 4.30133E5 | MINimum | MAXimum
Unit	s (seconds)
*RST Value	0.03333
Examples	PER 0.001 PER MIN
Query Syntax	[SOURce:]PERiod?
Returned Parameters	<NR3>
Related Commands	PULS:COUN PULS:DCYC PULS:HOLD PULS:WIDT

PULSe:WIDTh

This command sets the width of a transient output pulse. The command parameters are mode-dependent.

Command Syntax	[SOURce:]PULSe:WIDTh<NRf+>
Parameters	3-phase mode: 0 to 1.07533E6 | MINimum | MAXimum 1-phase mode: 0 to 4.30133E5 | MINimum | MAXimum
Unit	s (seconds)
*RST Value	0.01667 (equals the period of a single 60 Hz cycle)
Examples	PULS:WIDT 0.001 PULS:WIDT MIN
Query Syntax	[SOURce:]PULSe:WIDTh?
Returned Parameters	<NR3>
Related Commands	PULS:COUN PULS:DCYC PULS:HOLD PULS:PER

4.21 Source Subsystem - Current

This subsystem programs the output current of the CS Series AC Current source.

Subsystem Syntax

```
[SOURce:]
    CURRent
        :ALC
            [:STATe]  ON | OFF | REG        Sets Auto Level Control to on (trip), off, or regulation
        [:LEVel]
            [:IMMediate]
                [:AMPLitude] <n>            Sets the AC rms current amplitude
            :TRIGgered
                [:AMPLitude] <n>            Sets the transient current amplitude
        :MODE <mode>                        Sets the current mode (FIX|STEP|PULS|LIST)
        :SLEW
            [:IMMediate] <n> | INFinity     Sets the current slew rate
            :MODE <mode>                    Sets current slew mode (FIX|STEP|PULS|LIST)
            :TRIGgered <n> | INFinity       Sets the transient current slew rate
```

CURRent:ALC[:STATe]

These commands select the various auto level control (ALC) modes. The ALC mode uses the current measurement feedback to more precisely regulate the output current. The following ALC modes can be selected:

ON | 0 This enables the ALC trip mode. If the programmed current cannot be maintained, the output is disabled (relay opens) and a 801 Current error is generated.

OFF | 1 This disables the ALC mode.

REG | 2 This enables the ALC regulation mode. The output current is regulated based on the readback current but if regulation cannot be maintained, the output does not trip off. Instead, a status bit is set in the Event Status register to indicate the AC source is out of regulation.

Note that the command format will take either alphanumeric or integer data but the query form always returns numeric data (NR1).

Command Syntax	[SOURce:]CURRent:ALC[:STATe] <source>
Parameters	ON | OFF | REG | 0 | 1 | 2
*RST Value	REG
Examples	CURR:ALC ON
Query Syntax	[SOURce:] CURRent:ALC?
Returned Parameters	<NR1>
Related Commands	CURR

CURRent

Phase Selectable

This command programs the AC rms output current level of the AC source.

Command Syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]<NRf+>
Parameters	0.00 to 14.80
Unit	A (rms voltage)
*RST Value	0.00 A
Examples	CURR 0.1 CURR:LEV 0.1
Query Syntax	[SOURce:]CURRent[:LEVel] [:IMMediate][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	CURR:MODE CURR:TRIG

CURRent:TRIGgered

Phase Selectable

This command selects the AC rms current amplitude that the output waveform will be set to during a triggered step or pulse transient.

Command Syntax	[SOURce:]CURRent[:LEVel]:TRIGgered :AMPLitude]<NRf+>
Parameters	0.00 to 14.80
Unit	A (rms voltage)
*RST Value	0.00 A
Examples	CURR:TRIG 0.1 CURR:LEV:TRIG 0.1
Query Syntax	SOURce:]CURRent[:LEVel]:TRIGgered:AMPLitude]?
Returned Parameters	<NR3> If the TRIG level is not programmed, the IMM level is returned.
Related Commands	CURR CURR:MODE

CURRent:MODE

Phase Selectable

This command determines how the AC rms output current is controlled during a triggered output transient. The choices are:

FIXed	The current is unaffected by a triggered output transient.
STEP	The current is programmed to the value set by CURRent:TRIGgered when a triggered transient occurs.
PULSe	The current is changed to the value set by CURRent:TRIGgered for a duration determined by the pulse commands.
LIST	The current is controlled by the current list when a triggered transient occurs.

Command Syntax	[SOURce:]CURRent:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST
*RST Value	FIX
Examples	CURR:MODE LIST CURR:MODE FIX
Query Syntax	[SOURce:]CURRent:MODE?
Returned Parameters	<CRD>
Related Commands	CURR:TRG CURR

CURRent:SLEW

This command sets the slew rate for all programmed changes in the AC rms output current level of the AC source. A parameter of MAXimum or INFinity will set the slew to its maximum possible rate. The SCPI representation for INFinity is 9.9E37.

Command Syntax	[SOURce:]CURRent:SLEW[:IMMediate]<NRf+>|INFinity
Parameters	1E-3 to 9.9E37 | INFinity |MINimum | MAXimum
Unit	A/S (amps per second)
*RST Value	INFinity
Examples	CURR:SLEW 1 CURR:SLEW MAX CURR:SLEW INF
Query Syntax	[SOURce:]CURRent:SLEW[:IMMediate]?
Returned Parameters	<NR3>
Related Commands	CURR:SLEW:MODE CURR:SLEW:TRIG

CURRent:SLEW:MODE

Phase Selectable

This command determines how the output current slew rate is controlled during a triggered output transient. The choices are:

FIXed	The slew rate is unaffected by a triggered output transient.
STEP	The slew rate is programmed to the value set by VOLTage:SLEW:TRIGgered when a triggered transient occurs.
PULSe	The slew rate is changed to the value set by CURRent:SLEW:TRIGgered for a duration determined by the pulse commands.
LIST	The slew rate is controlled by the current slew list when a triggered transient occurs.

Command Syntax	[SOURce:]CURRent:SLEW:MODE<mode>
Parameters	FIXed | STEP | PULSe | LIST
*RST Value	FIX
Examples	CURR:SLEW:MODE LIST CURR:SLEW:MODE FIX
Query Syntax	[SOURce:]CURRent:SLEW:MODE?
Returned Parameters	<CRD>
Related Commands	CURR:SLEW:TRG CURR:SLEW

CURRent:SLEW:TRIGgered

Phase Selectable

This command selects the slew rate that will be set during a triggered step or pulse transient. A parameter of MAXimum or INFinity will set the slew to its maximum possible rate. The SCPI representation for infinity is 9.9E37.

Command Syntax	[SOURce:]CURRent:SLEW:TRIGgered<NRf+>|INFinity
Parameters	1E-3 to 9.9E37 | INFinity |MINimum | MAXimum
Unit	A/S (amps per second)
*RST Value	INFinity
Examples	CURR:SLEW:TRIG 1 CURR:SLEW:TRIG MAX CURR:SLEW:TRIG INF
Query Syntax	[SOURce:]CURRent:SLEW:TRIGgered?
Returned Parameters	<NR3>
Related Commands	CURR:SLEW:MODE CURR:SLEW

4.22 Status Subsystem Commands

This subsystem programs the CS Series AC Current source status registers. There are four groups of status registers; **Operation**, **Questionable**, **Questionable Instrument ISummary** and **Standard Event**. The Standard Event group is programmed with Common commands. The Operation, Questionable, and Instrument ISummary status groups each consist of the following five registers:

Condition	Enable	Event	NTR Filter	PTR Filter

Refer to Chapter 7 for more information about the status registers.

Subsystem Syntax

```
STATus
    :PRESet                             Presets all enable and transition registers to power-on
    :OPERation
        [:EVENt]                        Returns the value of the event register
        :CONDition                      Returns the value of the condition register
        :ENABle <n>                     Enables specific bits in the Event register
        :NTRansition<n>                 Sets the Negative transition filter
        :PTRansition<n>                 Sets the Positive transition filter
    :QUEStionable
        [:EVENt]                        Returns the value of the event register
        :CONDition                      Returns the value of the condition register
        :ENABle <n>                     Enables specific bits in the Event register
        :NTRansition<n>                 Sets the Negative transition filter
        :PTRansition<n>                 Sets the Positive transition filter
        :INSTrument
            :ISUMmary
                [:EVENt]                Returns the selected phase's event register value
                :CONDition              Returns the selected phase's condition register value
                :ENABle <n>             Enables specific bits in the selected phase's Event
                                        register
                :NTRansition<n>         Sets the selected phase's Negative transition filter
                :PTRansition<n>         Sets the selected Phase's Positive transition filter
```

STATus:PRESet

This command sets the Enable, PTR, and NTR registers of the status groups to their power-on values. These values are:

Enable Registers:	all bits set to 0 (OFF)
PTR Registers:	all defined bits set to 1 (ON)
NTR Registers:	all bits set to 0 (OFF)

Command Syntax	STATus:PRESet
Parameters	None
Examples	STAT:PRES

Bit Position	**15-9**	**8**	**7-6**	**5**	**4-1**	**0**
Bit Name	not used	CV	not used	WTG	not used	CAL
Bit Weight		256		32		1

Bit Position	15-9	8	7-6	5	4-1	0

CAL	Interface is computing new calibration constants
WTG	Interface is waiting for a trigger.
CV	Output voltage is regulated.

Table 4-3: Bit Configuration of Status Operation Registers

STATus:OPERation?

This query returns the value of the Operation Event register. The Event register is a read-only register which holds (latches) all events that are passed by the Operation NTR and/or PTR filter. Reading the Operation Event register clears it.

Query Syntax	STATus:OPERation[:EVENt]?
Parameters	None
Returned Parameters	<NR1>(Register Value)
Examples	STAT:OPER:EVEN?
Related Commands	*CLS STAT:OPER:NTR STAT:OPER:PTR

STATus:OPERation:CONDition?

This query returns the value of the Operation Condition register. This is a read-only register which holds the real-time (unlatched) operational status of the CS Series AC Current source.

Query Syntax	STATus:OPERation:CONDition?
Parameters	None
Examples	STAT:OPER:COND?
Returned Parameters	<NR1>(Register value)

STATus:OPERation:ENABle

This command and its query set and read the value of the Operation Enable register. This register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit (OPER) of the Status Byte register. The operation summary bit is the logical OR of all enabled Operation Event register bits.

Command Syntax	STATus:OPERation:ENABle <NRf+>
Parameters	0 to 32727
Default Value	0
Examples	STAT:OPER:ENAB 32 STAT:OPER:ENAB 1
Query Syntax	STATus:OPERation:ENABle?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:OPER:EVEN

STATus:OPERation:NTR STATus:OPERation:PTR

These commands set or read the value of the Operation NTR (Negative-Transition) and PTR (Positive-Transition) registers. These registers serve as polarity filters between the Operation Enable and Operation Event registers to cause the following actions:

- When a bit in the Operation NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set.
- When a bit of the Operation PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set.

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation Condition register sets the corresponding bit in the Operation Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation Condition register can set the corresponding bit in the Operation Event register.

Note: Setting a bit in the PTR or NTR filter can of itself generate positive or negative events in the corresponding Operation Event register.

Command Syntax	STATus:OPERation:NTRansition<NRf+> STATus:OPERation:PTRansition<NRf+>
Parameters	0 to 32727
Default Value	0
Examples	STAT:OPER:NTR 32 STAT:OPER:PTR 1
Query Syntax	STATus:OPERation:NTR? STATus:OPERation:PTR?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:OPER:ENAB

Bit Position	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8-5**	**4**	**3**	**2**	**1**	**0**
Bit Name	not used	not used	Isum	VL rms	Rail	not used	RI	not used	TO	UNR	not used	OCP	OV
Bit Weight			8192	4096	2048		512		16	8		2	1

Table 4-4: Bit Configuration of Questionable Registers

OV	over-voltage protection has tripped
OCP	over-current protection has tripped
UNR	output is unregulated
TO	over-temperature protection has tripped
RI	remote inhibit is active
Rail	loss of input phase detected
VL rms	rms voltage limit is active
Isum	summary of Isum registers

STATus:QUEStionable?

This query returns the value of the Questionable Event register. The Event register is a read-only register which holds (latches) all events that are passed by the Questionable NTR and/or PTR filter. Reading the Questionable Event register clears it.

On the CS Series AC Current source, each signal that is fed into the Questionable Status Condition register is logically-ORed from three corresponding status signals that originate from each phase.

Query Syntax	STATus:QUEStionable[:EVENt]?
Parameters	None
Returned Parameters	<NR1>(Register Value)
Examples	STAT:QUES:EVEN?
Related Commands	*CLS STAT:QUES:NTR STAT:QUES:PTR

STATus:QUEStionable:CONDition?

This query returns the value of the Questionable Condition register. That is a read-only register, which holds the real-time (unlatched) questionable status of the AC source.

Query Syntax	STATus:QUEStionable:CONDition?
Example	STAT:QUES:COND?
Returned Parameters	<NR1>(Register value)

STATus:QUEStionable:ENABle

This command sets or reads the value of the Questionable Enable register. This register is a mask for enabling specific bits from the Questionable Event register to set the questionable summary (QUES) bit of the Status Byte register. This bit (bit 3) is the logical OR of all the Questionable Event register bits that are enabled by the Questionable Status Enable register.

Command Syntax	STATus:QUEStionable:ENABle <NRf+>
Parameters	0 to 32727
Default Value	0
Examples	STAT:QUES:ENAB 18
Query Syntax	STATus:QUEStionable:ENABle?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:QUES:EVEN?

STATus:QUEStionable:NTR
STATus:QUEStionable:PTR

These commands allow the values of the Questionable NTR (Negative-Transition) and PTR (Positive-Transition) registers to be set or read. These registers serve as polarity filters between the Questionable Enable and Questionable Event registers to cause the following actions:

- When a bit of the Questionable NTR register is set to 1, then a 1-to-0 transition of the corresponding bit of the Questionable Condition register causes that bit in the Questionable Event register to be set.
- When a bit of the Questionable PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register.

Note: Setting a bit in the PTR or NTR filter can of itself generate positive or negative events in the corresponding Questionable Event register.

Command Syntax	STATus:QUEStionable:NTRansition<NRf+>
	STATus:QUEStionable:PTRansition<NRf+>
Parameters	0 to 32727
Default Value	0
Examples	STAT:QUES:NTR 16 STAT:QUES:PTR 512
Query Syntax	STATus:QUEStionable:NTRansition?
	STATus:QUEStionable:PTRansitiion?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:QUES:ENAB

Bit Position	15 13	12	11	10	9	8-5	4	3	2	1	0
Bit Name	not used	VL rms	Rail	not used	RI	not used	TO	UNR	not used	OCP	OV
Bit Weight		4096	2048		512		16	8		2	1

Table 4-5: Bit Configuration of Questionable Instrument Summary Registers

OV = over-voltage protection has tripped

OCP = over-current protection has tripped

UNR = output is unregulated

TO = over-temperature protection has tripped

RI = remote inhibit is active

Rail = loss of input phase voltage detected

VL rms = rms voltage limit is active

STATus:QUEStionable:INSTrument:ISUMmary?

Phase Selectable

This command returns the value of the Questionable Event register for a specific output of a three-phase AC source. The particular output phase must first be selected by INST:NSEL.

The Event register is a read-only register which holds (latches) all events that are passed by the Questionable NTR and/or PTR filter. Reading the Questionable Event register clears it.

Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary[:EVENt]?
Parameters	None
Returned Parameters	<NR1> (Register Value)
Examples	STAT:QUES:INST:ISUM:EVEN?
Related Commands	*CLS STAT:QUES:INST:ISUM:NTR STAT:QUES:INST:ISUM:PTR

STATus:QUEStionable:INSTrument:ISUMmary:CONDition?

Phase Selectable

This query returns the value of the Questionable Condition register for a specific output of a three-phase AC source. The particular output phase must first be selected by INST:NSEL.

The Condition register is a read-only register which holds the real-time (unlatched) questionable status of the CS Series AC Current source.

Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:CONDition?
Example	STAT:QUES:INST:ISUM:COND?
Returned Parameters	<NR1> (Register value)

STATus:QUEStionable:INSTrument:ISUMmary:ENABle

Phase Selectable

This command sets or reads the value of the Questionable Enable register for a specific output of a three-phase AC source. The particular output phase must first be selected by INST:NSEL.

The Enable register is a mask for enabling specific bits from the Questionable Event register to set the questionable summary (QUES) bit of the Status Byte register. This bit (bit 3) is the logical OR of all the Questionable Event register bits that are enabled by the Questionable Status Enable register.

Command Syntax	STATus:QUEStionable:INSTrument:ISUMmary:ENABle <NRf+>
Parameters	0 to 32767
Default Value	0
Examples	STAT:QUES:INST:ISUM:ENAB 18
Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:ENABle?
Returned Parameters	<NR1> (Register value)
Related Commands	STAT:QUES:INST:ISUM:EVEN?

STATus:QUEStionable:INSTrument:ISUMmary:NTR
STATus:QUEStionable:INSTrument:ISUMmary:PTR

These commands allow the values of the Questionable NTR (Negative-Transition) and PTR (Positive-Transition) registers to be set or read for a specific output of a three-phase AC source. The particular output phase must first be selected by INST:NSEL.

The NTR and PTR registers serve as polarity filters between the Questionable Enable and Questionable Event registers to cause the following actions:

- When a bit of the Questionable NTR register is set to 1, then a 1-to-0 transition of the corresponding bit of the Questionable Condition register causes that bit in the Questionable Event register to be set.
- When a bit of the Questionable PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register.

Note: Setting a bit in the PTR or NTR filter can of itself generate positive or negative events in the corresponding Questionable Event register.

Command Syntax	STATus:QUEStionable:INSTrument:ISUMmary :NTRansition<NRf> STATus:QUEStionable:INSTrument:ISUMmary :PTRansition<NRf>
Parameters	0 to 32727
Default Value	0
Examples	STAT:QUES:INST:ISUM:NTR 16 STAT:QUES:INST:ISUM:PTR 512
Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:NTRansition? STATus:QUEStionable:INSTrument:ISUMmary:PTRansition?
Returned Parameters	<NR1> (Register value)
Related Commands	STAT:QUES:INST:ISUM:ENAB

4.23 System Commands

The System Commands control the system-level functions of the CS Series AC Current source.

Subsystem Syntax

SYSTem	
:CONFiguration	Queries the source configuration data
:NOUTputs <n>	Selects the number of output phases
:ETIMe	Returns accumulated on time
:ERRor?	Returns the error number and error string
:LANGuage	Sets the programming language
:LOCal	Go to local mode (RS-232, USB, and LAN only)
:REMote	Go to remote mode (RS-232, USB and LAN only)
:RWLock	Local Lock out (front panel LOCAL button) (RS-232, USB and LAN only)
:TEMPerature?	Returns temperature inside unit.
:VERSion?	Returns the SCPI version number

SYSTem:CONFiguration

This query format of this command returns the installed options and configuration settings of the power source. It is not possible to change the configuration as only the query form is supported.

Command Syntax	SYSTem:CONFigure?
Query Syntax	SYSTem:CONFigure?
Returned Parameters	<AARD>
Related Commands	*OPT?

SYSTem:CONFiguration:NOUTputs

This command selects the number of outputs phases for the AC source. This requires that the AC source is capable of switching between single and three-phase mode.

Note: Execution of this command disables all outputs, clears lists and *RCL states to the initialization default values, reconfigures current readback and programming calibration constants.

Once configured, the instrument behaves transparently as either a single-phase source or as a three phase source depending on the selected configuration.

Command Syntax	SYSTem:CONFigure:NOUTputs <n>
Parameters	1 or 3
Examples	SYSTem:CONFigure:NOUT 3
Query Syntax	SYSTem:CONFigure:NOUT?
Returned Parameters	<NR1>
Related Commands	CAL:CURR

SYSTem:ETIMe?

This command will return the total number of accumulated hours, minutes and seconds.

Command Syntax	SYSTem:ETIMe?
Parameters	none
Example	SYST:ETIM?
Returned Parameters	<NR1>,<NR1>,<NR1>

SYSTem:ERRor?

This query returns the next error number followed by its corresponding error message string from the remote programming error queue. The queue is a FIFO (first-in, first-out) buffer that stores errors as they occur. As it is read, each error is removed from the queue. When all errors have been read, the query returns 0, No Error. If more errors are accumulated than the queue can hold, the last error in the queue is -350, Too Many Errors.

Query Syntax	SYSTem:ERRor?
Parameters	None
Returned Parameters	<NR1>,<SRD>
Example	SYST:ERR?

SYSTem:LANGuage

Sets the command language of the AC Power Source to either SCPI or an available alternate programming language such as APE (Abbreviated Plain English). The language selection is stored in non-volatile memory and is retained after power-off. Both the command and query form can be given regardless of the current language.

Command Syntax	SYSTem:LANGuage:<language>
Parameters	SCPI | APEN
Example	SYST:LANG:SCPI
Query Syntax	SYSTem LANGuage?
Returned Parameters	<CRD>

SYSTem:LOCal

This command can only be used with the RS-232 interface. It sets the interface in Local state, which enables the front panel controls. For GPIB, use the Local bus message, which controls the ATN control line.

Command Syntax	SYSTem:LOCal
Parameters	None
Example	SYST:LOC
Related Commands	SYST:REM SYST:RWL

SYSTem:REMote

This command can only be used with the RS-232, USB, and LAN interfaces. It sets the interface in the Remote state, which disables all front panel controls except the Local key. Pressing the Local key while in the Remote state returns the front panel to the Local state. For GPIB, use the Local bus message, which controls the ATN control line.

Command Syntax	SYSTem:REMotel
Parameters	None
Example	SYST:REM
Related Commands	SYST:LOC SYST:RWL

SYSTem:RWLock

This command can only be used with the RS-232, USB, and LAN interfaces. It locks out the Local key on the front panel. The SYST:LOC command must be used to unlock the front panel. For GPIB, use the Local Lockout bus message.

Command Syntax	SYSTem:RWLock
Parameters	None
Example	SYST:RWL
Related Commands	SYST:LOC

SYSTem:TEMPerature?

This command will return the internal ambient temperature of the power source in degrees Celsius.

Command Syntax	SYSTem:TEMPerature?
Parameters	none
Example	SYST:TEMP?

SYSTem:VERSion?

This query returns the SCPI version number to which the AC source complies. The returned value is of the form YYYY.V, where YYYY represents the year and V is the revision number for that year.

Query Syntax	SYSTem:VERSion?
Parameters	None
Returned Parameters	<NR2>
Example	SYST:VERS?

4.24 Trace Subsystem Commands

This subsystem programs the output waveform of the 3000CS and the 4500CS. Two waveform commands are available: TRACe and DATA. These commands are interchangeable; they both perform the same function.

Subsystem Syntax

```
TRACe | DATA
    :CATalog?                                          Return list of defined waveforms
    [:DATA] <waveform_name>, <n>  , <n>                Assign values to a waveform
    :DEFine <waveform_name>[, <waveform_name>|1024]    Create and name new waveform
    :DELete
        [:NAME] <waveform_name>                        Delete waveform to free its
                                                       memory
        :ALL                                           Deletes all user defined
                                                       waveforms at once.
```

TRACe
DATA

These commands set the values of a user-defined waveform table. The first parameter is the name of a waveform that was previously defined with TRACe:DEFine. Following the name are 1024 data points that define the relative amplitudes of exactly one cycle of the waveform. The first data point defines the relative amplitude that will be output at 0 degrees phase reference. An error will occur if exactly 1024 data points are not sent with the command. See section 6.9 for additional information on using the TRAC:DEF and TRAC:DATA commands.

Data points can be in any arbitrary units. The AC source scales the data to an internal format that removes the dc component and ensures that the correct ac rms current is output when the waveform is selected. When queried, trace data is returned as normalized values in the range of ±1. You can query the predefined SINusoid, SQUare, or CSINusoid waveform shapes, but you cannot use the predefined names as names for your waveform.

Waveform data is stored in nonvolatile memory and is retained when input power is removed. Up to 50 user-defined waveforms may be created and stored. The *RST and *RCL commands have no effect on user-defined waveforms. A waveform can be selected for output using the FUNCtion:SHAPe, FUNCtion:SHAPe:TRIGgered, or LIST:SHAPe commands.

Command Syntax:	TRACe[:DATA]<waveform_name>,<NRf> {,<NRf>}
	DATA[:DATA]<waveform_name>,<NRf>{,<NRf>}

Parameters	<waveform_name>, <amplitude>
Example	TRAC flattop,0.1,0.3,0.7,.....-0.7,-0.3,-0.1
Query Syntax:	TRACe[:DATA]?<waveform_name> DATA[:DATA]?<waveform_name>
Returned Parameters	<NR3> (a total of 1024 data points)
Related Commands	TRAC:DATA TRAC:DEL FUNC:SHAP

TRACe:CATalog
DATA:CATalog

These commands return a list of defined waveform names. The list includes both pre-defined waveforms such as SINusoid, SQUare, and CSINusoid, as well as any user-defined waveforms.

Query Syntax:	TRACe:CATalog? DATA:CATalog?		
Returned Parameters	<SRD>		
Example	TRAC:CAT?		
Related Commands	TRAC:DATA	TRAC:DEL	FUNC:SHAP

TRACe:DEFine
DATA:DEFine

These commands define a new waveform with the name <waveform_name> and allocates storage for its data. The waveform name can then be referenced by the TRACe:DATA command to define its data values. See section 6.9 for additional information on using the TRAC:DEF and TRAC:DATA commands.

An optional second argument is accepted for SCPI compatibility although it serves no useful purpose in the AC source. The second argument can be the name of an existing waveform, or the number of points in the trace. When a second name is sent, the data from the first waveform name is copied to the second. When the number of points in the trace is sent, only the number 1024 is accepted.

Command Syntax:	TRACe:DEFine <waveform_name> [, <waveform_name>|1024] DATA:DEFine <waveform_name> [, <waveform_name>|1024]		
Parameters	<waveform_name>		
Example	TRAC:DEF flattop		
Related Commands	TRAC:DATA	TRAC:DEL	FUNC:SHAP

TRACe:DELete
DATA:DELete

These commands delete the user-defined waveform table with the name <waveform_name> and makes its memory available for other waveforms.

Command Syntax:	TRACe:DELete[:NAME] <waveform_name> DATA:DELete[:NAME] <waveform_name>		
Parameters	<waveform name>		
Example	TRAC:DEL flattop		
Related Commands	TRAC:DATA	TRAC:DEL	FUNC:SHAP

TRACe:DELete:ALL
DATA:DELete:ALL

These commands delete all user-defined waveforms in the data table and makes the entire waveform memory available.

Command Syntax:	TRACe:DELete:ALL DATA:DELete:ALL		
Parameters	none		
Example	TRAC:DEL:ALL		
Related Commands	TRAC:DATA	TRAC:DEL	FUNC:SHAP

4.25 Trigger Subsystem

This subsystem controls the triggering of the AC source. See chapter 6.6 under Triggering Output Changes for an explanation of the Trigger Subsystem. The INITiate commands control the initialization of both the transient and measurement trigger systems.

The trigger subsystem must first be enabled using the INITiate commands or no triggering action will occur.

Subsystem Syntax

```
ABORt                                   Resets the trigger system to the Idle state
INITiate
    [:IMMediate]                        Initiates the system for one trigger
        :SEQuence[1|3]                  Initiates a specific numbered sequence
        :NAME TRANsient | ACQuire       Initiates a specific named sequence
    :CONTinuous
        :SEQuence[1] <bool>             Sets continuous initialization
        :NAME TRANsient <bool>          Sets continuous initialization
TRIGger
    [:SEQuence1 | :TRANsient]
        [:IMMediate]                    Triggers the output immediately
        :DELay <n>]                     Sets the trigger delay time
        :SOURce <source>                Sets the trigger source (BUS|EXT|IMM)
    :SEQuence2 | :SYNCronize
        :SOURce PHAse | IMMediate       Sets the synchronous source
    :PHASe <n>]                         Sets the synchronous phase reference
    :SEQuence3 | :ACQuire
        [:IMMediate]                    Triggers the measurement immediately
         :SOURce <source>               Sets the trigger source (BUS|EXT|TTLT)
    :SEQuence1
        :DEFine TRANsient               Sets or queries the SEQ1 name
    :SEQuence2
        :DEFine SYNChronize             Sets or queries the SEQ2 name
    :SEQuence3
        :DEFine ACQuire                 Sets or queries the SEQ3 name
```

ABORt

This command resets the measurement and transient trigger systems to the Idle state. Any output transient or measurement that is in progress is immediately aborted. ABORt also cancels any lists or pulses that may be in process.

ABORt also resets the WTG bit in the Operation Condition Status register. ABORt is executed at power turn-on and upon execution of *RCL, RST, or any implied abort command (see List Subsystem).

Note: If INITiate:CONTinuous ON has been programmed, the trigger subsystem initiates itself immediately after ABORt, thereby setting the WTG bit.

Command Syntax	ABORt
Parameters	None
Examples	ABOR
Related Commands	INIT *RST *TRG TRIG

INITiate:SEQuence
INITiate:NAME

The INITiate commands control the initiation of both the transient generator and the measurement trigger systems. They cause the trigger system to make a transition from the Idle state to the Waiting-for-Trigger state. If the trigger system is not in the Idle state, the initiate commands are ignored.

INITiate:SEQuence and INITiate:NAME initiate the trigger systems to reference trigger sequences. INITiate:SEQuence references a trigger sequence by its number, while INITiate:NAME references a sequence by its name. The correspondence between sequence names and numbers is:

Sequence Number	Sequence Name	Description
1(default)	TRANsient	Step, pulse, or list transient trigger sequence
3	ACQuire	Measurement acquire trigger sequence

Command Syntax	INITiate[:IMMediate]:SEQuence[1 | 3] INITiate[:IMMediate]:NAME<name>
Parameters	For INIT:NAME TRANsient | ACQuire
Examples	INIT:SEQ1 INIT:NAME ACQ
Related Commands	ABOR INIT:CONT TRIG *TRG

INITiate:CONTinuous:SEQuence
INITiate:CONTinuous:NAME

These commands control the transient generator trigger system.

- 1 or ON continuously initiates the transient trigger system.
- 0 or OFF turns off continuous triggering. In this state, the trigger system must be initiated for each triggered event using INITiate:SEQuence.

INITiate:CONTinuous:SEQuence references the transient trigger sequence by its number, while INITiate:CONTinuous:NAME references it by its name.

Command Syntax	INITiate:CONTinuous:SEQuence[1]<bool> INITiate:CONTinuous:NAME TRANsient,<bool>
Parameters	0|1|OFF|ON
*RST Value	OFF
Examples	INIT:CONT:SEQ ON INIT:CONT:NAME TRAN, 1
Related Commands	ABOR INIT:CONT TRIG *TRG

TRIGger

When the trigger subsystem has been initiated, the TRIGger command generates a trigger signal regardless of the selected trigger source.

Command Syntax	TRIGger[:SEQuence1][:IMMediate] TRIGger[:TRANsient][:IMMediate]
Parameters	None
Examples	TRIG TRIG:TRAN TRIG:SEQ1:IMM
Related Commands	ABOR TRIG:SOUR TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:DELay

This command sets the time delay between the detection of a trigger signal and the start of any corresponding trigger action. After the time delay has elapsed, the trigger is implemented unless the trigger system is also waiting for a sync signal that has been specified by TRIGger:SYNChronous:PHASe.

Command Syntax	TRIGger[:SEQuence1]:DELay<NRf+> TRIGger[:TRANsient]:DELay<NRf+>
Parameters	3-phase models: 0 to 1.07533E6|MINimum|MAXimum 1-phase models: 0 to 4.30133E5|MINimum|MAXimum
Unit	s (seconds)
*RST Value	0
Examples	TRIG:DEL .25 TRIG:DEL MAX TRIG:TRAN:DEL 1
Query Syntax	TRIGger[:SEQuence1]:DELay? TRIGger[:TRANsient]:DELay?
Returned Parameters	<NR3>
Related Commands	ABOR TRIG TRIG:SOUR TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SOURce

This command selects the trigger source for the first sequence in generating a step, pulse, or list output as follows:

BUS	IEEE-488 device, *TRG, or <GET> (Group Execute Trigger)
EXTernal	AC source's backpanel Trigger In1 SMA connector
IMMediate	trigger is generated as soon as the trigger system is initiated.

Command Syntax	TRIGger[:SEQuence1]:SOURce<CRD> TRIGger[:TRANsient]:SOURce<CRD>
Parameters	BUS | EXTernal | IMMediate
*RST Value	BUS
Examples	TRIG:SOUR BUS TRIG:TRAN:SOUR EXT
Query Syntax	TRIGger[:SEQuence1]:SOURce? TRIGger[:TRANsient]:SOURce?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SEQuence2:SOURce
TRIGger:SYNChronize:SOURce

These commands select the synchronizing trigger source in generating a step, pulse, or list output as follows:

IMMediate starts the transient output immediately, unless a delay time other than 0 has been specified by TRIGger:DELay. In this case the transient output starts after the expiration of the delay time.

PHASe starts the transient output at the reference phase set by TRIG:SYNC:PHAS.

Command Syntax	TRIGger:SEQuence2:SOURce<CRD>
	TRIGger:SYNChronize:SOURce<CRD>
Parameters	IMMediate | PHASe
*RST Value	IMM
Examples	TRIG:SYNC:SOUR IMM TRIG:SEQ2:SOUR PHAS
Query Syntax	TRIGger:SEQuence2:SOURce?
	TRIGger:SYNChronize:SOURce?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG:DEL TRIG:SYNC
	TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SEQuence2:PHASe
TRIGger:SYNCHronize:PHASe

These commands set the phase angle with respect to an internal phase reference at which PHASe:SYNChronous:SOURce becomes true. The range is from -360 to +360 degrees.

Command Syntax	TRIGger:SEQuence2:PHASe<NRf+>
	TRIGger:SYNChronize:PHASe<NRf+>
Parameters	-360° to +360°
*RST Value	0
Examples	TRIG:SYNC:PHAS 90 TRIG:SEQ2:PHAS 180
Query Syntax	TRIGger:SEQuence2:PHASe?
	TRIGger:SYNChronize:PHASe?
Returned Parameters	<NR3>
Related Commands	ABOR TRIG:DEL TRIG:SYNC
	TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SEQuence3
TRIGger:ACQuire

When the trigger subsystem has been initiated, these commands generate a measurement trigger regardless of the selected trigger source. The measurement trigger causes the AC source to digitize the instantaneous output current and voltage for several output cycles and store the results in a buffer.

The FETCh commands return the requested calculation from this acquired data. When the measurement completes, the WTG bit in the Status Operation Condition register is cleared.

Command Syntax	TRIGger:SEQuence3[:IMMediate] TRIGger:ACQuire[:IMMediate]
Parameters	None
Examples	TRIG:ACQ TRIG:SEQ3:IMM
Related Commands	ABOR TRIG:SOUR TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SEQuence3:SOURce
TRIGger:ACQuire:SOURce

These commands select the trigger source for a triggered measurement sequence as follows:

BUS	IEEE-488 device, *TRG, or <GET> (Group Execute Trigger)
EXTernal	backpanel Trigger In1 SMA connector
TTLTrg	the signal driving the Trigger Out1 SMA

Command Syntax	TRIGger:SEQuence3:SOURce<CRD>
Command Syntax	TRIGger:ACQuire:SOURce<CRD>
Parameters	BUS | EXT | TTLT
*RST Value	BUS
Examples	TRIG:ACQ:SOUR BUS TRIG:SEQ3:SOUR EXT
Query Syntax	TRIGger:SEQuence3:SOURce? TRIGger:ACQuire:SOURce?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG:SOUR TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

TRIGger:SEQuence1:DEFine
TRIGger:SEQuence2:DEFine
TRIGger:SEQuence3:DEFine

These commands define the names that are aliased to trigger sequences 1, 2 and 3. The command accepts only TRANsient for sequence 1, SYNChronous for sequence 2, and ACQuire for sequence 3 as predefined names. The query allows the user to query the instrument names aliased to sequences 1, 2, and 3.

Command Syntax	TRIGger:SEQuence1:DEFine TRANsient TRIGger:SEQuence2:DEFine SYNChronous TRIGger:SEQuence3:DEFine ACQuire
Parameters	TRANsient, SYNChronous, ACQuire
Examples	SEQ1:DEF TRAN SEQ3:DEF ACQ
Query Syntax	TRIGger:SEQuence1:DEFine? TRIGger:SEQuence2:DEFine? TRIGger:SEQuence3:DEFine?
Returned Parameters	<CRD>

5. Common Commands

Common commands begin with an * and consist of three letters (command) or three letters and a ? (query). Common commands are defined by the IEEE 488.2 standard to perform some common interface functions. The CS Series responds to the required common commands that control status reporting, synchronization, and internal operations. The CS also responds to optional common commands that control triggers, power-on conditions, and stored operating parameters.

Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data or status specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately. The description for each common command or query specifies any status registers affected. Refer to chapter 7 under Programming the Status and Event Registers which explains how to read specific register bits and use the information that they return.

Common Commands Syntax

*CLS	Clear status
*ESE <n>	Standard event status enable
*ESE?	Return standard event status enable
*ESR?	Return event status register
*IDN?	Return instrument identification
*OPC	Enable "operation complete" bit in ESR
*OPC?	Return a "1" when operation complete
*OPT?	Return option number
*PSC <bool>	Power-on status clear state set/reset
*PSC?	Return power-on status clear state
*RCL <n>	Recall instrument state
*RST	Reset
*SAV <n>	Save instrument state
*SRE <n>	Set service request enable register
*SRE?	Return service request enable register
*STB?	Return status byte
*TRG	Trigger
*TST?	Perform selftest, then return result
*WAI	Hold off bus until all device commands done

5.1 *CLS

This command clears the following registers (see chapter 7 under Programming the Status and Event Registers for descriptions of all registers):

- Standard Event Status
- Operation Status Event
- Questionable Status Event
- Status Byte
- Error Queue

Command Syntax	*CLS
Parameters	None

*ESE

This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event register (see *ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Event Register are logically ORed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. See section 7.5 for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

Bit Position	7	6	5	4	3	2	1	0
Bit Name	PON	not used	CME	EXE	DDE	QYE	not used	OPC
Bit Weight	128		32	16	8	4		1

Table 5-1: Bit Configuration of Standard Event Status Enable Register

CME	Command error	DDE	Device-dependent error
EXE	Execution error	OPC	Operation complete
PON	Power-on	QYE	Query error

Command Syntax	*ESE <NRf>
Parameters	0 - 255
Power-On Value	(See *PSC)
Example	*ESE 129
Query Syntax	*ESE?
Returned Parameters	<NR1>(Register value)
Related Commands	*ESR? *PSC *STB?

5.2 *ESR?

This query reads the Standard Event Status Event register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see *ESE). See section 7.5 for a detailed explanation of this register.

Query Syntax	*ESR?	
Parameters	None	
Returned Parameters	<NR1>(Register value)	
Related Commands	*CLS *ESE *ESE? *OPC	

5.3 *IDN?

This query requests the AC source to identify itself. It returns the data in four fields separated by commas.

Query Syntax	*IDN?	
Returned Parameters	<AARD>	
	Field	Information
	California Instruments	Manufacturer
	4500CS	10-digit model number
	12345	5-character serial number or 0
	<R>1.20	Revision levels of firmware
Example	CALIFORNIA INSTRUMENTS,4500CS,12345,1.20	

5.4 *OPC

This command causes the interface to set the OPC bit (bit 0) of the Standard Event Status register when the AC source has completed all pending operations. (See *ESE for the bit configuration of the Standard Event Status registers.) Pending operations are complete when:

- All commands sent before *OPC have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect output voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the CS Series AC Current source. The *OPC command provides notification that all overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the Idle state.

*OPC does not prevent processing of subsequent commands but Bit 0 will not be set until all pending operations are completed.

The query causes the interface to place an ASCII "1" in the Output Queue when all pending operations are completed.

Command Syntax	*OPC
Parameters	None
Query Syntax	*OPC?
Returned Parameters	<NR1>
Related Commands	*TRIG *WAI

5.5 *OPT?

This query requests the AC Source to identify any options that are installed. Options are identified by number. A 0 indicates no options are installed.

Query Syntax	*OPT?
Returned Parameters	<AARD>

5.6 *PSC

This command controls the automatic clearing at power-on of the Service Request Enable and the Standard Event Status Enable registers (see chapter 7 under Programming the Status and Event Registers for register details):

*PSC ON | 1	prevents the register contents from being saved causing them to be cleared at power-on. This prevents a PON event from generating SRQ at power-on.
PSC OFF | 0	saves the contents of the Standard Event Enable and Service Request Enable registers in nonvolatile memory and recalls them at power-on. This allows a PON event to generate SRQ at power-on.

Command Syntax	*PSC<bool>
Parameters	0 | 1 | OFF | ON
Example	*PSC 0 *PSC 1
Query Syntax	*PSC?
Returned Parameters	0 | 1
Related Commands	*ESE *SRE

5.7 *RCL

Warning **Recalling a previously stored state may place hazardous voltages at the AC source output.**

This command restores the AC source to a state that was previously stored in memory with a *SAV command to the specified location. All states are recalled with the following exceptions:

- CAL:STATe is set to OFF
- the trigger system is set to the Idle state by an implied ABORt command (this cancels any uncompleted trigger actions)

The device state stored in location 0 is automatically recalled at power turn-on when the OUTPut:PON:STATE is set to RCL0. This register only saves steady state setting, not the transient list data. All other registers (1-15) saved both the steady state settings and the transient list data record.

Command Syntax	*RCL <NRf>
Parameters	0 through 15
Example	*RCL 3
Related Commands	*PSC *RST *SAV

Note that recalling settings and transient list information from non-volatile memory takes some amount of time as shown in the table below. If you develop a test program that uses the *RCL or *SAV commands, insert the amount of delay shown in the table before sending the next command in your program.

Command:	*RCL 0	*RCL n (n = 1 to 15)	*SAV 0	*SAV n (n = 1 to 15)
Execution time:	20 msec	40 msec	100 msec	100 msec

5.8 *RST

This command resets the AC source to the factory-defined states shown in Table 5-2. This state applies if the PONS is set to use the RST state. If the PONS state is set to use REG0, the settings stored in setup register 0 take effect instead.

Item	Value	Item	Value
CAL:STAT	OFF	[SOUR:]FUNC:MODE	FIX
		[SOUR:]FUNC:TRIG	SIN
		[SOUR:]FUNC:CSIN	100%
		[SOUR:]LIST:COUN	1
		[SOUR:]LIST:STEP	AUTO

INIT:CONT	OFF	[SOUR:]PHAS	øA = 0°,
			øB = 240°,
			øC = 120°

INST:COUP	ALL	[SOUR:]PHAS:TRIG	øA = 0°,
			øB = 240°,
			øC = 120°

INST:NSEL	1	[SOUR:]PHAS:MODE	FIX
OUTP	OFF	[SOUR:]PULS:COUN	1
OUTP:DFI	OFF	[SOUR:]PULS:DCYC	50%
OUTP:DFI:SOUR	OFF	[SOUR:]PULS:HOLD	WIDT
OUTP:PROT:DEL	100ms	[SOUR:]PULS:PER	1
OUTP:RI:MODE	OFF	[SOUR:]PULS:WIDT	0.01667s
OUTP:TTLT	OFF	[SOUR:]CURR	0.000
OUTP:TTLT:SOUR	BOT	[SOUR:]CURR:TRIG	0.000
SENS:SWE:OFFS:POIN	0	[SOUR:]CURR:MODE	FIX
SENS:SWE:TINT	21µs		
SENS:WIND	Rectangular	[SOUR:]VOLT:SENS:SOUR	INT
[SOUR:]VOLT	135	[SOUR:]CURR:SLEW	INF
[SOUR:]VOLT:PROT:STAT	OFF	[SOUR:]CURR:SLEW:MODE	FIX
[SOUR:]FREQ	60Hz	[SOUR:]CURR:SLEW:TRIG	INF
[SOUR:]FREQ:MODE	FIX	SYST:CONF:NOUT	LAST
[SOUR:]FREQ:SLEW	INF	TRIG:DEL	0
[SOUR:]FREQ:SLEW:MODE	FIX	TRIG:SOUR	BUS
[SOUR:]FREQ:SLEW:TRIG	INF	TRIG:SEQ2:SOUR	IMM
[SOUR:]FREQ:TRIG	60Hz	TRIG:SEQ2:PHAS	0
[SOUR:]FUNC	SIN	TRIG:SEQ3:SOUR	BUS

*Table 5-2 : factory-defined *RST states*

Note ***RST does not clear any of the status registers or the error queue, and does not affect any interface error conditions. *RST does not affect the data in any of the lists.**
***RST sets the trigger system to the Idle state.**

Command Syntax	*RST	
Parameters	None	
Related Commands	*PSC	*SAV

5.9 *SAV

This command stores the present state of the AC source to a specified location in memory. Up to 16 setup states can be stored in nonvolatile memory. If a particular state is desired at power-on, it should be stored in location 0. It then will be recalled at power-on if the OUTPut:PON:STATe command is set to RCL0. Use *RCL to retrieve instrument states.

Command Syntax	*SAV
Parameters	0 to 15
Related Commands	*PSC *RCL *RST

Note that saving settings and transient list information from non-volatile memory takes some amount of time as shown in the table below. If you develop a test program that uses the *RCL or *SAV commands, insert the amount of delay shown in the table before sending the next command in your program.

Command:	*RCL 0	*RCL n (n = 1 to 15)	*SAV 0	*SAV n (n = 1 to 15)
Execution time:	20 msec	40 msec	100 msec	100 msec

5.10 *SRE

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (see *STB for its bit configuration) are allowed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in any Service Request Enable Register bit position enables the corresponding Status Byte Register bit and all such enabled bits then are logically ORed to cause Bit 6 of the Status Byte Register to be set. See chapter 7 under Programming the Status and Event Registers for more details concerning this process.

When the controller conducts a serial poll in response to SRQ, the RQS bit is cleared, but the MSS bit is not. When *SRE is cleared (by programming it with 0), the AC source cannot generate an SRQ to the controller.

Command Syntax	*SRE <NRf>
Parameters	0 to 255
Default Value	(See *PSC)
Example	*SRE 128
Query Syntax	*SRE?
Returned Parameters	<NR1>(Register binary value)
Related Commands	*ESE *ESR *PSC

5.11 *STB?

This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. Reading the Status Byte register does not clear it. The input summary bits are cleared when the appropriate event registers are read (see chapter 7 under Programming the Status and Event Registers for more information). A serial poll also returns the value of the Status Byte register, except that bit 6 returns Request for Service (RQS) instead of Master Status Summary (MSS). A serial poll clears RQS, but not MSS. When MSS is set, it indicates that the AC source has one or more reasons for requesting service.

Bit Position	7	6	5	4	3	2 - 0
Bit Name	OPER	MSS RQS	ESB	MAV	QUES	not used

Bit Position	7	6	5	4	3	2 - 0
Bit Weight	128	64	32	16	8	

Table 5-3: Bit Configuration of Status Byte Register

OPER =	operation status summary	MSS =	master status summary
ESB =	event status byte summary	RQS =	request for service
QUES =	questionable status summary	MAV =	message available

Query Syntax	*STB?
Returned Parameters	<NR1> (Register binary value)
Related Commands	*SRE *ESE *ESR

5.12 *TRG

This command generates a trigger to any subsystem that has BUS selected as its source (for example, TRIG:SOUR BUS).The command has the same affect as the Group Execute Trigger (<GET>) command.

Command Syntax	*TRG
Parameters	None
Related Commands	ABOR INIT TRIG[:IMM]

5.13 *TST?

This query causes the AC Source to do a self-test and report any errors. The selftest will run until the first error is encountered and terminate. The response to the query will either be the first error encountered or 0 is no error was found. (Selftest passed). For a list of error messages, refer to Appendix C.

Note: The selftest should always be run in 3 phase mode on all CS models. If the selftest is run in single phase mode, not all aspects of the Phase B and C hardware will be tested as a result.

Note: The selftest is always performed at 1000 Hz, 11.1 Arms. This is required to obtain the correct internal test voltage based on the buit-in selftest loads. However, the output relay is open during the selftest and no output is present on the output terminals during the selftest. At the end of the selftest, the output relay remains open and the current is set to 0A. The programmed frequency will return to the value set before execution of the selftest command.

Query Syntax	*TST?	
Returned Parameters	<NR1>	0 indicates the AC source has passed selftest. Nonzero indicates an error code.

5.14 *WAI

This command instructs the AC source not to process any further commands until all pending operations are completed. Pending operations are complete when:

- All commands sent before *WAI have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect output voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the AC source. The *WAI command prevents subsequent commands from being executed before any overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the Idle state.

*WAI can be aborted only by sending the an IEEE-488 DCL (Device Clear) command.

Command Syntax	*WAI
Parameters	None
Related Commands	*OPC

6. Programming Examples

6.1 Introduction

This chapter contains examples on how to program your AC source. Simple examples show you how to program:

- output functions such as current, frequency, and phase
- the transient waveform generator
- internal and external triggers
- measurement functions
- user-defined waveforms
- the status and protection functions

These examples in this chapter are generic SCPI commands. See chapter 2 for information about encoding the commands as language strings. Where appropriate, optional commands are shown for clarity in the examples.

6.2 Programming the Output

Power-on Initialization

When the AC source is first turned on, it wakes up with the output state set OFF. In this state the output voltage is set to 0. The following commands are given implicitly at power-on:

```
*RST
*CLS
STATus:PRESet
*SRE 0
*ESE 0
```

*RST is a convenient way to program all parameters to a known state. Refer to the *RST command in Section 5 to see how each programmable parameter is set by *RST. Refer to the *PSC command in Section 5 for more information on the power-on initialization of the *ESE and the *SRE registers.

Enabling the Output

To enable the output, use the command:

```
OUTPut ON
```

AC Current and Frequency

The AC rms output current is controlled with the CURRent command. For example, to set the AC output voltage to 0.125 amps rms, use:

```
CURR 0.125
```

All phases are programmed to 0.125 amps rms because the INSTrument:COUPle at *RST is set to ALL.

The AC source can be programmed to turn off its output if the AC output current exceeds a preset peak current limit. This protection feature is implemented with the CURRent:PROTection command as explained in chapter 4.

Maximum Current

The maximum rms output current that can be programmed can be queried with:

CURR? MAX

Frequency

The output frequency is controlled with the FREQuency command. To set the output frequency to 400 Hz, use:

FREQuency 400

Current and Frequency Slew Rates

Current Slew

The AC source has the ability to control the slew rate of AC amplitude and frequency changes. This can be used to generate ramps or to protect sensitive loads. To set the current slew rate to 20 amps per second, use:

CURRent:SLEW 20

At *RST the slew rate is set to INFinity, which means that AC current changes occur at the fastest possible slew rate. The slew rate applies to programmed changes in AC output amplitude while the unit is operating in fixed mode. Amplitude changes made by the step, pulse, and list transients are controlled by the same rules that apply to all other functions that are subject to transient control. See "Programming Transient Outputs".

Output current changes caused by the OUTPut:STATe command, by a protection feature disabling the output, or as a result of load changes are not subject to this slew rate control.

Frequency Slew

The AC source also has the ability to control the slew rate of frequency changes. To set the frequency slew rate to 30 Hz per second, use:

FREQuency:SLEW 30

At *RST the slew rate is set to INFinity, which means that frequency changes occur instantaneously. The frequency slew rate applies to programmed changes in frequency while the unit is operating in fixed mode. Frequency changes made by the step, pulse, and list transients are controlled by the same rules that apply to all other functions that are subject to transient control. See "Programming Transient Outputs".

6.2.1 Individual Phases

The following functions can be controlled separately in each phase:

- VOLTage
- CURRent
- PHASe
- MEASure
- FETCh
- CALibrate

Selecting a Phase

Two commands determine which output phase or phases receive commands in the three-phase mode. These are:

INSTrument:COUPle ALL | NONE

INSTrument:NSELect <n>

The *RST setting for INSTrument:COUPle is ALL. This setting causes programming commands to be sent to all output phases simultaneously.

To send a programming command to only one of the output phases, set INSTrument:COUPle to NONE, then select the desired output to receive the command with INSTrument:NSELect. For example, when the commands

INSTrument:COUPle NONE

INSTrument:NSELect 2

are sent, all subsequent voltage commands will go to output phase 2, and all measurement queries will return readings from output phase 2.

The INSTrument:COUPle command has no effect on queries. In the three-phase mode, queries are always directed to the output selected by INSTrument:NSELect.

Programming the Output Phase

You can control the phase of the AC current waveform relative to an internal reference with:

PHASe <n>

which sets the phase in degrees. If <n> is positive, the current waveform leads the internal reference.

The PHASe command sets the relative phase of each of the outputs. The INSTrument:COUPle setting is ignored by the PHASe command - it always controls the output selected by INSTrument:NSELect.

6.2.2 Voltage Limit

All models have a programmable rms voltage limit function. The command to set this limit is:

VOLTage<n>

where <n> is the rms voltage limit in volts.

If the load connected results in a voltage higher than the programmed limit, the output current is reduced to keep the rms voltage within the limit. Since the rms detection involves a filter time constant that is long compared to a single output cycle, the response time of the rms voltage limit is not instantaneous. When the output current is reduced, its waveform is preserved (the output waveform is attenuated, not clipped).

The AC source can be programmed to turn off its output if the rms voltage limit is reached. This protection feature is implemented with the VOLTage:PROTection:STATe command as explained in chapter 3.

6.3 Programming Output Transients

Output transients are used to:

- Synchronize output changes with a particular phase of the current waveform.
- Synchronize output changes with internal or external trigger signals.
- Simulate surge, sag, and dropout conditions with precise control of duration and phase.

- Create complex, multi-level sequences of output changes.
- Create output changes that have rapid or precise timing requirements.

The following AC source functions are subject to transient control:

- AC output current
- Frequency
- Phase
- AC current slew rate
- Frequency slew rate

The following transient modes can be generated:

Step generates a single triggered output change.

Pulse generates an output change which returns to its original state after some time period.

List generates a sequence of output changes, each with an associated dwell time or paced by triggers.

Fixed turns off the transient functions, which means that only the IMMediate values are used as the data source for a particular function.

At *RST all functions are set to FIXed, which turns off the transient functions.

6.3.1 Transient System Model

Figure 6-1 is a model of the transient system. The figure shows the transient modes and the source of the data that generates each mode. When a trigger is received in step or pulse modes, the triggered functions are set from their IMMediate to their TRIGgered value.

Note: In Step mode, the triggered value becomes the immediate value. In Pulse mode, the functions return to their immediate value during the low portion of the pulse.

If there are no further pulses, the immediate value remains in effect. In List mode, the functions return to their immediate value at the completion of the list.

You can mix FIXed, STEP, PULSe, and LIST modes among most functions. When a trigger is received, each function will react in a manner defined by its mode. However, this is subject to the following limitation to ensure the proper output voltage in all cases:

The AC current and current slew functions cannot be set to Step or Pulse mode if one of them is set to List mode.

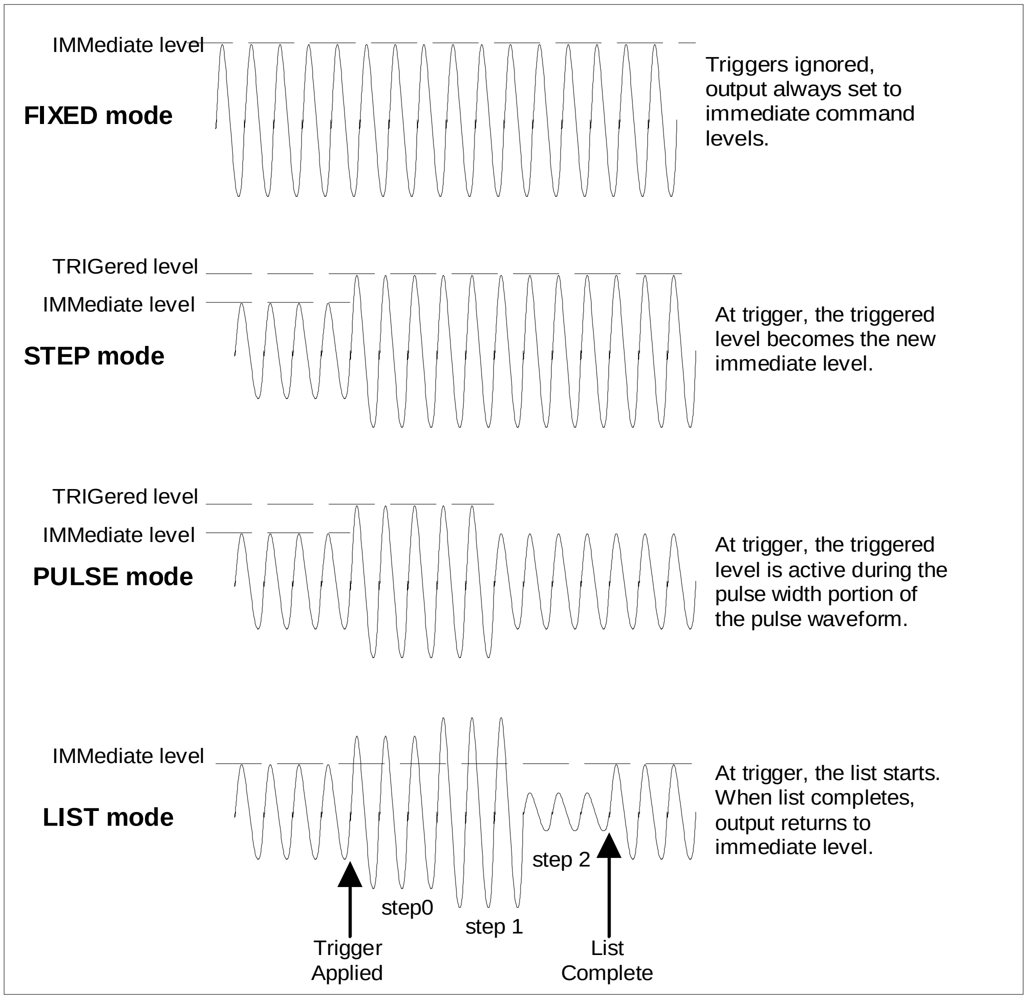

IMMediate level
FIXED mode
Triggers ignored, output always set to immediate command levels.
TRIGered level
IMMediate level
STEP mode
At trigger, the triggered level becomes the new immediate level.
TRIGered level
IMMediate level
PULSE mode
At trigger, the triggered level is active during the pulse width portion of the pulse waveform.
IMMediate level
LIST mode
At trigger, the list starts. When list completes, output returns to immediate level.
step0
step 1
step 2
Trigger Applied
List Complete

Figure 6-1: Model of transient system.

6.4 Step and Pulse Transients

Proceed as follows to setup step and pulse transients:

1. Set the functions that you do **not** want to generate transients to FIXed mode. A convenient way to do this is with the *RST command. Then set the mode of the function that will generate the transient to STEP or PULSe as required. For example, to enable the voltage function to generate a single triggered output voltage change, use:

   **RST*

   CURRent:MODE STEP

2. Set the triggered level of the function that will generate the transient. For example, if the previously programmed current function is going to step the output current amplitude to 0.135 amps upon receipt of a trigger, use:

   CURRent:TRIGger 0.135

3. Select the trigger source that will generate the trigger. For example, to select the external Trigger In1 SMA connector as the trigger source, use:

   TRIGger:TRANsient:SOURce EXTernal

Trigger sources are discussed in detail under "Triggering Output Changes".

4. Only perform this step if you have selected PULSE as the transient mode in Step 1. Specify the pulse count, the pulse period, and then either the duty cycle or the pulse width using the following commands:

• PULSe:COUNt 1	specifies 1 output pulse
• PULSe:PERiod 1	specifies a pulse period of 1 second
• PULSe:DCYCle 50	specifies a duty cycle of 50%
• PULSe:WIDTh .5	specifies a pulse width of .5 seconds (not necessary in this case since a duty cycle has already been specified)

5. Initiate the transient trigger system to enable it to receive a trigger. To enable the trigger system for one transient event use:

   INITiate:IMMediate:SEQuence1

6. To enable the transient system indefinitely use:

   INITiate:CONTinuous:SEQuence1 ON

   Trigger the transient. This is described in detail under "Triggering Output Changes".

Example

The following example programs a current dropout for 2 cycles of a 0.5 amps, 400 Hz output. The dropout begins at the positive peak of the output current waveform (90 degrees phase) and is triggered by IEEE-488 bus trigger.

*RST	Begin at power-on state
CURR 0.5	Set initial output current (immediate-level)
FREQ 400	Set initial output frequency
OUTP ON	Enable the output
CURR:MODE PULS	Enable output to generate pulses when triggered
CURR:TRIG 0	Set the current dropout (triggered level)
PULS:WIDT .03333	Set pulse width for 2 periods
TRIG:SOUR BUS	Respond to IEEE-488 bus triggers
TRIG:SYNC:SOUR PHAS	Synchronize triggers to internal phase reference
TRIG:SYNC:PHAS 90	Sets internal phase reference point to 90 degrees
INIT:SEQ1	Set to Wait-for-trigger state
<device trigger>	Send the IEEE-488 bus trigger

6.5 List Transients

List mode lets you generate complex sequences of output changes with rapid, precise timing, which may be synchronized with internal or external signals. Each function that can participate in output transients can also have an associated list of values that specify its output at each list point.

You can program up to 100 settings (or points) in the list, the time interval (dwell) that each setting is maintained, the number of times that the list will be executed, and how the settings change in response to triggers.

All list point data is stored in nonvolatile memory. This means that the programmed data for any list function will be retained when the AC source is turned off.

Lists are paced by a separate list of dwell times which define the duration of each output setting. Therefore, each of the up to 100 list points has an associated dwell time, which specifies the time (in seconds) that the output remain at that setting before moving on to the next setting.

The following procedure shows how to generate a simple list of current and frequency changes:

1. Set the mode of each function that will participate in the output sequence to LIST. For example:

   CURRent:MODE LIST

   FREQuency:MODE LIST

2. Program the list of output values for each function. The list commands take a comma-separated list of arguments. The order in which the arguments are given determines the sequence in which the values will be output. For example, to cycle the voltage through a sequence, a list may include the following values:

   LIST:CURR 0.120, 0.132, 0.108, 0.120, 0.132, 0.108, 0.120, 0.132, 0.108

   You can specify lists for more than one function. For example, to synchronize the previous current list with another list that varies the output frequency from nominal, to high, to low, the lists may include the following values:

   LIST:CURR 0.120, 0.132, 0.108, 0.120, 0.132, 0.108, 0.120, 0.132, 0.108

   LIST:FREQ 400, 400, 400, 440, 440, 440, 360, 360, 360

   All lists must have the same number of data values or points, or an error will occur when the transient system that starts the sequence is later initiated. The exception is when a list has only one item or point. In this case the single-item list is treated as if it had the same number of points as the other lists, with all values being equal to the one item. For example:

   LIST:CURR 0.110, 0.120, 0.130;FREQuency 400

   is the same as:

   LIST:CURR 0.110, 0.120, 0.130

   LIST:FREQuency 400, 400, 400

3. Determine the time interval that the output remains at each level or point in the list before it advances to the next point. The time is specified in seconds. For example, to specify five dwell intervals, use:

   LIST:DWELL 1, 1.5, 2, 2.5, 3

   The number of dwell points must equal the number of output points. If a dwell list has only one value, that value will be applied to all points in the output list.

4. Determine the number of times the list is repeated before it completes. For example, to repeat a list 10 times use:

   LIST:COUNt 10

   Entering INFinity makes the list repeat indefinitely. At *RST, the count is set to 1.

5. Determines how the list sequencing responds to triggers. For a closely controlled sequence of output levels, you can use a dwell-paced list. To cause the list to be paced by dwell time use:

   LIST:STEP AUTO

   As each dwell time elapses, the next point is immediately output. This is the *RST setting.

   If you need the output to closely follow asynchronous events, then a trigger-paced list is more appropriate. In a trigger-paced list, the list advances one point for each trigger received. To enable trigger-paced lists use:

   LIST:STEP ONCE

   The dwell time associated with each point determines the minimum time that the output remains at that point. If a trigger is received before the previous dwell time completes, the trigger is ignored. Therefore, to ensure that no triggers are lost, program the dwell time to zero.

6. Use the transient trigger system to trigger the list. This is described in detail under "Triggering Output Changes" (Section 6.6).

6.6 Triggering Output Changes

The AC source has two independent trigger systems. One is used for generating output changes, and the other is used for triggering measurements. This section describes the output trigger system. The measurement trigger system is described under "Triggering Measurements".

The basic components of both systems are the same, but the transient trigger system has additional delay and phase synchronization features that the measurement trigger system does not have.

The following transient trigger sources can be selected:

IMMediate	generates a trigger when the trigger system is initiated.
BUS	selects IEEE-488 bus triggers.
EXTernal	selects the external Trigger In1 SMA connector.

6.6.1 SCPI Triggering Nomenclature

In SCPI terms, trigger systems are called sequences. When more than one trigger system exists, they are differentiated by naming them SEQuence1, SEQuence2, ... etc. In the AC source, SEQuence1 is the transient trigger system, SEQuence2 is the phase synchronization trigger system, and SEQuence3 is the measurement trigger system.

The AC source uses aliases with more descriptive names for these sequences. These aliases can be used instead of the sequence forms.

Sequence Form	Alias
SEQuence1	TRANsient
SEQuence2	SYNChronize
SEQuence3	ACQuire

6.6.2 Output Trigger System Model

Figure 6-2 is a model of the output trigger system. The rectangular boxes represent states. The arrows show the transitions between states. These are labeled with the input or event that causes the transition to occur.

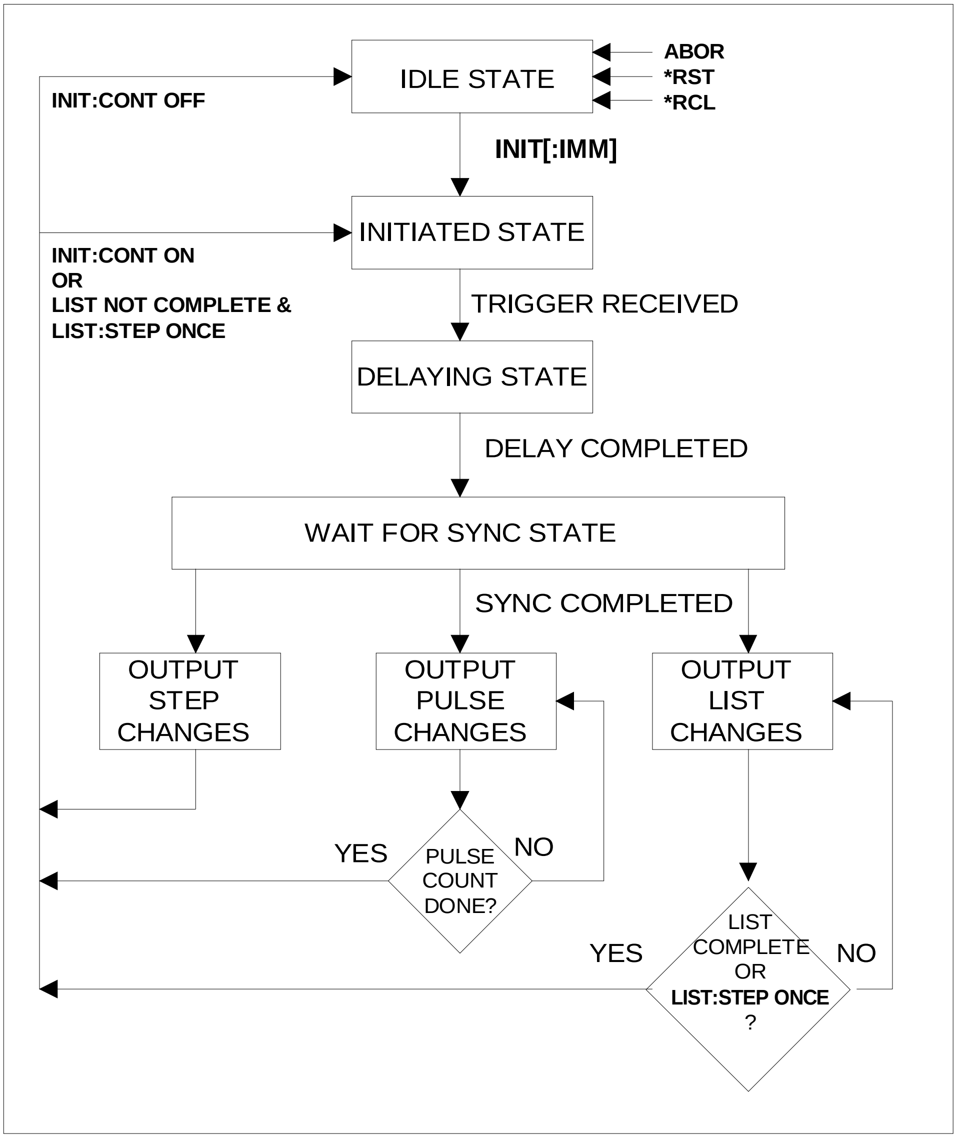


Figure 6-2: Model of output trigger system.

6.6.3 Initiating the Output Trigger System

When the AC source is turned on, the trigger subsystem is in the idle state. In this state, the trigger subsystem ignores all triggers. Sending the following commands at any time returns the trigger system to the Idle state:

*ABORt   *RST   *RCL*

The INITiate commands move the trigger system from the Idle state to the Initiated state. This enables the AC source to receive triggers. To initiate for a single triggered action, use:

INITiate:IMMediate:SEQuence1 or

INITiate:IMMediate:NAME TRANsient

After a trigger is received and the action completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered action is desired. To keep a trigger system initiated for multiple actions without having to send an initiate command for each trigger, use:

INITiate:CONTinuous:SEQuence1 ON or

INITiate:CONTinuous:NAME TRANsient, ON

The SEQuence2 (or SYNChronize) trigger sequence does not have an INITiate command. It is always initiated.

6.6.4 Selecting the Output Trigger Source

The trigger system is waiting for a trigger signal in the Initiated state. Before you generate a trigger, you must select a trigger source. To select the external Trigger In1 SMA as the source, use:

TRIGger:SEQuence1:SOURce EXTernal or

TRIGger:TRANsient:SOURce EXTernal

To select IEEE-488 bus triggers (group execute trigger, device trigger, or *TRG command), use:

TRIGger:SEQuence1:SOURce BUS or

TRIGger:TRANsient:SOURce BUS

To select a trigger source that is always true, use:

TRIGger:SEQuence1:SOURce IMMediate or

TRIGger:TRANsient:SOURce IMMediate

The immediate source can be combined with INITiate:CONTinuous:SEQuence1 ON to generate repetitive output transients. A transition from the Initiated state to the Delay state is made when the trigger signal is received.

6.6.5 Specifying a Trigger Delay

A time delay can be programmed between the receipt of the trigger signal and the start of the output transient. At *RST the trigger delay is set to 0, which means that there is no delay. To program a delay, use:

TRIGger:SEQuence1:DELay .01 or

TRIGger:TRANsient:DELay .01

which sets a delay time of 10 milliseconds. A trigger delay can only be programmed for SEQuence1 (or TRANsient) triggers. When the programmed trigger delay has elapsed, the trigger system transitions from the Delay state to the Wait-for-sync state.

6.6.6 Synchronizing Output Changes to a Reference Phase Angle

An output transient normally occurs immediately when the trigger signal is received, or after the delay has expired if a trigger delay has been set. For some applications it is desirable that the transient is synchronized with a particular phase of the output waveform such as the zero crossing point or the positive peak.

To synchronize the start of a transient with a particular phase of the internal phase reference, you must select PHASE as the trigger source. Use:

TRIGger:SEQuence2:SOURce PHASe or

TRIGger:SYNChronize:SOURce PHASe

To select the desired phase, use:

TRIGger:SEQuence2:PHASe 90 or

TRIGger:SYNChronize:PHASe 90

which specifies the 90 degree phase angle of the internal phase reference as the point where the transient begins.

To turn off transient phase synchronization, use:

TRIGger:SYNChronous:SOURce IMMediate

When IMMediate is selected, the trigger system transitions through the Delaying and Wait-for-sync states and goes directly to the Output state. This is the parameter selected at *RST.

6.6.7 Generating Triggers

Providing that you have specified the appropriate trigger source, you can generate triggers as follows:

Single Triggers

- By sending one of the following over the IEEE-488:
  - TRIGger:IMMediate
  - *TRG
  - a group execute trigger
- By applying a signal with a high-to-low transition to the Trig In1 SMA connector.

Continuous Triggers

- By sending the following commands over the IEEE-488:

TRIGger:SEQuence1:SOURce IMMediate

INITiate:CONTinuous:SEQuence1 ON

When the trigger system enters the Output Change state upon receipt of a trigger (see Figure 6-2), the triggered functions are set to their programmed trigger levels. When the triggered actions are completed, the trigger system returns to the Idle state.

6.6.8 Specifying a Dwell Time for Each List Point

Each frequency and current list point has an associated dwell time specified by:

LIST:DWELl <n> {,<n>}

where <n> specifies the dwell time in seconds. The number of dwell points must equal the number of output points. If a dwell list has only one value, that value will be applied to all points in the output list. After each new output level or point is programmed, the output remains at that point in the list for the programmed dwell interval before the list advances to the next point. Only an ABORt command can transfer the system out of the Dwelling state.

At the end of the dwell interval, the transition to the next state depends on whether or not the list has completed its sequencing and the state of the LIST:STEP command (see Figure 6-2).

- If the list is completed, the trigger system returns to the Idle state.
- If the list is not completed, then the system reacts as follows:

LIST:STEP ONCE	programs the trigger system to return to the Initiated state to wait for the next trigger.
LIST:STEP AUTO	programs the trigger system to immediately execute the next list point.

6.7 Making Measurements

The AC source has the capability to return a number of current, voltage, and power measurements. When the AC source is turned on, it is continuously sampling the instantaneous output voltage and current for several output cycles and writing the results into a buffer. The buffer holds 4096 voltage and current data points.

The AC source uses the data from the voltage and current buffer to calculate the requested measurement information. Data in the voltage and current buffers is always re-acquired for subsequent measurement requests. There are two ways to make measurements:

- Use the MEASure commands to immediately start acquiring new voltage and current data, and return measurement calculations from this data as soon as the buffer is full. This is the easiest way to make measurements, since it requires no explicit trigger programming.
- Use an acquisition trigger to acquire the voltage and current data from the buffer. Then use the FETCh commands to return calculations from the data that was retrieved by the acquisition trigger. This method gives you the flexibility to synchronize the data acquisition with an external signal. FETCh commands do not trigger the acquisition of new measurement data, but they can be used to return many different calculations from the data that was retrieved by the acquisition trigger.

Making triggered measurements with the acquisition trigger system is discussed under "Triggering Measurements".

Note **For each MEASure form of the query, there is a corresponding query that begins with the header FETCh. FETCh queries perform the same calculation as their MEASure counterparts, but do not cause new data to be acquired. Data acquired by an explicit trigger or a previously programmed MEASure command are used.**

6.7.1 Voltage and Current Measurements

To read the sum of AC and DC components of the rms voltage or current, use:

MEASure:VOLTage:ACDC? or

MEASure:CURRent:ACDC?

To measure the DC voltage or current components, use:

MEASure:VOLTage:DC? or

MEASure:CURRent:DC?

To measure the maximum current amplitude, use:

MEASure:CURRent:AMPLitude:MAXimum?

6.7.2 Power Measurements

The MEASure and FETCh queries can return real, apparent, and reactive power measurements as well as DC power and power factor using the following commands:

MEASure:POWer:AC:APParent?	measures the AC component of apparent power in VA
MEASure:POWer:AC:REACtive?	measures the reactive power
MEASure:POWer:AC:REAL?	measures the in-phase component of power in watts
MEASure:POWer:AC:PFACtor?	returns the output power factor
MEASure:POWer:AC:TOTal?	measures the total real power being sourced
MEASure:POWer:DC?	measures the DC component of power

6.7.3 Harmonic Measurements

To return the percentage of total harmonic distortion in the output voltage or current, use the following commands:

MEASure:CURRent:HARMonic:THD?

MEASure:VOLTage:HARMonic:THD?

6.7.4 Simultaneous Output Phase Measurements

You can return simultaneous measurements from all output phases of the source in the three phase mode using the FETCh query. Unlike MEASure queries, FETCh queries do not trigger the acquisition of new data when they are executed. First, you must initiate the measurement trigger system and generate a measurement trigger as explained in the following section "Triggering Measurements". When the measurement data has been acquired by the voltage and current data buffers for each output phase, use INSTrument:NSELect to select each phase, and FETCh to return the specified measurement data. The following commands return rms voltage:

INSTrument:NSELect 1

FETCh:VOLTage:AC?

INSTrument:NSELect 2

FETCh:VOLTage:AC?

INSTrument:NSELect 3

FETCh:VOLTage:AC?

6.7.5 SCPI Measurement Triggering Nomenclature

As previously explained under "Triggering Output Changes", the AC source uses the following sequence name and alias for the measurement trigger system. This alias can be used instead of the sequence form.

Sequence Form	Alias
SEQuence3	ACQuire

6.7.6 Measurement Trigger System Model

Figure 6-3 is a model of the measurement trigger system. The rectangular boxes represent states. The arrows show the transitions between states. These are labeled with the input or event that causes the transition to occur.

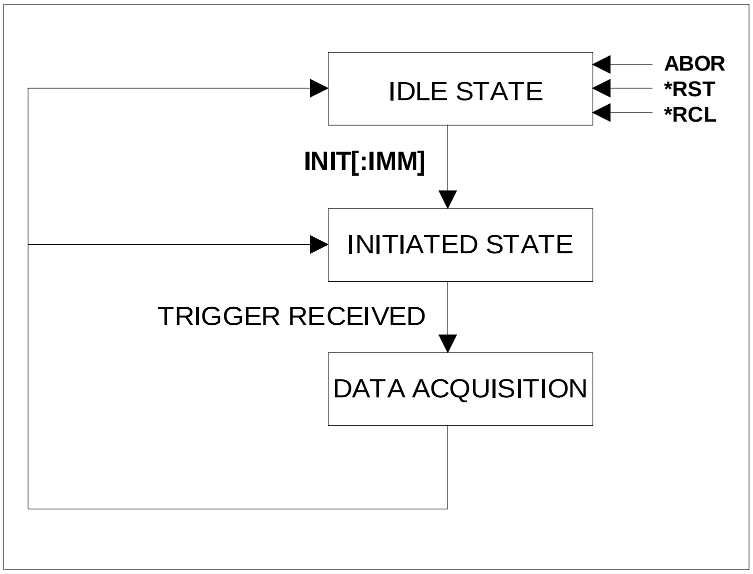


Figure 6-3: Model of Measurement triggers.

6.7.7 Initiating the Measurement Trigger System

When the AC source is turned on, the trigger system is in the idle state. In this state, the trigger system ignores all triggers. Sending the following commands at any time returns the trigger system to the Idle state:

*ABORt   *RST   *RCL*

The INITiate commands move the trigger system from the Idle state to the Initiated state. This enables the AC source to receive triggers. To initiate for a measurement trigger, use:

INITiate:IMMediate:SEQuence3   or

INITiate:IMMediate:NAME ACQuire

After a trigger is received and the data acquisition completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered acquisition is desired.

Note: You cannot initiate measurement triggers continuously. Otherwise, the measurement data in the data buffer would continuously be overwritten by each triggered measurement.

6.7.8 Selecting the Measurement Trigger Source

The trigger system is waiting for a trigger signal in the Initiated state. Before you generate a trigger, you must select a trigger source. To select the external Trigger In1 SMA as the source, use:

TRIGger:SEQuence3:SOURce EXTernal   or

TRIGger:ACQuire:SOURce EXTernal

To select IEEE-488 bus triggers (group execute trigger, device trigger, or *TRG command), use:

TRIGger:SEQuence3:SOURce BUS or

TRIGger:ACQuire:SOURce BUS

To select the signal driving the Trigger Out1 SMA connector, use:

TRIGger:SEQuence3:SOURce TTLTrg or

TRIGger:ACQuire:SOURce TTLTrg

6.7.9 Generating Measurement Triggers

Providing that you have specified the appropriate trigger source, you can generate triggers as follows:

- By sending one of the following over the IEEE-488:
  - TRIGger:SEQuence3:IMMediate
  - TRIGger:ACQuire:IMMediate
  - *TRG
  - a group execute trigger
- By applying a signal with a high-to-low transition to the Trig In1 SMA connector.
- By generating an output transient that causes the Trig Out1 SMA connector to output a pulse.

6.7.10 Using the DFI Output to Indicate Error Conditions

The Discrete Fault Indicator output on the rear of the CS Series AC Current source unit can be used to provide a control or status signal to an external device. The following example illustrates the use of the DFI output to signal a current limit fault condition.

The DFI output is the floating collector and emitter of an opto isolator. As such, these have to be terminated somehow. One possible way to terminate this signal is to tie FLT high to INH high and FLT low to INH low; this works as long as you are not using the INH inputs.

Next, the OVP detector has be to activated by sending the following command:

```
VOLT:PROT:STATE ON
```

To enable the DFI circuit, send:

```
OUTP:DFI:SOURCE QUES
OUTP:DFI:STATE ON
```

To enable the OVP signal to get to the STAT:QUES register, send:

```
STAT:QUES:PTR 2
STAT:QUES:ENAB 2
```

After sending these commands, the DFI output will indicate when the RMS OVP goes active. Note that when an OVP fault happens, the PROT lamp comes on indicating that one has to send OUTP:PROT:CLEAR to re-enable the source once the over voltage condition is removed. Note also that the DFI output will stay low until the questionable status register (STAT:QUES?) is read.

6.8 Controlling the Instantaneous Voltage and Current Data Buffers

Measurements taken by the CS Series measurement systems are based on a digital representation of the voltage and current waveforms. Both waveforms are digitized with 24 bits of amplitude resolution and down to 10.4 or 31.25 µsec of time resolution. (96 Ks/sec real-time sampling rate in single-phase / three-phase mode). The available memory depth to hold this information is 4K (4096 samples) for each waveform. This section covers the SCPI commands that can be used to control the sampling rate and the position of the trigger point in the 4K deep data buffer.

6.8.1 Retrieving Measurement Acquistion Data

Data can be retrieved by sending the “MEAS:ARR:VOLT?” or “MEAS:ARR:CURR?” queries. The data returned will be in either binary or ASCII format depending on the state of the “MEAS:ARR:MOD” command. The sample below assumes the binary data mode is used and only 1K samples are requested from the measurement data buffer.

```
Public Sub QueryMeasBuffer(ByRef Buffer() As Single)
'=============================================================
'Sample uses fixed 1K block size (4 sections)
'=============================================================
Dim i,j             As Integer
Dim iBlkSize        As Integer
Dim iMaxPoints      As Integer
Dim iPointer        As Integer
Dim s1              As String
Dim gl_bAsciiTransfer as boolean
'=============================================================
    'Initialize Size
    gl_bAsciiTransfer = True
    iBlkSize = 4
    iMaxPoints = iBlkSize * 256
    'Add block size of iBlkSize and offset of 0 parameters to measurement command
    cmd = "FETCH:ARR:VOLT? " + Cstr(iBlkSize) + ",0"
    'Allocate buffer space
    s1 = Space(4 * (iBlkSize * 256) + 6) 'Allow 6 extra bytes for #46384
    'Send query command using GPIB I/F Ctrlr 0
    SendGPIB 0, cmd
    ReceiveGPIB 0, AC_GPIB_Adr, s1, STOPend
    'Process received data
    If Len(s1) Then
      'Start at first char after header #44096
      If Left$(s1, 1) = "#" Then
        iPointer = Val(Mid$(s1, 2, 1))
        'Add 2 for #n and 1 to point to first actual data byte
        iPointer = iPointer + 3
        For i = 0 To iMaxPoints
          s = ""
          For j = 0 To iBlkSize - 1
            s = s + Mid$(s1, iPointer + j, 1)
          Next j
          iPointer = iPointer + iBlkSize
          Buffer(i) = StringToIEEEFloat(s, gl_bAsciiTransfer)
        Next i
      End If
    End If
End Sub
```

The required data format conversion from IEEE floating point to single precision floating points is shown in the next section.

6.8.2 VB6 Sample: Converting IEEE format trace data to floating point

Converting the received IEEE floating point format waveform data to a single precision floating point value can be accomplished using the following sample routine:

```
Public Function StringToIEEEFloat(ByVal sData As String, ByVal bAsciiMode As Boolean) As
Single
'==============================================================
'bAsciiMode flag is used if data is received as 8 ascii chars
'representing Hex 0-9,A-F. If bAsciiMode flag is false, then
'data is process as 4 char representing a byte each. Ascii
'mode is needed for DCBS windows
'==============================================================
  Dim i         As Integer
  Dim j         As Integer
  Dim iChar     As Integer
  Dim expo      As Long
  Dim mantisse  As Long
  Dim expo_val  As Variant
  Dim mant_f    As Single
  Dim c(3)      As Long    'Must use 32 bit integers to allow for
                      'intermediate result of 24 bit shift
  Dim sign      As Boolean
'==============================================================
Const MANT_MAX = &H7FFFFF
Const EXPO_MAX = 2 ^ 126
'==============================================================

On Error GoTo FloatConvError
  If bAsciiMode Then
    'Retrieve ASC values from eight hex byte input data
    sData = UCase(sData)
    For i = 0 To 3
      c(i) = 0
      For j = 0 To 1
        iChar = AscB(Mid$(sData, i * 2 + j + 1, 1)) - 48
        If iChar > 9 Then iChar = iChar - 7
        c(i) = c(i) * 16 * j + iChar
      Next j
    Next i
  Else
    'Retrieve ASC values from four byte input data
    'Note: Don't use ASCB or ASCW functions as results will differ
    'based on character sets, even on non DCBS Windows
    'Retrieve ASC values from four byte input data
    For i = 0 To 3
      c(i) = Asc(Mid$(sData, i + 1, 1))
    Next i
  End If
  'Get sign bit
  sign = ((c(0) And &H80) = &H80)
  'Get exponent value less sign bit
  expo = (c(0) And &H7F) * 2
  'Pick up exponent sign
  If (c(1) And &H80) = &H80 Then expo = expo Or 1
  'get data less exponent sign bit
  c(1) = c(1) And &H7F
  mantisse = c(1) * &H10000 + c(2) * &H100 + c(3)
  mant_f = mantisse / MANT_MAX
  'Process exponent
  If (expo <> 0) And (expo <> &HFF) Then
    expo = expo - 127
    mant_f = mant_f + 1
    expo_val = 2 ^ Abs(expo)
    If (expo > 0) Then mant_f = mant_f * expo_val
    If (expo < 0) Then mant_f = mant_f / expo_val
  Else
    If (mant_f <> 0) Then
      If expo = 0 Then
        mant_f = mant_f / EXPO_MAX
      Else
        mant_f = mant_f * EXPO_MAX
      End If
    End If
  End If
  'Append number sign and return value
  If sign Then mant_f = -mant_f
  StringToIEEEFloat = mant_f
Exit Function
'==============================================================
FloatConvError:
  'Conversion errors are truncated to zero
  StringToIEEEFloat = 0
  Exit Function

End Function
```

6.8.3 Varying the Voltage and Current Sampling Rate

At *RST, the output voltage and current sampling rate is 96 kHz (period = 10.4 μsec). This means that it takes about 43 milliseconds to fill up 4096 data points in the voltage and current data buffers with the information required to make a measurement calculation. You can vary this data sampling rate with:

SENSe:SWEep:TINTerval <sample period>

The sample period can be programmed from a minimum period of 10 microseconds (the default), to 100 microseconds in 10 microsecond steps. In the three phase mode, the three phases are multiplexed so the minimum sample period becomes 31.2 microseconds. The maximum sample period in three phase mode is 312 microseconds.

6.8.4 Pre-event and Post-event Triggering

The range for this offset is 4096 to 2E9 points. As shown in the following figure, when the offset is negative, the values at the beginning of the data record represent samples taken prior to the trigger. When the value is 0, all of the values are taken after the trigger. Values greater than zero can be used to program a delay time from the receipt of the trigger until the data points that are entered into the buffer are valid.

(Delay time = Offset × Sample period)

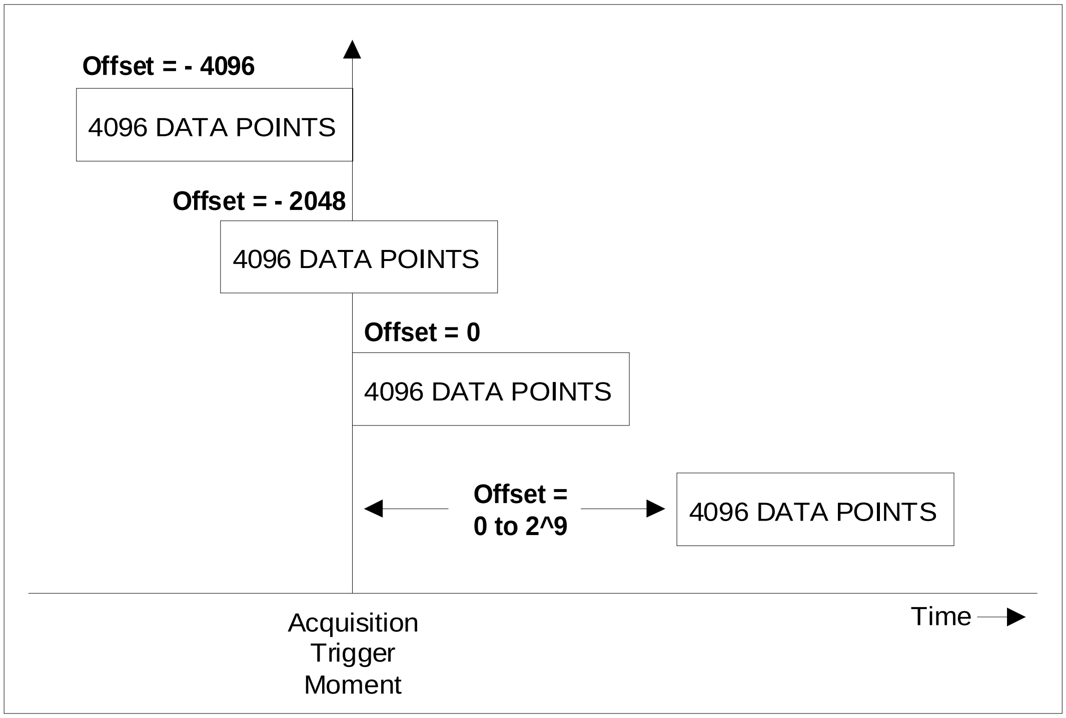


Figure 6-4: Pre- and Post Event Triggering.

6.9 Downloading Arbitrary Waveforms

The CS Series supports the use of arbitrary waveforms. Up to 50 user defined waveforms can be downloaded to the CS by assigning a user defined name of up to 12 characters long and sending the formatted waveform data over the bus. File names longer than 12 characters will be truncated on the WAVEFORM LCD display and make it difficult to select the right waveform from the front panel, especially if the first 12 characters are the same.

Note: Avoid using names that conflict with standard waveforms such as SIN, SINE, CLIPPED, SQUARE.

Once downloaded, user waveforms can be recalled from either the front panel using the MENU key or over the bus using the FUNCtion command. See section 4.17.

6.9.1 Waveform Record Data Format

The total number of datapoints required to define a waveform register is 1024. Each data points should be scaled between –1.000 and +1.000. If scaled differently, the CS controller will automatically scale the data received. The TRAC:DATA command is used to send the data in comma delimited form to the power source controller.

The datapoints correspond to a single waveform period. When selected for output, the waveform register will be output at the programmed frequency. Each register should be set up to contain a single period of the desired output waveform only. It is important to ensure that the start and end point of the waveform data matches (equal value). If not, a discontinuity will occur each time the register repeats at the output while selected.

6.9.2 Download Sample

The Visual Basic 6 code sample below provides a simple illustration on how to download a user defined waveform. The data points are passed in a single dimension array sWaveformPoints of type Single. (single precision floating point).

Note: User defined waveforms are retained in non-volatile memory. The process of writing to this memory takes a certain amount of time. Allow a 500 msec delay between sending the last data value and sending the next bus command.

Sample code:

```
Global sWaveformPoints(1023) As Single       'Waveform datapoints, base 0

Private Sub Download(ByVal strWaveName As String, ByVal iPoints As Integer, _
                     ByRef sBuffer() As Single)
'==========================================================================
'This procedure transfers a block of Data from the PC
'to the AC Source. The waveform name is passed as a parameter
'==========================================================================
Dim i As Integer
Dim strCmd As String
'==========================================================================
On Error GoTo XFRArbError
    ‘Define catalog entry user defined name (max length is 12 char)
    strCmd = "TRAC:DEF " + Left$(strWaveName,12)
    Send gl_iNIBrdId, AC_Adr, s, NLend
    ‘Send data buffer
    strCmd =  = "TRAC:DATA " + Left$(strWaveName,12)
    ‘Assemble data buffer string command
    For i = 0 to iPoints –1
```

```
      strCmd = strCmd + "," + FrmOut$(sBuffer(i), "0.0###")   'Format 4 digits
    Next i
    Send gl_iNIBrdId, AC_Adr, strCmd, Nlend
    ‘Insert 500 msec delay to allow update of Flash memory
    msec_delay 500
    Exit Sub

XFRArbError:
    MsgBox Error(Err), vbOKOnly + vbInformation, PROGNAME


End Sub
```

6.10 Command Processing Times

Most commands are processed immediately after they are received in the order they have been received. Some commands however take a long time to complete. During this time, the power source may not be able to accept additional commands. This issue must be considered when developing application programs. In situations like this, time must be allowed through a time delay or by interfacing to other instruments in the test system while the power source is allowed to finish.

The *OPC 1 and *OPC? Commands can be used to determine if a command has been completed. However, some commands will not accept the *OPC? Query either if the processor is busy. In this case, a delay must be allowed before the *OPC? Query or the next command is sent.

The following table lists those commands and their required processing times for reference.

Command	**Processing Time**	**Comment**
*SAV n	100 msec	Save settings and transient list data to non volatile memory. Flash memory access times are longer.
TRAC:DATA	500 msec	Save waveform data to non volatile memory. Flash memory access times are longer.
OUTP	Default: 100 msec	This output relay and range change relay settling time delays can be set to value up to 1000 msec using the PONS:REL:HOLD command. See section 4.13, “Power On Subsystem”. Adjust delay to value set as needed.
VOLT:RANG	Default: 100 msec	This output relay and range change relay settling time delays can be set to value up to 1000 msec using the PONS:REL:HOLD command. See section 4.13, “Power On Subsystem”.
SYST:CONF:NOUT	100 msec	Phase mode change. Switching between single or three phase output configuration requires reinitialization of controller.
*RST	1 sec	Softboot reset of controller requires reinitialization.
*TST?	10 sec	Selftest execution time.
CAL:XXXX	50 msec	Save output coeficient calibration data to non volatile memory. Flash memory access times are longer.
CAL:SAVE	300 msec	Save calibration data to non volatile memory. Flash memory access times are longer.

Table 6-1: Command Processing Times.

7. Programming the Status and Event Registers

You can use status register programming to determine the operating condition of the AC source at any time. For example, you may program the AC source to generate an interrupt (assert SRQ) when an event such as a current limit occurs. When the interrupt occurs, your program can then act on the event in the appropriate fashion.

Figure 7-1 shows the status register structure of the AC Source. Table 7-2 defines the status bits. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard IEEE-488 functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The Operation Status, Questionable Status, and Questionable Instrument Isummary Status registers implement functions that are specific to the AC source.

7.1 Power-On Conditions

Refer to the *RST command description in chapter 4, Table 5-2 for the power-on conditions of the status registers.

7.2 Operation Status Group

The Operation Status registers record signals that occur during normal operation. The group consists of the following registers:

Register	Command	Description
Condition	STAT:OPER:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:OPER:PTR <n>	A positive transition filter that functions as described under STAT:OPER:NTR|PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:OPER:NTR <n>	A negative transition filter that functions as described under STAT:OPER:NTR|PTR commands in chapter 4. It is a read/write register.
Event	STAT:OPER:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:OPER:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

Table 7-1: Operation Status registers

The outputs of the Operation Status register group are logically-ORed into the OPER(ation) summary bit (7) of the Status Byte register.

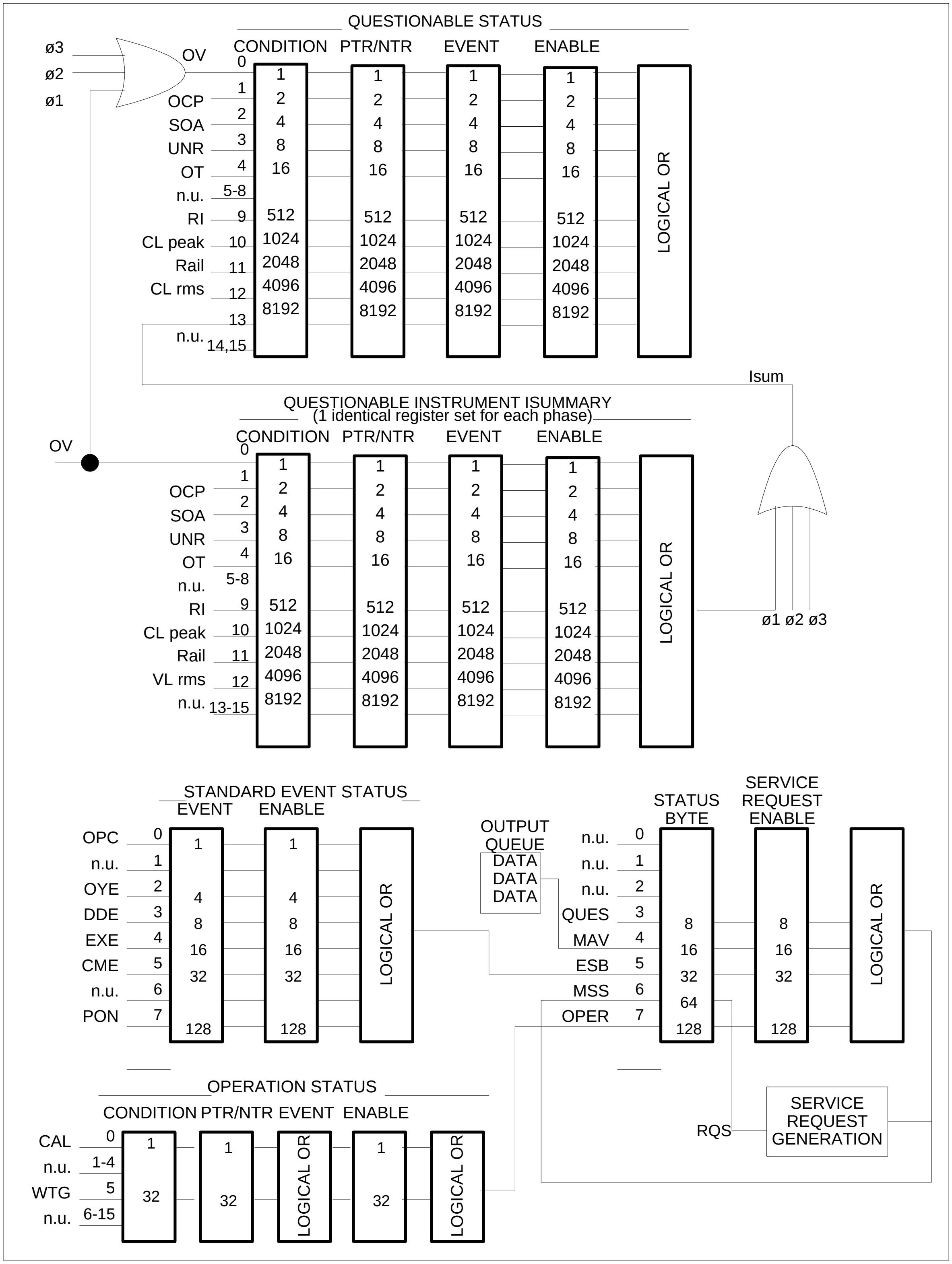


Figure 7-1: Status Register Model.

Bit	Signal	Meaning
		Operation Status Group
0	CAL	Interface is computing new cal constants
5	WTG	Interface is waiting for a trigger
8	CV	The output voltage is regulated
		Questionable and Questionable Instrument Isummary Status Groups
0	OV	The overvoltage protection circuit has tripped
1	OCP	The overcurrent protection circuit has tripped
3	UNR	The output is unregulated
4	TO	An overtemperature condition has occurred
9	RI	The remote inhibit state is active
11	Rail	Loss of input phase voltage detected
12	VLrms	The voltage limit circuit is active
13	Isum	Summary of QUES:INST:ISUM registers
		Standard Event Status Group
0	OPC	Operation complete
2	QYE	Query error
3	DDE	Device-dependent error
4	EXE	Execution error
5	CME	Command error
7	PON	Power-on
		Status Byte and Service Request Enable Registers
3	QUES	Questionable status summary bit
4	MAV	Message Available summary bit
5	ESB	Event Status Summary bit
6	MSS	Master Status Summary bit
	RQS	Request Service bit
7	OPER	Operation status summary bit

Table 7-2: Bit Configurations of Status Registers

7.3 Questionable Status Group

The Questionable Status registers record signals that indicate abnormal operation of the AC source. As shown in Figure 7-1, the group consists of the same type of registers as the Status Operation group.

Register	Command	Description
Condition	STAT:QUES:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:QUES:PTR <n>	A positive transition filter that functions as described under STAT:QUES:NTR|PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:QUES:NTR <n>	A negative transition filter that functions as described under STAT:QUES:NTR|PTR commands in chapter 4. It is a read/write register.
Event	STAT:QUES:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:QUES:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

Table 7-3: Questionable Status registers

The outputs of the Questionable Status group are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register.

Note: If the AC source is in the three-phase mode, each signal that is fed into the Questionable Status Condition register is logically-ORed from three corresponding status signals that originate from each phase. Figure 7-1 illustrates this for the OV bit; the same illustration also applies to the other bits in the Condition register.

7.4 Questionable Instrument Isummary Status Group

Although only one group of Questionable Instrument Isummary Status registers is shown in Figure 7-1, there are actually three identical register groups. With the exception of bit 13, the register structure is the same as the Questionable Status group. These three register groups monitor the status signals of each individual phase of the three-phase AC source. To determine which phase of the AC source is currently selected, use:

INSTrument:NSELect?

To set or read the status registers of another phase, first use:

INSTrument:NSELect <n>

where <n> is the phase number. Then send the appropriate register commands.

Register	Command	Description
Condition	STAT:QUES:INST:ISUM:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:QUES:INST:ISUM:PTR <n>	A positive transition filter that functions as described under STAT:QUES:INST:ISUM:NTR|PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:QUES:INST:ISUM:NTR <n>	A negative transition filter that functions as described under STAT:QUES:INST:ISUM:NTR|PTR commands in chapter 4. It is a read/write register.
Event	STAT:QUES:INST:ISUM:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:QUES:INST:ISUM:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

Table 7-4: Questionable Instrument Isummary Status registers

The outputs of the Questionable Instrument Isummary Status group are logically-ORed into the Isum bit (13) of the Questionable Condition register.

7.5 Standard Event Status Group

This group consists of an Event register and an Enable register that are programmed by Common commands. The Standard Event register latches events relating to interface communication status (see Figure 7-1). It is a read-only register that is cleared when read.

The Standard Event Enable register functions similarly to the enable registers of the Operation and Questionable status groups.

Command	Action
*ESE	programs specific bits in the Standard Event Enable register.
*PSC ON	clears the Standard Event Enable register at power-on.
*ESR?	reads and clears the Standard Event register.

The PON (Power On) Bit

The PON bit in the Standard Event register is set whenever the AC source is turned on. The most common use for PON is to generate an SRQ at power-on following an unexpected loss of power. To do this, bit 7 of the Standard Event Enable register must be set so that a power-on event registers in the ESB (Standard Event Summary Bit). Bit 5 of the Service Request Enable register must be set to permit an SRQ to be generated, and *PSC OFF must be sent. The commands to accomplish these conditions are:

```
*PSC OFF
*ESE 128
*SRE 32
```

7.6 Status Byte Register

This register summarizes the information from all other status groups as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The bit configuration is shown in Table 5-3.

Command	Action
*STB?	reads the data in the register but does not clear it (returns MSS in bit 6)
serial poll	reads and clears the data in the register (returns RQS in bit 6)

The MSS Bit

This is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the AC source has one or more reasons for requesting service. *STB? reads the MSS in bit position 6 of the response but does not clear any of the bits in the Status Byte register.

The RQS Bit

The RQS bit is a latched version of the MSS bit. Whenever the AC source requests service, it sets the SRQ interrupt line true and latches RQS into bit 6 of the Status Byte register. When the controller does a serial poll, RQS is cleared inside the register and returned in bit position 6 of the response. The remaining bits of the Status Byte register are not disturbed.

The MAV bit and Output Queue

The Output Queue is a first-in, first-out (FIFO) data register that stores AC source-to-controller messages until the controller reads them. Whenever the queue holds one or more bytes, it sets the MAV bit (bit 4) of the Status byte register.

7.7 Examples

7.7.1 Determining the Cause of a Service Interrupt

You can determine the reason for an SRQ by the following actions:

1. Determine which summary bits are active. Use

   **STB?* or
   serial poll

2. Read the corresponding Event register for each summary bit to determine which events caused the summary bit to be set. Use:

   STATus:QUEStionable:EVENt?

   STATus:OPERation:EVENt?

   ESR?

Note: When an Event register is read, it is cleared. This also clears the corresponding summary bit.

3. Remove the specific condition that caused the event. If this is not possible, the event may be disabled by programming the corresponding bit of the status group Enable register or NTR|PTR filter. A faster way to prevent the interrupt is to disable the service request by programming the appropriate bit of the Service Request Enable register.

7.7.2 Servicing Questionable Status Events

This example assumes you want a service request generated whenever the AC source's overvoltage, overcurrent, or overtemperature circuits have tripped. From Figure 7-1, note the required path for Questionable Status conditions at bits 0, 1, and 4 to generate a service request (RQS) at the Status Byte register. The required register programming is as follows:

1. Program the Questionable Status PTR register to allow a positive transition at bits 0, 1, or 4 to be latched into the Status Event register. Use:

   STATus:QUEStionable:PTR 19 (1 + 2 + 16 = 19)

2. Program the Questionable Status Enable register to allow the latched events to be summed into the QUES summary bit. Use:

   STATus:QUEStionable:ENABle 19

3. Program the Service Request Enable register to allow the QUES summary bit from the Status Byte register to generate RQS. Use:

   **SRE 8*

4. When you service the request, read the event register to determine which Questionable Status Event register bits are set and clear the register for the next event. Use:

   STATus:QUEStionable:EVENt?

7.7.3 Monitoring Both Phases of a Status Transition

You can monitor a status signal for both its positive and negative transitions. For example, to generate RQS when the AC source either enters the CLrms (rms current limit) condition or leaves that condition, program the Questionable Status PTR/NTR filter as follows:

STATus:QUEStionable:PTR 4096;NTR 4096

*STATus:QUEStionable:ENABle 4096;*SRE 8*

The PTR filter will cause the QUES summary bit to set RQS when CLrms occurs. When the controller subsequently reads the event register with STATus:QUEStionable:EVEN?, the register is cleared. When CLrms subsequently goes false, the NTR filter causes the QUES summary bit to again set RQS.

7.7.4 Programming the Trigger In and Trigger Out SMA connectors

The AC source has two SMA connectors labeled Trigger In and Trigger Out (see Figure 7-2). Refer to specifications in the User's Manual for the electrical parameters.

Trigger In SMA

This chassis-referenced digital input can be selected as a source for transient or measurement triggers. This allows an action to be synchronized to an external signal. The trigger is recognized on a high-to-low transition of the input signal. The minimum pulse width of the signal is 1 microsecond. To select the Trigger In connector as the trigger source, use:

TRIGger:SEQuence1:SOURce EXTernal or
TRIGger:TRANsient:SOURce EXTernal

TRIGger:SEQuence3:SOURce EXTernal or
TRIGger:ACQuire:SOURce EXTernal

Trigger Out SMA (Function Strobe)

This chassis-referenced digital output can be programmed to supply a pulse output at the leading or trailing edge of a step or pulse, or at the leading edge of any point in a list sequence. The output signal is nominally a 10 microsecond low-true pulse. To enable the Trigger Out connector, use:

OUTPut:TTLTrg:STATe ON

At *RST, the Trigger Out connector is off.

To select a trigger source for the Trigger Out connector, use:

OUTPut:TTLTrg:SOURce BOT | EOT | LIST

BOT -	specifies that the pulse is output at the beginning of a transient. This is the *RST setting.
EOT -	specifies that the pulse is output at the end of a transient.
LIST -	specifies that the pulse position is defined by the LIST:TTLTrg command.

You can also specify the Trigger Out connector as a trigger source for measurement trigger sequences. Use:

TRIGger:SEQuence3:SOURce TTLT or
TRIGger:ACQuire:SOURce TTLT

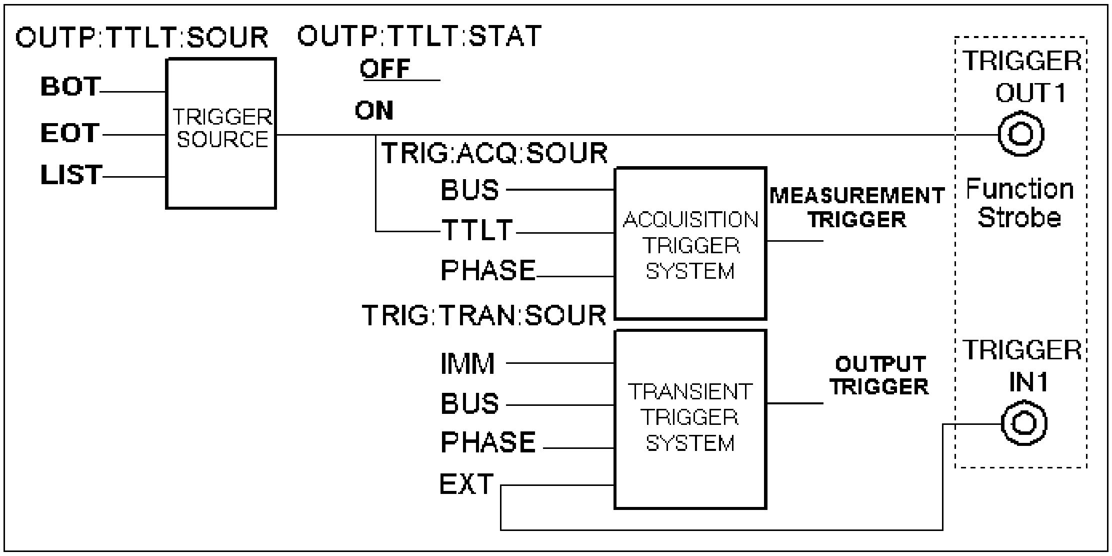


Figure 7-2: SMA Connector Trigger Model.

7.8 Remote Inhibit and Discrete Fault Indicator

The remote inhibit and discrete fault indicators are implemented through the respective INH and FLT connections on the rear panel. Refer to the User's Manual for the electrical parameters.

7.8.1 Remote Inhibit (RI)

Remote inhibit is an external, chassis-referenced logic signal routed through the rear panel INH connection, which allows an external device to signal a fault.

To select an operating modes for the remote inhibit signal, use:

OUTPut:RI:MODE LATChing | *LIVE* | *OFF*

LATChing	causes a low-true signal on the INH input to disable the output. The only way to clear the latch is by sending an OUTPut:PROTection:CLEAR command while the INH input is false.
LIVE	allows the RI input to disable the output in a non-latching manner. When INH is low true, the output is disabled. When INH is high, it has no effect on the output.
OFF	disables the INH input.

7.8.2 Discrete Fault Indicator (DFI)

The discrete fault indicator is a chassis-referenced, open-collector logic signal connected to the rear panel FLT connection, that can be used to signal external devices when a fault condition is detected. To select the internal fault source that drives this signal, use:

OUTPut:DFI:SOURce QUEStionable | *OPERation* | *ESB* | *RQS* | *OFF*

QUEStionable	selects the Questionable event summary bit (bit 3 of the Status Byte Register)
OPERation	selects the Operation Event summary bit (bit 7 of the Status Byte Register)
ESB	selects the Standard Event summary bit (bit 5 of the Status Byte Register)
RQS	selects the Request Service bit (bit 6 of the Status Byte Register)
OFF	selects no DFI source

To enable or disable the DFI output, use:

OUTPut:DFI:STATe ON | *OFF*

7.9 SCPI Command Completion

SCPI commands sent to the AC source are processed either sequentially or in parallel. Sequential commands finish execution before a subsequent command begins. Parallel commands allow other commands to begin executing while the parallel command is still executing. Commands that affect list and trigger actions are among the parallel commands.

The *WAI, *OPC, and *OPC? common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. The syntax and parameters for these commands are described in chapter 4. Some practical considerations for using these commands are as follows:

*WAI This prevents the AC source from processing subsequent commands until all pending operations are completed.

*OPC? This places a 1 in the Output Queue when all pending operations have completed. Because it requires your program to read the returned value before executing the next program statement, *OPC? can be used to cause the controller to wait for commands to complete before proceeding with its program.

*OPC This sets the OPC status bit when all pending operations have completed. Since your program can read this status bit on an interrupt basis, *OPC allows subsequent commands to be executed.

8. Option Commands

8.1 Introduction

The 4500CS is a specific configuration AC Current Source. As such, there are no options required or available.

The CS Series AC Current source will report any installed options, including any firmware options, when queried using the SYST:CONF:BYTE? or *OPT? command..

Appendix A : SCPI Command tree

```
SCPI Common Commands
    *CLS
    *ESE <value>
    *ESE?
    *ESR?
    *IDN?
    *OPC
    *OPC?
    *OPT?
    *PSC <bool>
    *PSC?
    *RCL <value>
    *RST
    *SAV <value>
    *SRE <value>
    *SRE?
    *STB?
    *TRG
    *TST?
    *WAI

ABORt

CALibrate
    :MEAS
        :CURRent
            SPHase
        :VOLTage
    :PASSword
    :PHASe
    :SAVE
    [:SOURce]
        :VOLTage

DIAGnostic
    :RESet
    :TEMPerature:AMBient?

INITiate
    [:IMMediate]
        :SEQuence[1 | 3]
        :NAME TRANsient | ACQuire
    :CONTinuous
        :SEQuence[1] <bool>
        :NAME TRANsient, <bool>

INSTrument
    :COUPle ALL | NONE
    :NSELect 1 | 2 | 3
    :SELect OUTPut1 | OUTPut2 | OUTPut 3

LIMit
    :CURRent
    :FREQuency:HIGH
    :FREQuency:LOW
    :PHASe
    :VOLTage:HIGH
    :VOLTage:LOW

MEASure | FETCh
    [:SCALar]
        :CURRent
            [:DC]?
            :AC?
            :ACDC?
            :HARMonic
                [:AMPLitude]? <n>
                :PHASe? <n>
                :THD?
            :NEUTral
                [:DC]?
                :AC?
                :ACDC?
                :HARMonic
                    [:AMPLitude]? <n>
                    :PHASe? <n>
        :FREQuency?
        :POWer
            [:DC]?
            :AC
                [:REAL]?
                :APParent?
                :REACtive?
                :PFACtor?
                :TOTal?
        :VOLTage
            [:DC]?
            :AC?
            :ACDC?
            :AMPLitude
                :MAX?
                :RESet
            :CREStfactor?
            :HARMonic
                [:AMPLitude]? <n>
                :PHASe? <n>
                :THD?
    :THD:MODE RMSQ | FUND

OUTPut
    [:STATe] <bool>
    :DFI
        [:STATe] <bool>
        :SOURce QUES | OPER | ESB | RQS | OFF
    :PON
        :STATe RST | RCL0
    :PROTection
        :CLEar
        :DELay <n>
    :RI
        :MODE LATCHing | LIVE | OFF
    TTLTrig
        MODE TRIG | FSTR
        [STATe] <bool>
        :SOURce BOT | EOT | LIST

PONSetup
    :CLOCk
    :PEAK:CURRent[:PROTection]
    :RELay:HOLD

SENSe
    [:COUPle] AC | DC
    :CURRent
        :ACDC
            :RANGe
                [:UPPer] <n>

[SOURce:]
    CURRent
```

```
        :ALC
                [:STATe]  ON | OFF | REG
        [:LEVel]
             [:IMMediate]
                [:AMPLitude] <n>
             :TRIGgered
                [:AMPLitude] <n>
        :MODE FIXed | STEP | PULSe | LIST
        :SLEW
                [:IMMediate] <n> | INFinity
                :MODE FIXed | STEP | PULSe | LIST
                :TRIGgered <n> | INFinity
    FREQuency
        [:CW | :IMMediate] <n>
            :MODE FIXed|STEP|PULSe|LIST|SENS|EXT
        :SLEW
                [:IMMediate] <n> | INFinity
                :MODE FISed | STEP | PULSe | LIST
                :TRIGgered <n> | INFinity
        :TRIGgered <n>
    LIST
        :COUNt <n> | INFinity
        :CURRent
             [:LEVel] <n>{,<n>}
                :POINts?
             :SLEW  <n>{,<n>}
                :POINts?
        :DWELl <n>{,<n>}
                :POINts?
        :FREQuency
             [:LEVel] <n>{,<n>}
                :POINts?
             :SLEW  <n>{,<n>}
                :POINts?
        :PHASe <n>{,<n>}
                :POINts?
        :SHAPe <shape>{,<shape>}
                :POINts?
        :STEP ONCE | AUTO
        :TTLTrg <bool>{,<bool>}
    PHASe
        [:IMMediate] <n>
        :MODE FIXed | STEP | PULSe | LIST
        :TRIGgered <n>
    PULSe
        :COUNt <n> | INFinity
        :DCYCle <n>
        :HOLD WIDTh | DCYCle
        :PERiod <n>
        :WIDTh <n>
    VOLTage
        [:LEVel]
             [:IMMediate]
                [:AMPLitude] <n>
        :PROTection
                DELay <n>
                :STATe <bool>
        :SENSe
                :SOURce INT | EXT

STATus
    :OPERation
        [:EVENt]?
        :CONDition?
        :ENABle <n>
        :NTRansition <n>
        :PTRansition <n>
    :PRESet
    :QUEStionable
        [:EVENt]?
        :CONDition?
        :ENABle <n>
        :INSTrument
            :ISUMmary
                [:EVENt]?
                :CONDition?
                :ENABle <n>
                :NTRansition <n>
                :PTRansition <n>
        :NTRansition <n>
        :PTRansition <n>

SYSTem
    :CONFiguration
        :NOUTputs <n>
    :ETIMe?
    :ERRor?
    :LANGuage SCPI | APEN
    :REMote
    :RWLOCK
    :TEMPerature?
    :VERSion?

TRIGger
    [:TRANsient | SEQuence1]
        [:IMMediate]
        :SOURce BUS | EXTernal | IMMediate
        DELay <n>
    :SYNChronize | :SEQuence2
        :SOURce PHASe | IMMediate
        :PHASe <n>
    :ACQuire | :SEQuence3
        [:IMMediate]
        :SOURce BUS | EXTernal | IMMediate
    :SEQuence1
        :DEFine TRANsient
    :SEQuence2
        :DEFine SYNChronize
    :SEQuence3
        :DEFine ACQuire
```

Appendix B : SCPI Conformance Information

SCPI Version

The power source conforms to SCPI version 1992.0.

SCPI Confirmed commands

```
ABOR
CAL:DATA
CAL:STAT
INIT[:IMM]
INIT[:IMM]:SEQ or NAME
INIT:CONT:SEQ or NAME
INST:COUP
INST:NSEL
MEAS or FETC[:SCAL]:CURR[:DC]?
MEAS or FETC[:SCAL]:CURR:AC?
MEAS or FETC[:SCAL]:FREQ?
MEAS or FETC[:SCAL]:POW[:DC]?
MEAS or FETC[:SCAL]:POW:AC[:REAL]?
MEAS or FETC[:SCAL]:VOLT:AC?
MEAS or FETC:ARR:CURR[:DC]?
MEAS or FETC:ARR:VOLT[:DC]?
OUTP:[STAT]
OUTP:COUP
OUTP:PROT:CLE
OUTP:PROT:DEL
OUTP:TTLT:MODE
OUTP:TTLT[:STAT]
PONS:CLOC
SENSe[:COUPle]
SENSe:CURR:ACDC:RANG[:UPP]
SENS:SWE:OFFS:POIN
SENS:SWE:TINT
[SOUR:]VOLT[:LEV][:IMM][:AMPL]
[SOUR:]VOLT:PROT:DEL
[SOUR:]VOLT:PROT:STAT
[SOUR:]VOLT:SENS:SOUR
[SOUR:]FREQ[:CW or :IMM]
[SOUR:]FREQ:MODE
[SOUR:]LIST:COUN
[SOUR:]LIST:CURR
[SOUR:]LIST:CURR:POIN?
[SOUR:]LIST:DWEL
[SOUR:]LIST:DWEL:POIN?
[SOUR:]LIST:FREQ[:LEV]
[SOUR:]LIST:FREQ[:LEV]:POIN?
[SOUR:]LIST:VOLT[:LEV]
[SOUR:]LIST:VOLT[:LEV]:POIN?
[SOUR:]PHAS[:IMM]
[SOUR:]PULS:COUN
[SOUR:]PULS:DCYC
[SOUR:]PULS:HOLD
[SOUR:]PULS:PER
[SOUR:]PULS:WIDT
[SOUR:]CURR[:LEV][:IMM][:AMPL]
[SOUR:]CURR [:LEV][:TRIG][:AMPL]
[SOUR:]CURR:MODE
[SOUR:]CURR:SLEW[:IMM]
STAT:OPER[:EVEN]?
STAT:OPER:COND?
STAT:OPER:ENAB?
STAT:OPER:NTR
STAT:OPER:PTR
STAT:PRES
STAT:QUES:
STAT:QUES[:EVEN]?
STAT:QUES:COND?
STAT:QUES:ENAB?
STAT:QUES:NTR
STAT:QUES:PTR
STAT:QUES:INST:ISUM[:EVEN]?
STAT:QUES:INST:ISUM:COND?
STAT:QUES:INST:ISUM:ENAB
STAT:QUES:INST:ISUM:NTR
STAT:QUES:INST:ISUM:PTR
SYST:ERR?
SYST:LANG
SYST:VERS?
TRAC or DATA:CAT?
TRAC or DATA[:DATA]
TRAC or DATA:DEF
TRAC or DATA:DEL[:NAME]
TRIG:SEQ<n>:DEL
TRIG:SEQ<n>[:IMM]
TRIG:SEQ<n>:SOUR
TRIG:SOUR
*CLS
*ESE   *ESE?   *ESR?
*IND?
*OPC   *OPC?   *OPT?
*PSC   *PSC?
*RCL   *RST
*SAV   *SRE   *STB?
*TRG   *TST?
*WAI
```

Non-SCPI commands.

```
CAL:MEAS:CURR
CAL:MEAS:CURR:SPH
CAL:MEAS:VOLT
CAL:PASS
CAL:PHAS
CAL:SAVE
CAL:VOLT
DIAG:RES
DIAG:TEMP:AMB
LIM:CURR
LIM:FREQ:HIGH
LIM:FREQ:LOW
LIM:PHAS
LIM:VOLT:HIGH
LIM:VOLT:LOW
MEAS or FETC[:SCAL]:CURR:ACDC?
MEAS or FETC[:SCAL]:CURR:HARM[:AMPL]?
MEAS or FETC[:SCAL]:CURR:HARM:PHAS?
MEAS or FETC[:SCAL]:CURR:HARM:THD?
MEAS or FETC[:SCAL]:CURR:NEUT[:DC]?
MEAS or FETC[:SCAL]:CURR:NEUT:AC?
MEAS or FETC[:SCAL]:CURR:NEUT:DC?
MEAS or FETC[:SCAL]:CURR:NEUT:ACDC?
MEAS or FETC[:SCAL]:CURR:NEUT:HARM[:AMPL]?
MEAS or FETC[:SCAL]:CURR:NEUR:HARM:PHAS?
MEAS or FETC[:SCAL]:POW:AC:APP?
MEAS or FETC[:SCAL]:POW:AC:REAC?
MEAS or FETC[:SCAL]:POW:AC:PFAC?
MEAS or FETC[:SCAL]:POW:AC:TOT
MEAS or FETC[:SCAL]:VOLT:ACDC?
MEAS or FETC[:SCAL]:VOLT:AMPL:MAX?
MEAS or FETC[:SCAL]:VOLT:AMPL:RES
MEAS or FETC[:SCAL]:VOLT:CRES?
MEAS or FETC[:SCAL]:VOLT:HARM[:AMPL]?
MEAS or FETC[:SCAL]:VOLT:HARM:PHAS?
MEAS or FETC[:SCAL]:VOLT:HARM:THD?
MEAS or FETC:ARR:CURR:HARM[:AMPL]?
MEAS or FETC:ARR:CURR:HARM:PHAS?
MEAS or FETC:ARR:CURR:NEUT[:DC]?
MEAS or FETC:ARR:CURR:NEUT:HARM[:AMPL]?
MEAS or FETC:ARR:CURR:NEUT:HARM:PHAS?
MEAS:ARR:MOD
MEAS or FETC:ARR:VOLT:HARM[:AMPL]?
MEAS or FETC:ARR:VOLT:HARM:PHAS?
MEAS:THD:MODE
OUTP:DFI[:STAT]
OUTP:DFI:SOUR
OUTP:IMP:REAL
OUTP:IMP:REAC
OUTP:RI:MODE
OUTP:TTLT:SOUR
PONS:CLOC
PONS:PEAK:CURR[:PROT]
PONS:RELAY:HOLD
[SOUR:]CURR:ALC[:STAT]
[SOUR:]CURR:SLEW:MODE
[SOUR:]CURR:SLEW:TRIG
[SOUR:]FREQ:SLEW[:IMM}
[SOUR:]FREQ:SLEW:MODE
[SOUR:]FREQ:SLEW:TRIG
[SOUR:]FREQ:TRIG
[SOUR:]FUNC[:SHAP]:CSIN
[SOUR:]FUNC[:SHAP]:MODE
[SOUR:]FUNC[:SHAP]:TRIG
[SOUR:]LIST:FREQ:SLEW
[SOUR:]LIST:FREQ:SLEW:POIN?
[SOUR:]LIST:PHAS
[SOUR:]LIST:PHAS:POIN?
[SOUR:]LIST:SHAP
[SOUR:]LIST:SHAP:POIN?
[SOUR:]LIST:STEP
[SOUR:]LIST:TTLT
[SOUR:]LIST:TTLT:POIN?
[SOUR:]LIST:CURR:SLEW
[SOUR:]LIST:CURR:SLEW:POIN?
[SOUR:]PHAS:MODE
[SOUR:]PHAS:TRIG
[SOUR:]VOLT:PEAK[:IMM]
[SOUR:]VOLT:PEAK:MODE
[SOUR:]VOLT:PEAK:TRIG
[SOUR:]VOLT:SENS[:SOUR]
SYST:CONF?
SYST:CONF:NOUT?
SYST:ETIM?
SYST:LOC
SYST:REM
SYST:RWL
SYST:TEMP?
TRIG:SEQ2 or SYNC:PHAS
TRIG:SEQ:DEF
```

Appendix C : Error Messages

This appendix gives the error numbers and descriptions that are returned by the AC power source. Error numbers are returned in two ways:

- Error numbers are displayed on the front panel
- Error numbers and messages are read back with the “SYSTem:ERRor?” query. SYSTem:ERRor? returns the error number into a variable and returns two parameters: an NR1 and a string.

When errors occur, the Standard Event Status register records them as shown in the following table.

Bit Set	Code Error	Type
5	-100 thru -199	Command
4	-200 thru -299	Execution
3	-300 thru -399 or 1 thru 799	Device-dependent Note that errors over 800 do not set the ESR.
2	-400 thru -499	Query

Standard Event Status register error bits

The following table lists the errors that are associated with SCPI syntax errors and interface problems. It also lists the device dependent errors. Information inside the brackets is not part of the standard error message, but is included for clarification.

Some error messages are maintained for back ward compatibility with Agilent/HP 6834B models.

Number	Message String	Cause	Remedy
0	"No error"	No errors in queue	
-100	"Command error"	Unable to complete requested operation	Unit may be in a mode inconsistent with request.
-102	"Syntax error"	Command syntax incorrect.	Misspelled or unsupported command
-103	"Invalid separator"	SCPI separator not recognized	See SCPI section of programming manual.
-104	"Data type error"	Data type invaled.	Check command for supported data types
-108	"Parameter not allowed"	One or more additional parameters were received.	Check programming manual for correct number of parameters
-109	"Missing parameter"	Too few parameters received for requested operation	Check programming manual for correct number of parameters
-110	"Command header error"	Command header incorrect	Check syntax of command.
-111	"Header separator error"	Invalid command separator used.	Use semi-colon to separate command headers
-112	"Program mnemonic too long"	Syntax error	Check programming manual for correct command syntax

Number	Message String	Cause	Remedy
-113	"Undefined header"	Command not recognized error	Check programming manual for correct command syntax
-120	"Numeric data error"	Data received is not a number	Check programming manual for correct command syntax
-121	"Invalid character in number"	Number received contains non-numeric character(s)	Check programming manual for correct command syntax
-123	"Exponent too large"	Exponent in number exceeds limits	Check programming manual for correct parameter range
-128	"Numeric data not allowed"	Number received when number is not allowed.	Check programming manual for correct command syntax
-168	"Block data not allowed"	Block data was sent.	Check programming manual for correct command syntax
-200	"Execution error"	Command could not be executed	Command may be inconsistent with mode of operation.
-201	"Invalid while in local"	Command issued but unit is not in remote state	Put instrument in remote state before issuing GPIB commands.
-203	"Command protected"	Command is locked out	Some commands are supported by the unit but are locked out for protection of settings and are not user accessible.
-210	"Trigger error"	Problem with trigger system.	Unit could not generate trigger for transient execution or measurement.
-211	"Trigger ignored"	Trigger request has been ignored.	Trigger setup incorrect or unit was not armed when trigger was received. Check transient system or measurement trigger system settings.
-213	"Init ignored"	Initialization request has been ignored	Unit was told to go to armed state but was unable to do so. Could be caused by incorrect transient system or measurement acquisition setup.
-220	"Parameter error"	Parameter not allowed.	Incorrect parameter or parameter value. Check programming manual for allowable parameters
-221	"Setting conflict"	Transient programmed with more than 1 mode.	Check other settings. E.g. Redefine transient mode. As result of *TST? execution, indicates ALC mode is off or waveform not set to Sine.
-222	"Data out of range"	Parameter data outside of allowable range.	Check programming manual for allowable parameter values
-223	"Too much data"	More data received than expected	Check programming manual for number of parameters or data block size
-224	"Illegal parameter value"	Parameter value is not suppored	Check programming manual for correct parameters
-226	"Lists not same length"	One or more transient lists programmed has different length.	All lists must be of same length or transient cannot be compiled and executed.

Number	Message String	Cause	Remedy
-241	"Hardware missing"	N/A	N/A
-254	"Media full"	No storage space left to save settings or data.	Delete other settings or data to make room.
-255	“Directory full”	Too many waveform directory entries	Delete one or more waveforms from waveform memory to make room.
-256	“File name not found”	Waveform requested not in directory	Check waveform directory for waveform names present.
-257	“File name error”	Incorrect filename	Too many or non ASCII characters used in waveform file definition.
-283	“Illegal variable name”	Variable name illegal.	Use ASCII characters only
-300	"Device specific error"	Hardware related error	Check hardware for proper operation.
-311	"Memory error"	Waveform memory checksum error.	May be the result of incomplete user-defined waveform download. Check interface and try downloading waveform again. Successful download may clear this error condition. Alternatively, use TRAC:DEL:ALL command to clear waveform memory.
-314	"Save/recall memory lost"	User setup register contents lost	Store setup in same register again.
-315	"Configuration memory lost"	Hardware configuration settings lost.	Contact CI service department at support@calinst.com to obtain instructions on restoring configuration data.
-330	"Self-test failed"	Internal error	Contact CI service department at support@calinst.com
-350	"Queue overflow"	Message queue full.	Too many message. Read status using SYST:ERR query until 0, "No Error" is received indicating queue empty.
-400	"Query error"	Unable to complete query.	Check programming manual for correct query format and parameters
-410	"Query INTERRUPTED"	Query issued but response not read.	Check application program for correct flow. Response must be read after each query to avoid this error.
-420	"Query UNTERMINATED"	Query incomplete.	Check for terminator after query command.
-430	"Query DEADLOCKED"	Query cannot be completed	Check application program for multiple queries
-440	"Query UNTERMINATED"	Query incomplete.	Check for terminator after query command.
0	"No error"	No errors in queue	
2	" Non-volatile RAM CONFIG section checksum failed"	Controller failure during Self-test.	Contact CI service department at support@calinst.com
3	" Non-volatile RAM CAL section checksum failed"	Controller failure during Self-test.	Contact CI service department at support@calinst.com

Number	Message String	Cause	Remedy
4	" Non-volatile RAM WAVEFORM section checksum failed"	Controller failure during Self-test.	Contact CI service department at support@calinst.com
10	"Ram self test	Controller failure during Self-test.	Contact CI service department at support@calinst.com
40	"Voltage self test error, output 1	No. 1/A amplifier in Master source has no output during Self-test.	Contact CI service department at support@calinst.com
41	"Voltage self test error, output 2	No. 2/B amplifier in Master source has no output during Self-test	Contact CI service department at support@calinst.com
42	"Voltage self test error, output 3	No. 3/C amplifier in Master source has no output during Self-test	Contact CI service department at support@calinst.com
43	"Current self test error, output 1	No. 1/A amplifier in Aux. Source has no output during Self-test.	Contact CI service department at support@calinst.com
44	"Current self test error, output 2	No. 2/B amplifier in Aux. Source has no output during Self-test.	Contact CI service department at support@calinst.com
45	"Current self test error, output 3	No. 3/C amplifier in Aux. Source has no output during Self-test.	Contact CI service department at support@calinst.com
216	" RS-232 receiver framing error"	Communication failure.	Check RS232 port settings and cable.
217	" RS-232 receiver parity error"	Communication failure.	Check RS232 port settings and cable.
218	" RS-232 receiver overrun error"	Communication failure.	Check RS232 port settings and cable.
402	"CAL password is incorrect"	Calibration password does not equal high voltage range value.	Re-enter correct password.
403	"CAL not enabled"	No password entered for calibration	Enter correct CAL password.
600	"Systems in mode:list have different list lengths"	Transient lists have unequal lengths	Check list settings and correct to same no of data points.
601	"Requested voltage and waveform exceeds peak voltage capability"	Wave shape selected and RMS voltage combine to exceed peak voltage capability.	Reduce RMS or crest factor of wave shape.
602	"Requested voltage and waveform exceeds transformer volt-second rating"	The selected wave shape exceeds output transformer capability.	The volt-second product of he wave form (magnitude and time in the + and – half of wave form).
603	"Command only applies to RS-232 interface"	Command not	Do not use command.

Number	Message String	Cause	Remedy
	to RS-232 interface"	relevant for GPIB interface.	
604	"Trigger received before requested number of pre-trigger readings"	Data acquisition pre-trigger buffer not filled yet.	Hold off trigger or reduce pre-trigger delay.
605	"Requested RMS current too high for voltage range"	Max RMS current is function of voltage range selected.	Reduce programmed RMS current limit or select low voltage range.
606	"Waveform data not defined"	No waveform name specified	Specify waveform name before sending waveform data.
607	"VOLT,VOLT:SLEW, and FUNC:SHAPe modes incompatible"	Conflict between wave shape and programmed slew	Reduce slew or change waveform type.
608	"Measurement overrange"	Measurement data out of range.	
609	"Output buffer overrun"	Too much data in output buffer.	Check receive mode on application program. Program is not reading data sent by AC source.
610	"Command cannot be given with present SYST:CONF setting"	Command conflicts with available hardware or firmware option settings.	Check configuration for available options and features.
801	"Output current fault"	- Output current does not match program value when ALC is on. - Over load - High impedance or open load. - No output current	Load impedance exceeds compliance voltage limit and unit is in Constant Current (CC) mode of operation. Reduce load or increase Voltage setting.
802	"Voltage limit fault"	Voltage limit exceeded.	Load impedance exceeds compliance voltage limit and unit is in Constant Current (CC) mode of operation. Reduce load or increase Voltage setting.
803	"Temperature fault"	Amplifier heat sink temp. too high.	Reduce load. Ensure proper air flow and exhaust clearance. Check fan(s) for operation.
804	"External sync. error"	Could not sync to external sync signal.	External sync signal missing, disconnected or out of range.
805	"Initial memory lost"	Initial settings could not be recalled at power-up.	Save power on settings again to overwrite old content.
806	"Limit memory lost"	Hardware configuration settings could not be recalled at power-up.	Contact CI service department at support@calinst.com to obtain instructions on restoring configuration data.
807	"System memory lost"	Memory corrupted during power-up.	Recycle power.
808	"Calibration memory lost"	Calibration data lost during power-up.	Contact CI service department at support@calinst.com to obtain instructions on restoring calibration data or recalibrate unit.

Number	Message String	Cause	Remedy
813	"Missing list parameter"	One or more transient list parameters missing.	Check programmed lists.
814	"Voltage peak error "	Peak voltage exceeds internal bus voltage	This error may occur when selecting user defined wave shapes with higher crest factors. Reduce programmed RMS value.
815	"Slew time exceed dwell"	Time needed to slew to final value is less than dwell time.	Check dwell times in transient list settings. Increase dwell time or change slew rate for affected parameter.
816	"Illegal during transient"	Operation requested not available while transient is running.	Wait till transient execution is completed or abort transient execution first.
817	"Output relay must be closed"	Transient programmed with output relay open.	Close relay before attempting transient operation.
819	"Clock and sync must be internal"	Operation not possible with external clock	Switch to internal sync. (Default)
820	"Input buffer full"	Too much data received.	Break up data in smaller blocks.
821	"Amplifier unbalance"	Hardware error. An amplifier has an overload condition.	Check amplifier balance adjustment. If error persists contact CI service at support@calinst.com .
822	"Waveform harmonics limit"	Harmonic contents of user defined wave shape is too high and could damage amplifier output stage.	Reduce harmonic content or reduce fundamental frequency programmed.
823	"Amplifier fault"	An amplifier failure. Can be reported at any time.	Determine which amplifier is at fault with self-test or checking LED on Relay Board. Replace amplifier.
824	“Auxiliary down”	One or more auxiliary units is not powered up or not working.	Turn on all auxiliary units.
825	“Over voltage prot trip”	Over voltage detected on output	Check output voltage for correct RMS value.
826	“Peak current prot trip”	Peak current limit exceeded.	Peak current exceeded. Could be caused by switching EUT on or off.
827	“Frequency error”	Frequency error during self-test.	Correct frequency was not measured during self-test. May be result of 801 error.
828	“Phase error”	Self test error phase angle	Correct phase angle was not measured during self-test. May be result of 801 error.
829	“Dc component exceed limit”	Too much DC content in loaded ARB waveform.	Check waveform programming.

Table 8-1: SCPI error codes and messages.

Index

*

*CLS 101
*ESE 101
*ESR? 101
*IDN? 102
*OPC 102
*OPT? 102
*PSC 102
*RCL 103
*RST 104
*SAV 105
*SRE 105
*STB? 105
*TRG 106
*TST? 106
*WAI 107

A

ABORt 95
address
  IEEE 9

B

block header 37, 41
Bus speed
  delays 128

C

CALibrate:MEASure:CURRent 31
CALibrate:MEASure:CURRent:SPHase 31
CALibrate:MEASure:VOLTage 31
CALibrate:PASSword 32
CALibrate:SAVE 32
CALibrate[:SOURce]:CURRent 32
CALibrate[:SOURce]:PHASe 32
command
  subsystem 30
Command
  delays 128
commands
  common 13
  coupled 14
Conventions 11
Copyright 2
coupling 14
CURRent 81
CURRent:ALC[:STATe] 80
CURRent:MODE 81
CURRent:SLEW 82
CURRent:SLEW:MODE 82
CURRent:SLEW:TRIGgered 82
CURRent:TRIGgered 81

D

DATA 93
DATA:CATalog 94
DATA:DEFine 94
DATA:DELete 94
DATA:DELete:ALL 94

E

Ext. Trigger 57

F

FETCh:ARRay:CURRent:HARMonic:PHASe?38
FETCh:ARRay:CURRent:HARMonic? 38
FETCh:ARRay:CURRent:NEUTral:HARMonic:PHASe? 39
FETCh:ARRay:CURRent:NEUTral:HARMonic? 39
FETCh:ARRay:CURRent:NEUTral? 38
FETCh:ARRay:CURRent? 37
FETCh:ARRay:VOLTage:HARMonic? 42
FETCh:ARRay:VOLTage? 41
FETCh:CURRent:AC? 43
FETCh:CURRent:ACDC? 44
FETCh:CURRent:HARMonic:PHASe? 44
FETCh:CURRent:HARMonic:THD? 45
FETCh:CURRent:HARMonic? 44
FETCh:CURRent:NEUTral:AC? 45
FETCh:CURRent:NEUTral:ACDC? 45
FETCh:CURRent:NEUTral:HARMonic:PHASe? 46
FETCh:CURRent:NEUTral:HARMonic? 46
FETCh:CURRent:NEUTral? 45
FETCh:CURRent? 43
FETCh:FREQuency? 48
FETCh:POWer:AC:APParent? 49
FETCh:POWer:AC:PFACtor? 50
FETCh:POWer:AC:REACtive? 50
FETCh:POWer:AC:TOTal? 50
FETCh:POWer:AC? 49
FETCh:POWer? 49
FETCh:VOLTage :AMPLitude:RESet 52
FETCh:VOLTage :CRESTfactor? 52
FETCh:VOLTage:AC? 51
FETCh:VOLTage:ACDC? 51
FETCh:VOLTage:AMPLitude:MAXimum? 52
FETCh:VOLTage:HARMonic:PHASe? 53
FETCh:VOLTage:HARMonic:THD? 53
FETCh:VOLTage:HARMonic? 53
FETCh:VOLTage? 51
firmware
  updates 10
format
  waveform data 39

- formats
  - data 17
- FREQency:SLEW:TRIGgered 66
- FREQuency 65
- FREQuency:MODE 65
- FREQuency:SLEW 66
- FREQuency:SLEW:MODE 66
- FREQuency:TRIGgered 67
- FUNCtion 68
- Function Strobe 57
- FUNCtion:CSINusoid 69
- FUNCtion:MODE 68
- FUNCtion:TRIGgered 69

H

- header
  - optional 16
  - separator 15
- **Header** 12
- **headers**
  - **optional** 13
- Headers 15

I

- IEEE 8
  - setting address 9
- **Initialization** 108
- INITiate:CONTinuous:NAME 96
- INITiate:CONTinuous:SEQuence 96
- INITiate:NAME 95
- INITiate:SEQuence 95
- INSTrument:COUPle 34
- INSTrument:NSELect 34
- INSTrument:SELect 34
- IP address 26

L

- LAN Option
  - IP address 26
  - Port number 27
- LIMit: VOLTage:HIGH 36
- LIMit: VOLTage:LOW 36
- LIMit:CURRent 35
- LIMit:FREQuency:HIGH 35
- LIMit:FREQuency:LOW 35
- LIMit:PHASe 35
- LIST: CURRent:SLEW 71
- LIST: CURRent:SLEW:POINts? 71
- LIST:COUNt 71
- LIST:CURRent 71
- LIST:CURRent:POINts? 71
- LIST:DWELl 72
- LIST:DWELl:POINts? 72
- LIST:FREQuency 72
- LIST:FREQuency:POINts? 72
- LIST:FREQuency:SLEW 73
- LIST:FREQuency:SLEW:POINts? 73
- LIST:PHASe 73
- LIST:PHASe:POINts? 73
- LIST:STEP 73
- LIST:TTLTrg 74
- LIST:TTLTrg:POINts? 74

M

- MEASure:ARRay:CURRent:HARMonic:PHASe? 38
- MEASure:ARRay:CURRent:HARMonic? 38
- MEASure:ARRay:CURRent:NEUTral:HARMonic:PHASe? 39
- MEASure:ARRay:CURRent:NEUTral:HARMonic? 39
- MEASure:ARRay:CURRent:NEUTral?" 38
- MEASure:ARRay:CURRent? 37
- MEASure:ARRay:MODe 39
- MEASure:ARRay:VOLTage:HARMonic? 42
- MEASure:ARRay:VOLTage? 41
- MEASure:CURRent:AC? 43
- MEASure:CURRent:ACDC? 44
- MEASure:CURRent:HARMonic:PHASe? 44
- MEASure:CURRent:HARMonic:THD? 45
- MEASure:CURRent:HARMonic? 44
- MEASure:CURRent:NEUTral:AC? 45
- MEASure:CURRent:NEUTral:ACDC? 45
- MEASure:CURRent:NEUTral:HARMonic:PHASe? 46
- MEASure:CURRent:NEUTral:HARMonic? 46
- MEASure:CURRent:NEUTral? 45
- MEASure:CURRent? 43
- MEASure:FREQuency? 48
- MEASure:POWer:AC:APParent? 49
- MEASure:POWer:AC:PFACtor? 50
- MEASure:POWer:AC:REACtive? 50
- MEASure:POWer:AC:TOTal? 50
- MEASure:POWer:AC? 49
- MEASure:POWer? 49
- MEASure:THDistortion:MODE 46, 54
- MEASure:VOLTage :AMPLitude:RESet 52
- MEASure:VOLTage :CRESTfactor? 52
- MEASure:VOLTage:AC? 51
- MEASure:VOLTage:ACDC? 51
- MEASure:VOLTage:AMPLitude:MAXimum? 52
- MEASure:VOLTage:HARMonic:PHASe? 53
- MEASure:VOLTage:HARMonic:THD? 53
- MEASure:VOLTage:HARMonic? 53
- MEASure:VOLTage? 51

O

- OUTPut 55
- OUTPut:DFI 55
- OUTPut:DFI:SOURce 56
- OUTPut:PON:STATe 56
- OUTPut:PROTection:CLEar 56

OUTPut:PROTection:DELay 57
OUTPut:RI:MODE 57
OUTPut:TTLTrg:MODE 57
OUTPut:TTLTrg:SOURce 58
OUTPut:TTLTrg[:STATe] 58

P

PHASe 75
PHASe:MODE 75
PHASe:TRIGgered 75
PONSetup:CLOCk 59
PONSetup:PEAK:CURRent[:PROTection] 59
PONSetup:RELay[:HOLD] 60
Port number
  TCP/IP 27
PULSe:COUNt 77
PULSe:DCYCle 77
PULSe:HOLD 77
PULSe:PERiod 78
PULSe:WIDTh 78

Q

queries 14

R

root 16
Root 12
root specifier 16
RS232C
  cable 28

S

SCPI 11
  Command Tree 12
  message 11
  message unit 14
  program message 11
  response message 11
  structure 14
selftest 106
SENSe:SWEep:OFFSet 61
SENSe:SWEep:OFFSet:POINts 61
SENSe:SWEep:TINTerval 62
SENSe[:COUPle] 61
Separator 15
STATus:OPERation:CONDition? 84
STATus:OPERation:ENABle 84
STATus:OPERation:NTR
  STATus:OPERation:PTR 84
STATus:OPERation? 84
STATus:PRESet 83
STATus:QUEStionable:CONDition? 86
STATus:QUEStionable:ENABle 86
STATus:QUEStionable:INSTrument:ISUMmary:
  CONDition? 88
STATus:QUEStionable:INSTrument:ISUMmary:
  ENABle 88
STATus:QUEStionable:INSTrument:ISUMmary:
  NTR 88
STATus:QUEStionable:INSTrument:ISUMmary:
  PTR 88
STATus:QUEStionable:INSTrument:ISUMmary?
  87
STATus:QUEStionable:NTR
  STATus:QUEStionable:PTR 86
STATus:QUEStionable? 85
SYSTem:CONFiguration 90
SYSTem:CONFiguration:NOUTputs 90
SYSTem:ERRor? 91
SYSTem:ETIMe? 90
SYSTem:LANGuage 91
SYSTem:LOCal 91
SYSTem:REMote 91
SYSTem:RWLock 91
SYSTem:TEMPerature? 92
SYSTem:VERSion? 92

T

Table of Contents 5
Table of Figures 7
Table of Tables 7
TCP/IP
  IP address 26
  Port number 27
terminator 16
TRACe 93
TRACe:CATalog 94
TRACe:DEFine 94
TRACe:DELete 94
TRACe:DELete:ALL 94
TRIGger 97
TRIGger:ACQuire 98
TRIGger:ACQuire:SOURce 99
TRIGger:DELay 97
TRIGger:SEQuence1:DEFine 99
TRIGger:SEQuence2:DEFine 99
TRIGger:SEQuence2:PHASe 98
TRIGger:SEQuence2:SOURce 98
TRIGger:SEQuence3 98
TRIGger:SEQuence3:DEFine 99
TRIGger:SEQuence3:SOURce 99
TRIGger:SOURce 97
TRIGger:SYNCHronize:PHASe 98
TRIGger:SYNChronize:SOURce 98

U

User manual 9
User's Manual 8

V

VB sample

waveform conversion 40, 123
VOLTage..63
VOLTage:PROTection:DELay........................63
VOLTage:PROTection:STATe........................64
VOLTage:SENSe:SOURce.............................64

W

waveform
data format modes 40, 123