Keysight Models 6811C, 6812C, and 6813C AC Power Solutions

Programming Guide

Notices

Copyright Notice

Manual Part Number

06811-90005

Edition

Edition 1, October 2, 2017

Printed in:

Printed in Malaysia

Published by:

Keysight Technologies
Bayan Lepas Free Industrial Zone,
11900 Penang, Malaysia

Technology Licenses

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

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR OF ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

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

La mention ATTENTION signale un danger. Si la manoeuvre ou la procédure correspondante n'est pas exécutée correctement, il peut y avoir un risque d'endommagement de l'appareil ou de perte de données importantes. En présence de la mention ATTENTION, il convient de ne pas poursuivre tant que les conditions indiquées n'ont pas été parfaitement comprises et remplies.

WARNING

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

La mention AVERTISSEMENT signale un danger pour la sécurité de l'opérateur. Si la manoeuvre ou la procédure correspondante n'est pas exécutée correctement, il peut y avoir un risque grave, voire mortel pour les personnes. En présence d'une mention AVERTISSEMENT, il convient de s'interrompre tant que les conditions indiquées n'ont pas été parfaitement comprises et satisfaites.

Safety Considerations

Read the information below before using this instrument.

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this guide violates safety standards for design, manufacture, and intended use of the instrument. Keysight Technologies assumes no liability for the customer’s failure to comply with these requirements.

WARNING

ENERGY HAZARD. AC sources can supply 425 V peak at their output. DEATH on contact may result if the output terminals or circuits connected to the output are touched when power is applied.

RISQUE ÉNERGÉTIQUE. Les sources AC peuvent fournir une tension de sortie de 425 V maximum. Tout contact peut entraîner la MORT si les circuits ou bornes de sortie connectés à la sortie sont touchés lorsqu’ils sont sous tension.

Waste Electrical and Electronic Equipment (WEEE) Directive

This instrument complies with the WEEE Directive marking requirement. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.

Product category:

With reference to the equipment types in the WEEE directive Annex 1, this instrument is classified as a “Monitoring and Control Instrument” product.

The affixed product label is as shown below.

Do not dispose in domestic household waste.

To return this unwanted instrument, contact your nearest Keysight Service Center, or visit http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information.

Sales and Technical Support

To contact Keysight for sales and technical support, refer to the support links on the following Keysight websites:

- www.keysight.com/find/6800
  (product-specific information and support, software and documentation updates)
- www.keysight.com/find/assist
  (worldwide contact information for repair and service)

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Table of Contents

List of Figures

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

List of Tables

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

1 General Information

This chapter provides an introduction to this guide.

About this Guide

This guide contains programming information for the Keysight Models 6811C, 6812C, and 6813C AC Power Solutions. These units will be referred to as “ac source” throughout this guide. You will find the following information in the rest of this guide:

– Chapter 1, an introduction to this guide.

– Chapter 2, an introduction to SCPI messages structure, syntax, and data formats.

– Chapter 3, a dictionary of the available SCPI commands.

– Chapter 4, information about the compatibility of the B and C versions of the ac source.

– Chapter 5, an introduction to programming the ac source with SCPI commands.

– Appendix A, the SCPI command tree.

– Appendix B, SCPI conformance information.

– Appendix C, list of error messages.

Documentation summary

The following documents that are related to this *Programming Guide* have additional helpful information for using the ac source.

– *Quick Start Guide*. Information on how to quickly get started using the ac source.

– *User’s and Service Guide*. Includes specifications and supplemental characteristics, how to use the front panel, how to connect to the ac source, and calibration procedures.

External References

SCPI references

The following documents will assist you with programming in SCPI:

– *Beginner's Guide to SCPI*. Keysight part no. H2325-90001. Highly recommended for anyone who has not had previous experience programming with SCPI.

– *Tutorial Description of the General Purpose Interface Bus*. Keysight part no. 5952-0156. Highly recommended for those not familiar with the IEEE 488.1 and 488.2 standards.

To obtain a copy of the above documents, contact your local Keysight Sales and Support Office.

GPIB references

The most important GPIB documents are your controller programming manuals - Keysight BASIC, GPIB Command Library for MS DOS, etc. Refer to these for all non-SCPI commands (for example: Local Lockout).

The following are two formal documents concerning the GPIB interface:

– *ANSI/IEEE Std. 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation*. Defines the technical details of the GPIB interface. While much of the information is beyond the need of most programmers, it can serve to clarify terms used in this guide 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.

Keysight IO Libraries Suite

Keysight IO Libraries Suite is a collection of free instrument control software that automatically discovers instruments and allows you to control instruments over LAN, USB, GPIB, RS-232, and other interfaces. For more information, or to download IO Libraries, go to www.keysight.com/find/iosuite.

2 Introduction to Programming

This chapter provides an introduction to SCPI messages structure, syntax, and data formats.

Introduction

The ac source supports the SCPI programming language on all of its remote I/O interfaces.

NOTE It is not recommended to send SCPI commands simultaneously using more than one remote I/O interface.

USB Capabilities of the AC Source

To use the USB interface for the ac source, no settings are required on the ac source.

Vendor Identification (VID) and Product Identification (PID)

The vendor identification (VID) and product identification (PID) of the ac sources are specified as below:

VID	PID	Keysight model
0x2A8D	0x3601	6811C
	0x3701	6812C
	0x3801	6813C

LAN Capabilities of the AC Source

The ac source conforms to the LAN eXtensions for Instrumentation (LXI) Device Specification 2011 version 1.4. The Telnet, Socket, and VXI-11 interfaces are available to communicate with the ac source that works on either static or dynamic IP address.

Instrument Web server

The ac source has a built-in Web server that gives you direct control from an Internet browser on your computer. With the Web server, you can access the front panel control functions including the LAN configuration parameters.

NOTE **The built-in Web server only operates over the LAN interface. It requires Internet Explorer 9 onwards.**

The Web server is enabled when shipped. Perform the following steps to launch the Web server:

1. Establish a LAN interface connection from your computer to the ac source.
2. To launch the Web server, enter the instrument's IP address or fully-qualified hostname into the browser's Address field.
3. To begin controlling your instrument, click the **Browser Web Control** button in the navigation bar on the left.
4. For additional help, click the “Help Page” button on the navigation bar.

GPIB Capabilities of the AC Source

All ac source functions except for setting the GPIB address are programmable over the GPIB.

GPIB address

The ac source operates from a GPIB address that is set from the front panel. To set the GPIB address, press the **Address** key on the front panel and enter the address using the Entry keys.

RS-232 Capabilities of the AC Source

The ac source provides an RS-232 programming interface, which is activated by commands located under the front panel **Address** key. All SCPI commands are available through RS-232 programming.

The EIA RS-232 Standard defines the interconnections between Data Terminal Equipment (DTE) and Data Communications Equipment (DCE). The ac source is designed to be a DTE. It can be connected to another DTE such as a PC COM port through a null modem cable.

NOTE The RS-232 settings in your program must match the settings specified in the front panel Address menu. Press the front panel **Address** key if you need to change the settings.

RS-232 data format

The RS-232 data is an 11-bit word with one start bit and two stop bits. The number of start and stop bits is not programmable. The following parity options are selectable using the front panel **Address** key:

EVEN	Seven data bits with even parity
ODD	Seven data bits with odd parity
NONE	Eight data bits without parity

Parity options are stored in non-volatile memory.

Baud rate

The front panel **Address** key lets you select one of the following baud rates, which is stored in non-volatile memory:1200, 2400, 4800, or 9600.

NOTE The ac source must have the same baud rate and parity as the controller.

RS-232 Programming Example

The following program illustrates how to program the ac source using RS-232 to set the output voltage and frequency and to read back the model number and output voltage.

```
' Program to write and read via RS232
' Configure serial port (e.g. COM1) for:
' Bits per second: select 9600 baud (or desired baudrate)
' Start bits: select 1
' Data bits: select 8
' Stop bits: select 2
' Parity: select None
' Flow control: select None
' Send line feed [Enter]
' Reserve 1000 character buffer for serial I/O
'
' Ensure identical baud and parity settings on ac source
  (Default - Baudrate: 9600, Parity: None)
'
' Send SCPI commands

*RST            ' Resets the ac source
VOLT 60         ' Set voltage to 60 volts
FREQ 50         ' Set frequency to 50 hertz
OUTPUT ON       ' Turn on the output
*IDN?           ' Query the ac source identification string
[Wait for response]
MEAS:VOLT:AC?   ' Query the ac source voltage
```

```
[Wait for response]
' END           ' End of main program
```

RS-232 troubleshooting

If you are having trouble communicating over the RS-232 interface, check the following:

- The computer and the ac source must be configured for the same baud rate, parity, and number of data bits. Note that the ac source is configured for 1 start bit and 2 stop bits (these values are fixed).
- The correct interface cables or adapters must be used, as described under “RS-232 connector” in the *User's Guide*. Note that even if the cable has the proper connectors for your system, the internal wiring may be incorrect.
- The interface cable must be connected to the correct serial port on your computer (COM1, COM2, etc.).

Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over the GPIB. 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. For example, you would use the same DISPlay command to control the ac source display and the display of a SCPI-compatible multimeter.

Conventions used in this guide

<table>
<tr><td>Angle brackets</td><td>< ></td><td>Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.</td></tr>
<tr><td>Vertical bar</td><td>|</td><td>Vertical bars separate alternative parameters. For example, NORM | TEXT indicates that either "TEXT" or "NORM" can be used as a parameter.</td></tr>
<tr><td>Square Brackets</td><td>[ ]</td><td>Items within square brackets are optional. The representation [SOURce:]LIST means that SOURce: may be omitted.</td></tr>
<tr><td>Braces</td><td>{ }</td><td>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.</td></tr>
<tr><td><code>Computer</code> font</td><td></td><td>Computer font is used to show program lines in text. <code>TRIGger:DELay .5</code> shows a program line.</td></tr>
</table>

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, "SCPI Command Tree” for the ac source SCPI tree structure.

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”.

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. The following figure shows a portion of a subsystem command tree, from which you access the commands located along the various paths (you can refer to the complete tree in Appendix A, "SCPI Command Tree”).

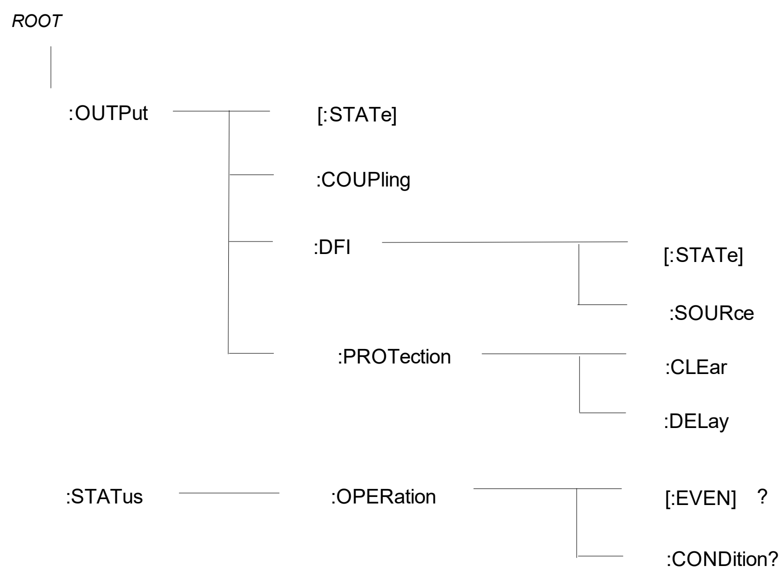

Figure 2-1 Partial command tree

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
- 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:

`OUTPut:PROTection:CLEAr` and

`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 subsytems.

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.

```
VOLTage:TRIGger 7.5;INITialize;*TRG
```

```
OUTPut OFF;*RCL 2;OUTPut ON
```

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.

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 voltage, voltage offset, and function shape commands
- the step, pulse, and list commands that control output voltages, voltage offsets, and function shapes
- the pulse commands that program the width, duty cycle, period, and the hold parameter
- the voltage range and current limit commands in some ac source models

As explained later in Chapter 5, “Programming Examples”, the order in which data is sent by these coupled commands can be important when more than one parameter is changed.

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).

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>
```

Combining message units

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

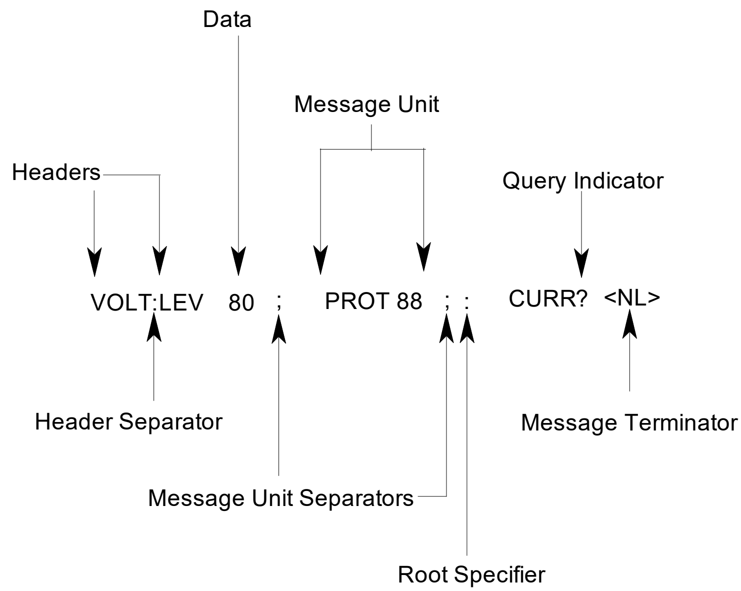

Figure 2-2 Command message structure

The basic parts of the above message are as follows:

Message component	Example
Headers	VOLT LEV PROT CURR
Header separator	The colon in VOLT:LEV
Data	8088
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.

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.

NOTE Short form headers result in faster program execution.

Header convention	In the command descriptions in Chapter 3, “Language Dictionary” of this guide, 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” on page 39, if you combine two or more message units into a compound message, you may need to enter the optional header.

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).

Message unit separator

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

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.

Message terminator

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

- 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 guide, 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.

SCPI Data Formats

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

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 | 1 or OFF | ON (0 = OFF; 1 = ON)
Listening formats	
<Nrf>	Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273273. 2.73E2
<Nrf+>	Expanded decimal format that includes <NRf> and MINMAX. Examples: 273 73.2 .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

Suffixes and multipliers

Class	Suffix	Unit	Unit with multiplier
Current	A	ampere	mA (milliampere)
Amplitude	V	volt	mV (millivolt)
Time	s	second	ms (millisecond)
Frequency	Hz	hertz	kHz (kilohertz)
		Common multipliers	
	1E3	K	kilo
	1E–3	m	milli
	1E–6	u	micro

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.

System Considerations

The remainder of this chapter addresses some system issues concerning programming. These are ac source addressing and the use of the following types of GPIB system interfaces:

- HP Vectra PC controller with Keysight 82335A GPIB Interface Command Library
- IBM PC controller with National Instruments GPIB-PCII Interface/Handler
- Keysight controller with Keysight BASIC Language System

Assigning the GPIB address in programs

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 following examples assume that the GPIB select code is 7, and the ac source will be assigned to the variable ACS.

```
1070 ACS=706                        ! Keysight 82335A Interface
1070 ASSIGN @ACS TO 706             ! Keysight BASIC Interface
```

For systems using the National Instruments DOS driver, the address is specified in the software configuration program (IBCONFIG.EXE) and assigned a symbolic name. The address then is referenced only by this name within the application program (refer to the National Instruments GPIB documentation).

Types of DOS drivers

The Keysight 82335A and National Instruments GPIB are two popular DOS drivers. Each is briefly described here. Refer to the software documentation supplied with the driver for more details.

Keysight 82335A driver

For GW-BASIC programming, the GPIB library is implemented as a series of subroutine calls. To access these subroutines, your application program must include the header file SETUP.BAS, which is part of the DOS driver software.

SETUP.BAS starts at program line 5 and can run up to line 999. Your application programs must begin at line 1000. SETUP.BAS has built-in error checking routines that provide a method to check for GPIB errors during program execution. You can use the error-trapping code in these routines or write your own code using the same variables as used by SETUP.BAS.

National Instruments GPIB driver

Your program must include the National Instruments header file DECL.BAS. This contains the initialization code for the interface. Prior to running any applications programs, you must set up the interface with the configuration program (IBCONF.EXE).

Your application program will not include the ac source symbolic name and GPIB address. These must be specified during configuration (when you run IBCONF.EXE). Note that the primary address range is from 0 to 30 but any secondary address must be specified in the address range of 96 to 126. The instrument expects a message termination on EOI or line feed, so set end of identify (EOI) with last byte of Write. It is also recommended that you set Disable Auto Serial Polling.

All function calls return the status word IBSTA%, which contains a bit (ERR) that is set if the call results in an error. When ERR is set, an appropriate code is placed in variable IBERR%. Be sure to check IBSTA% after every function call. If it is not equal to zero, branch to an error handler that reads IBERR% to extract the specific error.

Error handling

If there is no error-handling code in your program, undetected errors can cause unpredictable results. This includes “hanging up” the controller and forcing you to reset the system. Both of the above DOS drivers have routines for detecting program execution errors.

NOTE

Use error detection after every call to a subroutine.

Keysight BASIC controllers

The Keysight BASIC Programming Language provides access to GPIB functions at the operating system level. This makes it unnecessary to have the header files required in front of DOS applications programs. Also, you do not have to be concerned about controller “hangups” as long as your program includes a timeout statement. Because the ac source can be programmed to generate Service Request (SRQ) on errors, your program can use an SRQ service routine for decoding detected errors. The detectable errors are listed in Appendix C, "Error Messages”.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

3 Language Dictionary

This chapter provides a dictionary of the available SCPI commands.

Introduction

This section gives the syntax and parameters for all the IEEE 488.2 SCPI commands and the Common commands used by the ac sources when operating in Normal mode. It is assumed that you are familiar with the material in Chapter 2, "Introduction to Programming". Because the SCPI syntax remains the same for all programming languages, the examples given for each command are generic.

Syntax forms	Syntax definitions use the long form, but only short form headers (or “keywords”) appear in the examples. Use the long form to help make your program self-documenting.
Parameters	Most commands require a parameter and all queries will return a parameter.The range for a parameter may vary according to the model of ac source. Parameters for all models are listed in the Specifications table in the *User's Guide*.
Models	If a command only applies to specific models, those models are listed in the <Model> only entry. If there is no <Model> only entry, the command applies to all models.
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.
Order of presentation	The dictionary is organized as follows: – Subsystem commands, arranged by subsystem – IEEE 488.2 common commands

Subsystem Commands

Subsystem commands are specific to ac source functions. They can be a single command or a group of commands. The groups comprise 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:

Calibration Subsystem Commands

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

– Enable and disable the calibration mode
– Change the calibration password
– Calibrate the current and voltage output levels, and store new calibration constants in non-volatile memory.

Subsystem syntax

CALibrate | CALibration	
:COUNt?	Returns the calibration count of the ac source
:CURRent	
:AC	Begin AC current programming calibration sequence
:EXTernal	Initiates calibration of the external current measurement
:MEASure	Begin current measurement calibration sequence
:DATA <n>	Input a calibration measurement
:DATA?	Returns the present value of the phase calibration
:DATE <quoted_string>	Input a calibration date
:IMPedance	Begin output impedance calibration sequence
:LEVel <level>	Advance to next calibration step (P1 | P2 | P3 | P4)
:PASSword <n>	Set calibration password
:PHASe	
:AUXiliary	Initiates calibration of the auxiliary output phase
:PWM	
:FREQuency <n>	Trim pulse width modulator frequency
:RAMP <n>	Trim pulse width modulator ramp
:SAVE	Save new cal constants in non-volatile memory
:SECure	
:CODE <quoted_string>	Set calibration password
:STATe <bool>[, <n>]	Unsecures or secures the instrument for calibration
:SHUNt <n>	Allows you to enter the external current shunt value
:STATE <bool>[, <n>]	Enable or disable calibration mode
:STRing <quoted_string>	Stores a message in calibration memory

CALibrate | CALibration	
:VOLTage	
:AC	Begin AC voltage calibration sequence
:AUXiliary	
[:AC]	Initiates calibration of the auxiliary output voltage
:DC	Begin DC voltage calibration sequence
:EXTernal	Initiates calibration of the external voltage measurement
:OFFSet	Begin offset voltage calibration sequence
:PROTection	Begin voltage protection calibration sequence

CALibrate:COUNt?

CALibration:COUNt?

Returns the calibration count of the ac source. Read and record the initial count when you first receive your instrument. The value increments each time you store the calibration constants, as such a complete calibration adds many counts. The calibration count also increments whenever you save a calibration string, change the calibration password, or override the calibration security. You can perform this query regardless of whether the instrument is secured. This setting is non-volatile; it will not be changed by power cycling, a factory reset (*RST), or an instrument preset (SYSTem:PRESet).

Query syntax	CALibrate:COUNt? CALibration:COUNt?
Parameters	None
Examples	`CAL:COUN?`
Returned parameters	<NR1>
Related commands	SYST:SEC:COUN

CALibrate:CURRent:AC

CALibration:CURRent:AC

This command can only be used in the calibration mode. It initiates the calibration of the AC current limit and metering circuits.

Command syntax	CALibrate:CURRent:AC CALibration:CURRent:AC
Parameters	None
Examples	`CAL:CURR:AC`
Related commands	CAL:STAT CAL:SAVE CAL:LEV

CALibrate:CURRent:EXTernal

CALibration:CURRent:EXTernal

Keysight 6811C/6812C/6813C Option 020 only.

Initiates calibration of the external current measurement.

Command syntax	CALibrate:CURRent:EXTernal CALibration:CURRent:EXTernal
Parameters	None
Examples	`CAL:CURR:EXT`

CALibrate:CURRent:MEASure

CALibration:CURRent:MEASure

This command is used to initiate the calibration of the current metering circuits and the peak current limit circuits. It can only be used in the calibration mode.

Command syntax	CALibrate:CURRent:MEASure CALibration:CURRent:MEASure
Parameters	None
Examples	`CAL:CURR:MEAS`
Related commands	CAL:STAT CAL:SAVE CAL:LEV

CALibrate:DATA

CALibration:DATA

This command is only used in calibration mode. It enters a calibration value that you obtain by reading an external meter. You must first select a calibration level (with CALibrate:LEVel) for the value being entered. These constants are not stored in non-volatile memory until they are saved with CALibrate:SAVE. If CALibrate:STATE OFF is programmed without a CALibrate:SAVE, the previous calibration constants are restored.

Command syntax	CALibrate:DATA <NRf> CALibration:DATA <NRf>
Parameters	<external_reading>
Examples	`CAL:DATA 3222.3 MA` `CAL:DATA 5.000`
Related commands	CAL:STAT CAL:SAVE

CALibrate:DATA?

CALibration:DATA?

Keysight 6812/6813C Option 026 only.

Returns the present value of the phase calibration. This command can only be used during the auxiliary phase calibration procedure. It will return an error if it is used at any other time.

Query syntax	CALibrate:DATA? CALibration:DATA?
Returned parameters	<NR3>
Examples	`CAL:DATA?`

CALibrate:DATE

CALibration:DATE

Stores a message in calibration memory with a string input of 11 characters. Storing a calibration message overwrites the previous message. You can perform this query regardless of whether the instrument is secured. This setting is non-volatile; it will not be changed by power cycling or *RST or SYSTem:PRESet.

Command syntax	CALibrate:DATE <quoted_string> CALibration:DATE <quoted_string>
Parameters	<quoted_string>
Examples	`CAL:DATE “14-Jun-2016”`
Query syntax	CALibrate:DATE? CALibration:DATE?
Returned parameters	<SRD>
Related commands	CAL:STR

CALibrate:IMPedance

CALibration:IMPedance

This command can only be used in calibration mode. It calibrates the output impedance circuits. The command automatically performs the calibration. CALibrate:SAVE is required to store the impedance constant in non-volatile memory. CALibrate:IMPedance is a sequential command that takes several seconds to complete.

Command syntax	CALibrate:IMPedance CALibration:IMPedance
Parameters	None
Examples	`CAL:IMP`
Related commands	CAL:STAT CAL:SAVE

CALibrate:LEVel

CALibration:LEVel

This command can only be used in calibration mode. It is used to advance to the next state in the calibration sequence.

Command syntax	CALibrate:LEVel <level> CALibration:LEVel <level>
Parameters	P1 | P2 | P3 | P4
Examples	`CAL:LEV P2`
Related commands	CAL:STAT CAL:SAVE

CALibrate:PASSword

CALibration:PASSword

This command can only be used in calibration mode. It allows you to change the calibration password. A new password is automatically stored in non-volatile memory and does not have to be stored with CALibrate:SAVE. If the password is set to 0, password protection is removed and the ability to enter the calibration mode is unrestricted.

Command syntax	CALibrate:PASSword <NRf> CALibration:PASSword <NRf>
Parameters	0 (default)
Examples	`CAL:PASS 6812` `CAL:PASS 02.1997`
Related commands	CAL:STAT CAL:SEC:CODE

CALibrate:PHASe:AUXiliary

CALibration:PHASe:AUXiliary

Keysight 6812C/6813C Option 026 only.

Initiates calibration of the auxiliary output phase.

Command syntax	CALibrate:PHASe:AUXiliary <NRf> CALibration:PHASe:AUXliary <NRf>
Parameters	0 through 7
Examples	`CAL:PHAS:AUX 1`

CALibrate:PWM:FREQuency

CALibration:PWM:FREQuency

This command is only used during manufacture or repair. It trims the switching frequency of the power output stages. The numbers from 0 to 7 are internally mapped to 8 discrete frequencies.

Command syntax	CALibrate:PWM:FREQuency <NRf> CALibration:PWM:FREQuency <NRf>
Parameters	0 through 7
Examples	`CAL:PWM:FREQ 1`
Query syntax	CALibrate:PWM:FREQuency? CALibration:PWM:FREQuency?
Returned parameters	<NR1>
Related commands	CAL:PWM:RAMP

CALibrate:PWM:RAMP

CALibration:PWM:RAMP

This command modulates the slope of voltage ramp driving the power output stages. Varying the ramp affects the harmonic distortion of the output. The argument is a number from 0 to 255. This command is only used during manufacture or repair of the ac source.

Command syntax	CALCulate:PWM:RAMP <NRf> CALibration:PWM:RAMP <NRf>
Parameters	0 through 255
Examples	`CAL:PWM:RAMP 100`
Query syntax	CALibrate:PWM:RAMP? CALibration:PWM:RAMP?
Returned parameters	<NR1>
Related commands	CAL:PWM:FREQ

CALibrate:SAVE

CALibration: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 non-volatile memory.

Command syntax	CALibrate:SAVE CALibration:SAVE
Parameters	None
Examples	`CAL:SAVE`
Related commands	CAL:CURR CAL:VOLT CAL:STAT

CALibrate:SECure:CODE

CALibration:SECure:CODE

This command can only be used in calibration mode. It allows you to change the calibration password. A new password is automatically stored in non-volatile memory and does not have to be stored with CALibrate:SAVE. If the password is set to 0, password protection is removed and the ability to enter the calibration mode is unrestricted. This setting is non-volatile; it will not be changed by power cycling or *RST or SYSTem:PRESet.

Command syntax	CALibrate:SECure:CODE <NRf> CALibration:SECure:CODE <NRf>
Parameters	0 (default)
Examples	`CAL:SEC:CODE 6813` | `CAL:SEC:CODE 02.1997`
Related commands	CAL:PASS

CALibrate:SECure:STATe

CALibration:SECure:STATe

This command enables and disables calibration mode. The calibration mode must be enabled before the ac source will accept any other calibration commands. The first parameter specifies the enabled or disabled state. The second parameter is the password. It is required if the calibration mode is being enabled and the existing password is not 0. If the password is not entered or is incorrect, an error is generated and the calibration mode remains disabled. The query statement returns only the state, not the password.

Whenever the calibration state is changed from enabled to disabled, any new calibration constants are lost unless they have been stored with CALibrate:SAVE. Unsecures or secures the ac source for calibration.

Command syntax	CALibrate:SECure:STATe <bool>[, <NRf>] CALibration:SECure:STATe <bool>[, <NRf>]
Parameters	0 | 1 | OFF | ON [, <password>]
***RST value**	OFF
Examples	`CAL:SEC:STAT 1, 6812 | CAL:SEC:STAT OFF`
Query syntax	CALibrate:SECure:STATe? CALibration:SECure:STATe?
Returned parameters	0 | 1
Related commands	CAL:STAT

CALibrate:SHUNt

CALibration:SHUNt

Keysight 6811C/6812C/6813C Option 020 only.

Allows you to enter the external current shunt value (in ohms). The external current shunt is used for external current measurement. The programming range is 1E-6 to 1E6. This parameter is non-volatile, so there is no *RST value. As with other CAL commands, calibration must first be enabled with CAL:STATe, and the value must be made permanent with CAL:SAVE.

Command syntax	CALibrate:SHUNt <NRf+> CALibration:SHUNt <NRf+>
Parameters	1E-6 to 1E6 | MINimum | MAXimum
Unit	Ω (ohms)
Examples	`CAL:SHUN .01`

CALibrate:STATe

CALibration:STATe

This command enables and disables calibration mode. The calibration mode must be enabled before the ac source will accept any other calibration commands. The first parameter specifies the enabled or disabled state. The second parameter is the password. It is required if the calibration mode is being enabled and the existing password is not 0. If the password is not entered or is incorrect, an error is generated and the calibration mode remains disabled. The query statement returns only the state, not the password.

Whenever the calibration state is changed from enabled to disabled, any new calibration constants are lost unless they have been stored with CALibrate:SAVE.

Command syntax	CALibrate:STATe <bool>[, <NRf>] CALibration:STATe <bool>[, <NRf>]
Parameters	0 | 1 | OFF | ON [, <password>]
***RST value**	OFF
Examples	`CAL:STAT 1, 6812 | CAL:STAT OFF`
Query syntax	CALibrate:STATe? CALibration:STATe?
Returned parameters	<NR1>
Related commands	CAL:PASS CAL:SAVE

CALibrate:STRing

CALibration:STRing

Stores a message in calibration memory. Common messages include the last calibration date, calibration due date, or calibration department contact information. You can perform this query regardless of whether the ac source is secured. Storing a calibration message overwrites the previous message. This setting is non-volatile; it will not be changed by power cycling or *RST or SYSTem:PRESet.

Command syntax	CALibrate:STRing <quoted_string> CALibration:STRing <quoted_string>
Parameters	<quoted_string>
***RST value**	OFF
Examples	`CAL:STR “CAL BY JOHN [EXT: 1234]”`
Query syntax	CALibrate:STRing? CALibration:STRing?
Returned parameters	<SRD>
Related commands	CAL:DATE

CALibrate:VOLTage:AC

CALibration:VOLTage:AC

This command can only be used in calibration mode. It initiates the calibration of the AC voltage programming and metering circuits.

Command syntax	CALibrate:VOLTage:AC CALibration:VOLTage:AC
Parameters	None
Examples	`CAL:VOLT:AC`
Related commands	CAL:SAVE CAL:STAT

CALibrate:VOLTage:AUXiliary

CALibration:VOLTage:AUXiliary

Keysight 6812C/6813C Option 026 only.

Initiates calibration of the auxiliary output voltage.

Command syntax	CALibrate:VOLTage:AUXiliary[:AC] CALibration:VOLTage:AUXiliary[:AC]
Parameters	None
Examples	`CAL:VOLT:AUX`

CALibrate:VOLTage:DC

CALibration:VOLTage:DC

This command can only be used in calibration mode. It initiates the calibration of the DC voltage programming circuits.

Command syntax	CALibrate:VOLTage:DC CALibration:VOLTage:DC
Parameters	None
Examples	`CAL:VOLT:DC`
Related commands	CAL:SAVE CAL:STAT

CALibrate:VOLTage:EXTernal

CALibration:VOLTage:EXTernal

Keysight 6811C/6812C/6813C Option 020 only.

Initiates calibration of the external voltage measurement.

Command syntax	CALibrate:VOLTage:EXTernal CALibration:VOLTage:EXTernal
Parameters	None
Examples	`CAL:VOLT:EXT`

CALibrate:VOLTage:OFFSet

CALibration:VOLTage:OFFSet

This command can only be used in calibration mode. It initiates the calibration of the offset voltage programming circuits.

Command syntax	CALibrate:VOLTage:OFFSet CALibration:VOLTage:OFFSet
Parameters	None
Examples	`CAL:VOLT:OFFS`
Related commands	CAL:SAVE CAL:STAT CAL:LEV

CALibrate:VOLTage:PROTection

CALibration:VOLTage:PROTection

This command can only be used in calibration mode. It calibrates the overvoltage protection (OV) circuit. The command automatically performs the calibration. CALibrate:SAVE is required to store the new OV constant in non-volatile memory. CALibrate:VOLTage:PROTection is a sequential command that takes several seconds to complete.

Command syntax	CALibrate:VOLTage:PROTection CALibration:VOLTage:PROTection
Parameters	None
Examples	`CAL:VOLT:PROT`
Related commands	CAL:SAVE CAL:STAT

Display Subsystem Commands

This subsystem programs the front panel display of the ac source.

Subsystem syntax

DISPlay	
[:WINDow]	
[:STATe] <bool>	Enable/disable front panel display
:MODE <mode>	Set display mode (NORM | TEXT)
:TEXT	
[:DATA] <quoted_string>	Set text displayed in text mode
:CLEar	Clears text string set on DISP:TEXT

DISPlay

This command turns the front panel display on and off. It does not affect the annunciators.

Command syntax	DISPlay[:WINDow][:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	ON
Examples	`DISP:STAT 1` `DISP:STAT OFF`
Query syntax	DISPlay[:WINDow]:STATe?
Returned parameters	0 | 1
Related commands	DISP:MODE DISP:TEXT

DISPlay:MODE

This command sets the display to show either normal instrument functions, or to show a text message. Text messages are defined with DISPlay:TEXT:DATA.

Command syntax	DISPlay[:WINDow]:MODE <mode>
Parameters	NORMal | TEXT
***RST value**	NORMal
Examples	`DISP:MODE TEXT`
Query syntax	DISPlay[:WINDow]:MODE?
Returned parameters	<CRD>
Related commands	DISP DISP:TEXT

DISPlay:TEXT

This command sets the character string that is displayed when the display mode is set to TEXT. The argument is a quoted string limited to upper case alpha characters and numbers. The display is capable of showing up to 14 characters. If the string exceeds the display capacity, it will be truncated.

Command syntax	DISPlay[:WINDow]:TEXT[:DATA] <quoted_string>
Parameters	<quoted_string>
***RST value**	Null string
Examples	`DISP:TEXT "DO TEST1"`
Query syntax	DISPlay[:WINDow]:TEXT?
Returned parameters	<SRD> (the last programmed string)
Related commands	DISP DISP:MODE

DISPlay:TEXT:CLEar

Clears text string set on DISP:TEXT.

Command syntax	DISPlay[:WINDow]:TEXT:CLEar
Parameters	None
Examples	`DISP:TEXT:CLE`
Related commands	DISP:MODE DISP:TEXT

LXI Subsystem Commands

This subsystem supports LXI functionality.

Subsystem syntax

LXI	
:IDENtify	
[:STATe] <bool>	Shows or hides the LXI Web Identify indicator on the display.
:MDNS	
[:STATe] <bool>	Disables or enables the mDNS.
:ENABle<bool>	Disables or enables the mDNS.
:HNAMe	
[:RESolved]?	Returns the resolved (unique) mDNS hostname.
:SNAMe	
:DESired <quoted_string>	Sets the desired mDNS service name.
[:RESolved]?	Returns the desired mDNS service name.
:RESet	Resets LAN settings to a known operating state.
:RESTart	Restarts the LAN with the specified settings.

LXI:IDENtify[:STATe]

Shows or hides the LXI Web Identify indicator on the display. The indicator helps you identify the device associated with the LAN address. Press the **[Local]** key or send *RST to turn off the indicator.

Command syntax	LXI:IDENtify[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`LXI:IDEN:STAT ON` `LXI:IDEN 0`
Query syntax	LXI:IDENtify[:STATe]?
Returned parameters	0 | 1

LXI:MDNS[:STATe]

Disables or enables the multicast Domain Name System (mDNS), which provides the capabilities of a DNS server for service discovery in a small network without a DNS server. This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet. This parameter is set to its default value when the ac source is shipped from the factory and after SYSTem:SECurity:IMMediate.

Command syntax	LXI:MDNS[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	ON
Examples	`LXI:MDNS ON` `LXI:MDNS 1`
Query syntax	LXI:MDNS[:STATe]?
Returned parameters	0 | 1
Related commands	LXI:MDNS:ENAB

LXI:MDNS:ENABle

Disables or enables the multicast Domain Name System (mDNS), which provides the capabilities of a DNS server for service discovery in a small network without a DNS server. This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet. This parameter is set to its default value when the ac source is shipped from the factory and after SYSTem:SECurity:IMMediate.

Command syntax	LXI:MDNS:ENABle <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	ON
Examples	`LXI:MDNS:ENAB ON` `LXI:MDNS:ENAB 0`
Query syntax	LXI:MDNS:ENABle?
Returned parameters	0 | 1
Related commands	LXI:MDNS

LXI:MDNS:HNAMe[:RESolved]?

Returns the resolved (unique) mDNS hostname in the form K-<model number>-<serial>-N, where <serial> is the last 5 digits of the ac source's serial number. N is an integer appended if necessary, so that the name is unique. The desired name may be truncated, if necessary, to make room for the appended integer.

Query syntax	LXI:MDNS:HNAMe[:RESolved]?
Parameters	None
Examples	`LXI:MDNS:HNAM?`
Returned parameters	<SRD>
Related commands	LXI:MDNS:SNAM

LXI:MDNS:SNAMe:DESired

Sets the desired mDNS service name. This setting is non-volatile; it is not changed by power cycling, a *RST, or SYSTem:PRESet. This parameter is set to its default value when the ac source is shipped from the factory and after SYSTem:SECurity:IMMediate.

Command syntax	LXI:MDNS:SNAMe:DESired <quoted_string>
Parameters	<quoted_string>
Examples	`LXI:MDNS:SNAM:DES “Keysight 6811C ACS”`
Query syntax	LXI:MDNS:SNAMe:DESired?
Returned parameters	<SRD>
Related commands	LXI:MDNS:SNAM:RES

LXI:MDNS:SNAMe[:RESolved]?

The resolved mDNS service name will be the desired service name (LXI:MDNS:SNAMe:DESired), possibly with "(<N>)" appended. The N is an integer appended if necessary, to make the name unique. The desired name may be truncated, if necessary, to make room for the appended integer.

Query syntax	LXI:MDNS:SNAMe[:RESolved]?
Parameters	None
Examples	`LXI:MDNS:SNAM?`
Returned parameters	<SRD>
Related commands	LXI:MDNS:HNAM

LXI:RESet

Resets LAN settings to a known operating state, beginning with DHCP, and clears the Web Interface password. If DHCP fails, it uses Auto-IP. Depending on your network, the LAN interface may take several seconds to restart after this command is sent.

Command syntax	LXI:RESet
Parameters	None
Examples	`LXI:RES`
Related commands	LXI:REST

LXI:RESTart

Restarts the LAN with the current settings as specified by the SYSTem:COMMunicate:LAN commands. Depending on your network, the LAN interface may take several seconds to restart after this command is sent.

Command syntax	LXI:RESTart
Parameters	None
Examples	`LXI:REST`
Related commands	LXI:RES

Measurement Subsystem (Arrays) Commands

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.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

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
: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?

These queries return an array containing the instantaneous output current in amperes. 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 25 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.

Query syntax	MEASure:ARRay:CURRent[:DC]? FETCh:ARRay:CURRent[:DC]?
Parameters	None
Examples	`MEAS:ARR:CURR?` `FETC:ARR:CURR?`
Returned parameters	4096 NR3 values
Related commands	MEAS:ARR:VOLT?

MEASure:ARRay:CURRent:HARMonic?

FETCh:ARRay:CURRent:HARMonic?

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 12.6 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:ARR:VOLT:HARM?	MEAS:ARR:CURR:HARM:PHAS?

MEASure:ARRay:CURRent:HARMonic:PHASe?

FETCh:ARRay:CURRent:HARMonic:PHASe?

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 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 kHz are returned as 0.

Query syntax	MEASure:ARRay:CURRent:HARMonic:PHASe? FETCh:ARRay:CURRent:HARMonic:PHASe?	
Parameters	None	
Examples	`MEAS:ARR:CURR:HARM:PHAS?`	`FETC:ARR:CURR:HARM:PHAS?`
Returned parameters	51 NR3 values	
Related commands	MEAS:ARR:VOLT:HARM:PHAS?	MEAS:ARR:CURR:HARM?

MEASure:ARRay:VOLTage?

FETCh:ARRay:VOLTage?

These queries return an array containing the instantaneous output voltage in volts.

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 25 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.

Query syntax	MEASure:ARRay:VOLTage[:DC]? FETCh:ARRay:VOLTage[:DC]?
Parameters	None
Examples	`MEAS:ARR:VOLT?` `FETC:ARR:VOLT?`
Returned parameters	4096 NR3 values
Related commands	MEAS:ARR:CURR?

MEASure:ARRay:VOLTage:HARMonic?

FETCh:ARRay:VOLTage:HARMonic?

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 12.6 kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:ARR:CURR:HARM?	MEAS:ARR:VOLT:HARM:PHAS?

MEASure:ARRay:VOLTage:HARMonic:PHASe?

FETCh:ARRay:VOLTage:HARMonic:PHASe?

These queries return an array of harmonic phases of output voltage 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 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 kHz are returned as 0.

Query syntax	MEASure:ARRay:VOLTage:HARMonic:PHASe?	
	FETCh:ARRay:VOLTage:HARMonic:PHASe?	
Parameters	None	
Examples	`MEAS:ARR:VOLT:HARM:PHAS?`	`FETC:ARR:VOLT:HARM:PHAS?`
Returned parameters	51 NR3 values	
Related commands	MEAS:ARR:CURR:HARM:PHAS?	MEAS:ARR:VOLT:HARM?

Measurement Subsystem (Current) Commands

This subsystem programs the current measurement capability of the ac 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.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

Subsystem syntax

MEASure | FETCh	
[:SCALar]	
:CURRent	
[:DC]?	Returns DC component of the current
:AC?	Returns AC rms current
:ACDC?	Returns AC+DC rms current
:AMPLitude	
:MAXimum?	Returns peak current
:CREStfactor?	Returns current crest factor
: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

MEASure:CURRent?

FETCh:CURRent?

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	MEAS:VOLT? MEAS:CURR:AC?

MEASure:CURRent:AC?

FETCh:CURRent:AC?

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	MEAS:VOLT:AC? MEAS:CURR?

MEASure:CURRent:ACDC?

FETCh:CURRent:ACDC?

These queries return the AC+DC 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	MEAS:VOLT:ACDC? MEAS:CURR:AMPL:MAX?

MEASure:CURRent:AMPLitude:MAXimum?

FETCh:CURRent:AMPLitude:MAXimum?

These queries return the absolute value of the peak current as sampled over one measurement acquisition of 4096 data points.

Query syntax	MEASure:[SCALar]:CURRent:AMPLitude:MAXimum? FETCh:[SCALar]:CURRent:AMPLitude:MAXimum?
Parameters	None
Examples	`MEAS:CURR:AMPL:MAX?` `FETC:CURR:AMPL:MAX?`
Returned parameters	<NR3>
Related commands	MEAS:CURR:ACDC? MEAS:CURR:CRES?

MEASure:CURRent:CREStfactor?

FETCh:CURRent:CREStfactor?

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

Query syntax	MEASure:[SCALar]:CURRent:CREStfactor? FETCh:[SCALar]:CURRent:CREStfactor?
Parameters	None
Examples	`MEAS:CURR:CRES?` `FETC:CURR:CRES?`
Returned parameters	<NR3>
Related commands	MEAS:CURR:ACDC? MEAS:CURR:AMPL:MAX?

MEASure:CURRent:HARMonic?

FETCh:CURRent:HARMonic?

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 queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:CURR:HARM:PHAS? MEAS:CURR:HARM:THD?

MEASure:CURRent:HARMonic:PHASe?

FETCh:CURRent:HARMonic:PHASe?

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 queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:CURR:HARM? MEAS:CURR:HARM:THD?

MEASure:CURRent:HARMonic:THD?

FETCh:CURRent:HARMonic:THD?

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	MEAS:CURR:HARM? MEAS:CURR:HARM:PHAS?

Measurement Subsystem (Frequency) Commands

This subsystem programs the frequency measurement capability of the ac 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.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

Subsystem syntax

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

MEASure:FREQuency?

FETCh:FREQuency?

This query returns the output frequency in hertz.

For Keysight 6811C/6812C/6813C Option 020 only, when the Power Analyzer input is selected as the measurement source, this command returns the actual frequency measured at the V SENSE input channel. When the Main Output of the ac source is selected as the measurement source, this command returns the programmed output frequency.

NOTE **With no signal connected to the Power Analyzer input, this command may return a very high frequency value. This is normal, since there is no external frequency for the measurement circuit to detect.**

Query syntax	MEASure:[SCALar]:FREQuency? FETCh:[SCALar]:FREQuency?
Parameters	None

Examples	`MEAS:FREQ?`	`FETC:FREQ?`
Returned parameters	<NR3>	

Measurement Subsystem (Power) Commands

This subsystem programs the power measurement capability of the ac 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.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

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

MEASure:POWer?

FETCh:POWer?

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	MEAS:POW:AC?

MEASure:POWer:AC?

FETCh:POWer:AC?

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	MEAS:POW?

MEASure:POWer:AC:APParent?

FETCh:POWer:AC:APParent?

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	MEAS:POW:REAC? MEAS:POW:PFAC?

MEASure:POWer:AC:REACtive?

FETCh:POWer:AC:REACtive?

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

$$\text{VAR} = \sqrt{((\text{Apparent Power})^2 - (\text{Real Power})^2)}$$

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	MEAS:POW:AC:APP? MEAS:POW:PFAC?

MEASure:POWer:AC:PFACtor?

FETCh:POWer:AC:PFACtor?

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

$$\text{PFactor} = \frac{\text{Real Power}}{\text{Apparent Power}}$$

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	MEAS:POW:AC:APP? MEAS:POW:REAC?

Measurement Subsystem (TVOLt) Commands

For Keysight 6811C/6812C/6813C Option 020 only, these measurement commands are optimized for use in specific applications such as evaluating uninterruptible power supplies. In this application the new commands can be used to measure transfer time, peak voltage, and phase locked loop performance of the UPS.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

Subsystem syntax

MEASure | FETCh	
[:SCALar]	
:TVOLt? <level>, <occurrence>	Returns the time
:TVOLt	
:ABSolute? <level>, <start_time>, <min_pulse_width>	Returns the time

MEASure:TVOLt?

FETCh:TVOLt?

Keysight 6811C/6812C/6813C Option 020 only.

Returns the time at which the measured voltage crosses a user-specified voltage *level*. The sign and magnitude of *occurrence* define the event to be reported. For example, if *occurrence* is –3, the return value is the third time at which the voltage crosses the *level* in the negative-going direction. If *occurrence* is 4, the return value is the fourth time at which the voltage crosses the *level* in the positive-going direction.

The return value is with respect to the start of the measurement buffer. If no point on the waveform satisfies the specified conditions, the return value is 9.91000E+37.

Query syntax	MEASure:[SCALar]:TVOLt? <level>, <occurrence> FETCh:[SCALar]:TVOLt? <level>, <occurrence>
Parameters	–1E6 to 1E6 (for level) –4096 to +4096, but not 0 (for occurrence)
Examples	`MEAS:TVOL? 50, 3` `FETC:TVOL? 10, -1`
Returned parameters	<NR3>
Related commands	MEAS:TVOL:ABS?

MEASure:TVOLt:ABSolute?

FETCh:TVOLt:ABSolute?

Keysight 6811C/6812C/6813C Option 020 only.

Returns the time at which the absolute value of the measured voltage first exceeds *level*, with the following qualifications: The behavior of the signal before *start_time* is ignored, and the absolute value of the voltage must remain above level for at least the specified *min_pulse_width*.

Both *start_time* and the return value are with respect to the start of the measurement buffer. If no point on the waveform satisfies the specified conditions, the return value is 9.91000E+37.

Query syntax	MEASure:[SCALar]:TVOLt:ABSolute? <level>, <start_time>, <min_pulse_width> FETCh:[SCALar]:TVOLt:ABSolute? <level>, <start_time>, <min_pulse_width>
Parameters	0 to 1E6 (for level) 0 to 1E6 (for start time in seconds) 0 to 1E6 (for minimum pulse width in seconds)
Examples	`MEAS:TVOL:ABS? 50, 0, 0.001` `FETC:TVOL:ABS? 100, 0.005, 0`
Returned parameters	<NR3>
Related commands	MEAS:TVOL?

Measurement Subsystem (Voltage) Commands

This subsystem programs the voltage measurement capability of the ac 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.

For Keysight 6811C/6812C/6813C Option 020 only, all standard ac source MEASure or FETCh functions are available for the Main Output of the ac source as well as the Power Analyzer input. The SENSe:NSELect command selects the measurement source that will return data.

Subsystem syntax

MEASure | FETCh	
[:SCALar]	
:VOLTage	
[:DC]?	Returns DC component of the voltage
:AC?	Returns AC rms voltage
:ACDC?	Returns AC+DC rms voltage
: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
:RANKed? <n>	Returns the voltage value for the percentile rank given

MEASure:VOLTage?

FETCh:VOLTage?

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	MEAS:CURR? MEAS:VOLT:AC?

MEASure:VOLTage:AC?

FETCh:VOLTage:AC?

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	MEAS:CURR:AC? MEAS:VOLT?

MEASure:VOLTage:ACDC?

FETCh:VOLTage:ACDC?

These queries return the AC+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	MEAS:CURR:ACDC? MEAS:VOLT?

MEASure:VOLTage:HARMonic?

FETCh:VOLTage:HARMonic?

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 queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:VOLT:HARM:PHAS? MEAS:VOLT:HARM:THD?

MEASure:VOLTage:HARMonic:PHASe?

FETCh:VOLTage:HARMonic:PHASe?

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 queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6 kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6 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	MEAS:VOLT:HARM? MEAS:VOLT:HARM:THD?

MEASure:VOLTage:HARMonic:THD?

FETCh:VOLTage:HARMonic:THD?

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	MEAS:VOLT:HARM? MEAS:VOLT:HARM:PHAS?

MEASure:VOLTage:RANKed?

FETCh:VOLTage:RANKed?

Keysight 6811C/6812C/6813C Option 020 only.

Scans through the buffer of 4096 instantaneous voltage measurements, and returns the voltage value that corresponds to the percentile rank given. Specifying a percentile of 0 returns the lowest value in the buffer. A percentile of 100 returns the highest value in the buffer. A percentile of 50 returns the median value of the buffer. Stated another way, if you specify a percentile of 60, 60 percent of the voltage readings in the buffer are less that the returned value, and 40 percent of the readings in the buffer are greater than the returned value.

If the voltage waveform consists of rectangular pulses that have narrow overshoots, using percentiles of approximately 5 and 95 (depending on the width of the pulses and the width of the overshoots) is a good way to determine the voltages at the flat portions of the pulses.

Query syntax	MEASure:[SCALar]:VOLTage:RANKed? <percentile> FETCh:[SCALar]:VOLTage:RANKed? <percentile>
Parameters	0 to 100
Examples	`MEAS:VOLT:RANK? 50` `FETC:VOLT:RANK? 5`
Returned parameters	<NR3>

Output Subsystem Commands

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

Subsystem syntax

```
OUTPut
    [:STATe] <bool>          Enable/disable output voltage, current, power, etc.
    :COUPling <coupling>     Enables AC or DC output coupling (AC | DC)
    :DFI
        [:STATE] <bool>      Enable/disable DFI output
        :SOURce <source>     Selects an event source (QUES | OPER | ESB | RQS | OFF)
    :IMPedance
        [:STATE] <bool>      Enable/disable output impedance programming
        :REAL <n>            Sets resistive part of output impedance
        :REACtive <n>        Sets inductive part of output impedance
    :PON
        :STATe <state>       Set power-on state (RST | RCL0)
    :PROTection
        :CLEar               Reset latched protection
        :DELay <n>           Delay after programming/before protection
    :RI
        :MODE <mode>         Set remote inhibit input (LATC | LIVE | OFF)
    :TTLTrg
        [: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. The query form returns the output state.

For Keysight 6812C/6813C Option 026 only, this command controls the on/off state of both the main and auxiliary outputs.

Command syntax	OUTPut[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`OUTP 1` `OUTP:STAT ON`
Query syntax	OUTPut[:STATe]?
Returned parameters	0 | 1
Related commands	*RCL *SAV

OUTPut:COUPling

This command enables AC or DC output coupling. When the output coupling is set to AC, a DC leveling loop attempts to maintain zero average output voltage. The loop has a corner frequency of about 2 Hz. It will not prevent short transient waveforms that may have non-zero average voltage, but will cause a settling transient to an average value of 0 volts.

The output coupling must be set to DC to obtain DC output with VOLTage:OFFSet, or to generate output transients that have net DC components.

Command syntax	OUTPut:COUPling <coupling>
Parameters	AC | DC
***RST value**	AC
Examples	`OUTP:COUP DC`
Query syntax	OUTPut:COUPling?
Returned parameters	<CRD>
Related commands	*RCL *SAV

OUTPut:DFI

This command enables or disables the discrete fault indicator (DFI) signal to the ac source.

Command syntax	OUTPut:DFI[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`OUTP:DFI 1` `OUTP:DFI ON`
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 as follows:

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

Command syntax	OUTPut:DFI:SOURce <source>
Parameters	QUEStionable | OPERation | ESB | RQS | OFF
***RST value**	OFF
Examples	`OUTP:DFI:SOUR OPER`
Query syntax	OUTPut:DFI:SOURce?
Returned parameters	<CRD>
Related commands	OUTP:DFI

OUTPut:IMPedance

This command enables or disables the output impedance programming capability of the ac source.

Command syntax	OUTPut:IMPedance[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`OUTP:IMP 1` `OUTP:IMP ON`
Query syntax	OUTPut:IMPedance[:STATe]?
Returned parameters	0 | 1
Related commands	OUTP:IMP:REAL OUTP:IMP:REAC

OUTPut:IMPedance:REAL

This command sets the real part of the output impedance of the ac source. OUTPut:IMPedance:STATe must be enabled for the programmed impedance to affect the output.

Command syntax	OUTPut:IMPedance:REAL <NRf>
Parameters	0 to 1
Unit	Ω (ohms)
***RST value**	0
Examples	`OUTP:IMP:REAL 0.25`
Query syntax	OUTPut:IMPedance:REAL?
Returned parameters	<NR3>
Related commands	OUTP:IMP OUTP:IMP:REAC

OUTPut:IMPedance:REACtive

This command sets the real part of the output impedance of the ac source. OUTPut:IMPedance:STATe must be enabled for the programmed impedance to affect the output.

Command syntax	OUTPut:IMPedance:REACtive <NRf>
Parameters	0.00002 to 0.001
Unit	*H* (henrys)
***RST value**	0.00005
Examples	`OUTP:IMP:REAC 100E-6`
Query syntax	OUTPut:IMPedance:REAC?
Returned parameters	<NR3>
Related commands	OUTP:IMP OUTP:IMP:REAL

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), remote inhibit (RI), or power rail 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 *SAV *RCL

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 Questionable Status 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 100 | MAXimum | MINimum
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 *SAV *RCL

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.

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

This command enables or disables the ac source Trigger Out signal, which is available at a BNC connector on the rear of the instrument.

Command syntax	OUTPut:TTLTrg[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`OUTP:TTLT 1` `OUTP:TTLT ON`
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 BNC 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

Sense Subsystem Commands

This subsystem controls the measurement current range, the data acquire sequence, and the harmonic measurement window of the ac source.

For Keysight 6811C/6812C/6813C Option 020 only, the SENSe:NSELect and SENSe:FREQuency:SOURce let you select a measurement source and the frequency measurement channel for the Power Analyzer input. The SENSe:SWEep:TINTerval, SENSe:WINDow, and other measurement time-based commands apply to both measurement sources, regardless of which source is selected.

Subsystem syntax

```
SENSe
    :CURRent
        :ACDC
            :RANGe
                [:UPPer] <n>      Sets measurement current range
      :FREQuency
          :SOURce <source>        Selects the frequency measurement source
      :NSELect <n>                Selects the measurement source that returns the data of a query
      :SWEep
          :OFFSet
              :POINts <n>         Define trigger points relative to the start of the digitizer data record
          :TINTerval <n>          Sets the digitizer sample spacing
    :WINDow
        [:TYPE] <type>            Sets measurement window type (KBES | RECT)
```

SENSe:CURRent:ACDC:RANGe

This command sets the current measurement range. There are two current measurement ranges:

For Keysight 6811C only:

- **High Range**: 0 through 28.5671 A_{rms} (–40.4 A_{peak} through +40.4 A_{peak})
- **Low Range**: 0 through 2.85671 A_{rms} (–4.04 A_{peak} through +4.04 A_{peak})

For Keysight 6812C and Keysight 6813C:

- **High Range**: 0 through 57.1342 A_{rms} (–80.8 A_{peak} through +80.8 A_{peak})
- **Low Range**: 0 through 5.71342 A_{rms} (–8.08 A_{peak} through +8.08 A_{peak})

The high range covers the maximum current measurement capability of the instrument. The low range increases the low current measurement sensitivity by a factor of 10 for greater accuracy and resolution.

The value that you program with SENS:CURR:ACDC:RANG must be the maximum rms current that you expect to measure. Based on this value, the instrument will select the range that gives the best resolution in measuring a sinusoidal waveform of that rms value. The crossover value of the two ranges is 5.71342 A_{rms} (2.85671 A_{rms} for Keysight 6811C).

For Keysight 6811C/6812C/6813C Option 020:

Sets the current measurement range of the selected measurement source (refer to "SENSe:NSELect" on page 114). Normally when using the Main Output as the measurement source, you would put in a value and the ac source will automatically select the current range. If you are using the Power Analyzer input with an external current shunt as the measurement source, the crossover value of the current ranges can be determined with the formula *Crossover_Value = 0.05713/Shunt_Resistance*. Values greater than the crossover value select the high current range. Alternatively, you can simply program MIN or MAX to select the low or high current range.

Command syntax	SENSe:CURRent:ACDC:RANGe[:UPPer] <NRf+>
Parameters	0 through 57.1342 | MINimum | MAXimum (all except Keysight 6811C) 0 through 28.5671 | MINimum | MAXimum (Keysight 6811C only)
Unit	*A* (rms amperes)
***RST value**	MAX (high range)
Examples	`SENS:CURR:ACDC:RANGE MIN`
Query syntax	SENSe:CURRent:ACDC:RANGe?
Returned parameters	<NR3>
Related commands	SENS:SWE:TINT MEAS:ARR

SENSe:FREQuency:SOURce

Keysight 6811C/6812C/6813C Option 020 only.

This command selects the frequency measurement source for the Power Analyzer input. It only applies when the Power Analyzer input is active.

Accurate frequency measurements are critical when making all Harmonic measurement calculations. Other measurement calculations also use frequency measurement values, but they only slightly affect the accuracy of the measurement. The ac source normally measures the frequency at the V SENSE input channel. With this command you can select the I SENSE input channel as the frequency measurement source for the Power Analyzer input.

NOTE For Main Output measurements, the ac source uses the programmed output frequency value when making measurement calculations.

Command syntax	SENSe:FREQuency:SOURce <source>
Parameters	VOLTage | CURRent
***RST value**	VOLTage
Examples	`SENS:FREQ:SOUR CURR`
Query syntax	SENSe:FREQuency:SOURce?
Returned parameters	<CRD>

SENSe:NSELect

Keysight 6811C/6812C/6813C Option 020 only.

Selects the measurement source that will return data when a query is sent. A parameter value of 1 selects the Main Output, which measures the actual output voltage and current of the ac source. A value of 2 selects the Power Analyzer input, which measures the voltage and current at the V SENSE and I SENSE input channels.

Only the following commands are affected by the measurement source selection:

– Queries beginning with MEASure or FETCh

– SENSe:CURRent:ACDC:RANGe

Command syntax	SENSe:NSELect <NR1>
Parameters	1 | 2
***RST value**	1
Examples	`SENS:NSEL 2`
Query syntax	SENSe:NSELect?
Returned parameters	<NR1>

SENSe:SWEep:OFFSet:POINts

This command defines the trigger point 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 (zero)
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 when voltage and current digitization is controlled by the acquire trigger sequence. The sample period can be programmed from 25 to 250 microseconds in 25 microsecond increments.

NOTE All the MEASure commands use the ACQuire trigger sequence implicitly. These commands always set the sample period to 25 microseconds.

Command syntax	SENSe:SWEep:TINTerval <NRf+>
Parameters	25.037 through 250.37 | MAXimum | MINimum
Unit	*µs* (microseconds)
***RST value**	25.049 µs
Examples	`SENS:SWE:TINT 100E-6`
Query syntax	SENSe:SWEep:TINTerval?
Returned parameters	<NR3>
Related commands	SENS:SWE:OFFS:POIN MEAS:ARR

SENSe:WINDow

This command sets the window function which is used in harmonic measurements. KBESsel is the preferred window and should be used for most measurements. RECTangular is available for making harmonic measurements that comply with regulatory requirements for quasi-stationary harmonics.

When RECTangular is selected, the output frequency is constrained to frequencies that give an integer number of cycles in the acquired waveform buffers, and the measurement acquisition time is set to 0.1 seconds. Any programmed output frequency will be routed to the closest frequency that has this attribute. These frequencies are exact multiples of 10.000207 Hz.

Command syntax	SENSe:WINDow[:TYPE] <type>
Parameters	RECTangular | KBESsel
***RST value**	KBESsel
Examples	`SENS:WIND KBES`
Query syntax	SENSe:WINDow?
Returned parameters	<CRD>

Source Subsystem (Current) Commands

This subsystem programs the output current of the ac source.

Subsystem syntax

```
[SOURce:]
  :CURRent
    [:LEVel]
      [:IMMediate]
        [:AMPLitude] <n>  Sets the rms current limit
    :PEAK
      [:IMMediate] <n>    Sets the peak current limit
      :MODE <mode>        Sets peak current limit mode (FIX | STEP | PULS | LIST)
      :TRIGgered <n>
    :PROTection           Sets the transient level for peak current limit
      :STATe <bool>       Enable/Disable rms current limit protection
```

CURRent

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

Command syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>
Parameters	MINimum | MAXimum | 0 to 3.25 (6811C including all options) 0 to 6.50 (6812C including all options) 0 to 13.0 (6813C all options except Option 019) 0 to 16.0 (6813C Option 019)
Unit	*A* (rms amperes)
***RST value**	MAXimum
Examples	`CURR 5` `CURR:LEV .5`
Query syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
Returned parameters	<NR3>
Related commands	CURR:PROT:STAT VOLT:RANG

CURRent:PEAK

This command sets the output limit of the absolute value of peak instantaneous current.

Command syntax	[SOURce:]CURRent:PEAK[:IMMediate] <NRf+>
Parameters	MINimum | MAXimum | 0 to 3.25 (6811C including all options) 0 to 6.50 (6812C including all options) 0 to 13.0 (6813C all options except Option 019) 0 to 16.0 (6813C Option 019)
Unit	*A* (peak amperes)
***RST value**	13 (6811C/6812C including all options) 26 (6813C all options except Option 019) 32 (6813C Option 019)
Examples	`CURR:PEAK:IMM 15`
Query syntax	[SOURce:]CURRent:PEAK[:IMMediate]?
Returned parameters	<NR3>
Related commands	CURR:PEAK:MODE CURR:PEAK:TRIG

CURRent:PEAK:MODE

This command determines how the peak current limit is controlled during a triggered output transient. The choices are as follows:

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

Command syntax	[SOURce:]CURRent:PEAK:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`CURR:PEAK:MODE FIX`
Query syntax	[SOURce:]CURRent:PEAK:MODE?
Returned parameters	<CRD>
Related commands	CURR:PEAK CURR:PEAK:TRIG

CURRent:PEAK:TRIGgered

This command determines how the peak current limit is controlled during a triggered output transient. The choices are as follows:

Command syntax	[SOURce:]CURRent:PEAK:TRIGgered <NRf+>
Parameters	MINimum | MAXimum | 0 to 3.25 (6811C including all options) 0 to 6.50 (6812C including all options) 0 to 13.0 (6813C all options except Option 019) 0 to 16.0 (6813C Option 019)
Unit	*A* (peak amperes)
***RST value**	13 (6811C/6812C including all options) 26 (6813C all options except Option 019) 32 (6813C Option 019)
Examples	`CURR:PEAK:TRIG 15`
Query syntax	[SOURce:]CURRent:PEAK:TRIGgered?
Returned parameters	<NR3>
Related commands	CURR:PEAK CURR:PEAK:MODE

CURRent:PROTection:STATe

This command enables or disables the overcurrent (OC) protection function. If the overcurrent protection function is enabled and the ac source exceeds the programmed level, then the output is disabled and the Questionable Condition status register OC bit is set (refer to “Programming the Status Registers” on page 259 in Chapter 5, "Programming Examples"). An overcurrent condition can be cleared with OUTPut:PROTection:CLEar after the cause of the condition is removed.

NOTE Use OUTP:PROT:DEL to prevent momentary current limit conditions caused by programmed output changes from tripping the overcurrent protection.

Command syntax	[SOURce:]CURRent:PROTection:STATe <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`CURR:PROT:STAT 1` `CURR:PROT:STAT OFF`
Query syntax	[SOURce:]CURRent:PROTection:STATe?
Returned parameters	0 | 1
Related commands	OUTP:PROT:CLE OUTP:PROT:DEL

Source Subsystem (Frequency) Commands

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)
: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	MINimum | MAXimum | 0.001 to 1000 (6811C/6812C/6813C except Option 019) 0.001 to 100 (6813C Option 019)
Unit	*Hz* (hertz)
***RST value**	60
Examples	`FREQ 50`
Query syntax	[SOURce:]FREQuency[:CW | :IMMediate]?
Returned parameters	<NR3>
Related commands	FREQ:MODE FREQ:SLEW

FREQuency:MODE

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

- **FIXed**: The output frequency is unaffected by a triggered output transient.
- **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.

Command syntax	[SOURce:]FREQuency:MODE <mode>
Parameters	FIXed | STEP | PULse | LIST
***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	0 to 9.9E37 | MAXimum | MINimum | INFinity
***RST value**	MAXimum
Examples	`FREQ:SLEW:IMM 75` `FREQ:SLEW MAX`
Query syntax	[SOURce:]FREQuency:SLEW[:IMMediate]?
Returned parameters	<NR3>
Related commands	FREQ FREQ:SLEW:MODE

FREQuency:SLEW:MODE

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

- **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:SLEW:TRIGgered when a triggered transient occurs.
- **PULSe**: The frequency slew rate is changed to the value set by FREQuency:SLEW:TRIGgered for a duration determined by the pulse commands.
- **LIST**: The frequency slew rate is controlled by the frequency slew list when a triggered transient occurs.

Command syntax	[SOURce:]FREQuency:SLEW:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`FREQ:SLEW:IMM 75` `FREQ:SLEW MAX`
Query syntax	[SOURce:]FREQuency:SLEW:MODE?
Returned parameters	<CRD>
Related commands	FREQ FREQ:SLEW:TRIG

FREQuency: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	0 to 9.9E37 | MAXimum | MINimum | INFinity
***RST value**	MAXimum
Examples	`FREQ:SLEW:TRIG 75` `FREQ:SLEW:TRIG MAX`
Query syntax	[SOURce:]FREQuency:SLEW:TRIGgered?
Returned parameters	<NR3>
Related commands	FREQ FREQ:SLEW:MODE

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	MINimum | MAXimum | 0.001 to 1000 (6811C/6812C/6813C except Option 019) 0.001 to 100 (6813C Option 019)
Unit	*Hz* (hertz)
***RST value**	60
Examples	`FREQ:TRIG 50`
Query syntax	[SOURce:]FREQuency:TRIGgered?
Returned parameters	<NR3>
Related commands	FREQ FREQ:MODE

Source Subsystem (Frequency Modulation) Commands

Keysight 6811C/6812C/6813C Option 020 only.

The following SCPI commands are used to program the output frequency of the ac source. These commands apply only to the Main Output of the ac source and do not affect the measurement capability of the unit.

Subsystem syntax

[SOURce:]
FM

[:STATe] <bool>	Turns off or on the frequency modulation of the Main Output
:DEViation <n>	Sets the peak frequency deviation in hertz
:FREQuency <n>	Sets the modulating frequency in hertz

FM

Keysight 6811C/6812C/6813C Option 020 only.

Turns off or on the frequency modulation of the Main Output. Frequency modulation operates only if FREQuency:MODE is set to FIXed.

Command syntax	[SOURce:]FM[:STATe] <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`FM 1` `FM:STATE ON`
Query syntax	[SOURce:]FM[:STATe]?
Returned parameters	0 | 1
Related commands	FM:DEV FM:FREQ

FM:DEViation

Keysight 6811C/6812C/6813C Option 020 only.

Sets the peak frequency deviation of the output frequency modulation in hertz. The frequency deviation must be less than the programmed output frequency by at least 0.001 Hz. For example, if the output frequency is set to 60 Hz, and you program a deviation of 10 Hz, the output frequency will modulate between 50 and 70 Hz at a rate determined by the FM:FREQuency command.

Command syntax	[SOURce:]FM:DEViation <NRf>
Parameters	0 to 1000
Unit	*Hz* (hertz)
***RST value**	0
Examples	`FM:DEV 10`
Query syntax	[SOURce:]FM:DEViation?
Returned parameters	<NR3>
Related commands	FM FM:FREQ

FM:FREQuency

Keysight 6811C/6812C/6813C Option 020 only.

Sets the modulating frequency of the output frequency modulation in hertz.

Command syntax	[SOURce:]FM:FREQuency <NRf>
Parameters	0.001 to 1000
Unit	*Hz* (hertz)
***RST value**	0.1
Examples	`FM:FREQ 1`
Query syntax	[SOURce:]FM:FREQuency?
Returned parameters	<NR3>
Related commands	FM FM:DEV

Source Subsystem (Function) Commands

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 | <table>)
:MODE <mode>	Sets the waveform shape mode (FIX | STEP | PULS | LIST)
:TRIGgered <shape>	Sets the triggered transient shape (SIN | SQU | CSIN | <table>)
:CSINusoid <n> [THD]	Sets the % of peak at which the sine wave clips (or % THD)

FUNCtion

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

- **SINusoid**: A sine wave is the output.
- **SQUare**: A square wave is the output.
- **CSINusoid**: The output is a clipped sine wave. Both positive and negative peak amplitudes are clipped at a value determined by the FUNCtion:CSINusoid command.
- **<table>**: The output shape is described by one of the user-defined waveform tables

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]FUNCtion[:SHAPe][:IMMediate] <shape>
Parameters	SINusoid | SQUare | CSINusoid | <table>
***RST value**	SINusoid
Examples	`FUNC SIN` `FUNC:SHA SQU`
Query syntax	[SOURce:]FUNCtion[:SHAPe][:IMMediate]?
Returned parameters	<CRD>
Related commands	FUNC:MODE FUNC:TRIG VOLT

FUNCtion:MODE

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

- **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 voltage waveform when a triggered step or pulse transient occurs. The parameters are as follows:

- **SINusoid**: A sine wave is the output.
- **SQUare**: A square wave is the output.
- **CSINusoid**: The output is a clipped sine wave. Both positive and negative peak amplitudes are clipped at a value determined by the FUNCtion:CSINusoid command.
- **<table>**: The output shape is described by one of the user-defined waveform tables.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]FUNCtion[:SHAPe]:TRIGgered <shape>
Parameters	SINusoid | SQUare | CSINusoid | <table>
***RST value**	SINusoid
Examples	`FUNC:TRIG SIN` `FUNC:TRIG TABLE1`
Query syntax	[SOURce:]FUNCtion[:SHAPe]:TRIGgered?
Returned parameters	<CRD>
Related commands	FUNC FUNC:MODE VOLT

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? [THD]
Returned parameters	<NR3>
Related commands	FUNC FUNC:MODE

Source Subsystem (List) Commands

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 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 determines 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 non-volatile memory.

NOTE

MODE commands such as VOLTage: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 <n>{, <n>}         Sets the peak current limit list
      :POINts?                  Returns the number of peak current limit list 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
```

[SOURce:]	
LIST	
:SHAPe <shape>{, <shape>}	Sets the waveform shape list
:POINts?	Returns the number of shape list points
:STEP <step>	Specifies how the list responds to triggers (ONCE | AUTO)
:TTLTrg <bool>{, <bool>}	Defines the output marker list
:POINts?	Returns the number of output marker list points
:VOLTage	
[:LEVel] <n>{, <n>}	Sets the voltage list
:POINts?	Returns the number of voltage level points
:SLEW <n>{, <n>}	Sets the voltage slew list
:POINts?	Returns the number of voltage slew points
:OFFSet <n>{, <n>}	Sets the voltage offset list
:POINts?	Returns the number of voltage offset points
:SLEW <n>{, <n>}	Sets the offset voltage slew list
:POINts?	Returns the number of offset voltage slew 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 9.9E37, but any number greater than 2E9 is interpreted as infinity. Use INFinity to execute a list indefinitely.

Command syntax	[SOURce:]LIST:COUNt <NRf+> | INFinity
Parameters	1 to 9.9E37 | MINimum | MAXimum | INFinity
***RST value**	1
Examples	`LIST:COUN 3` `LIST:COUN INF`
Query syntax	[SOURce:]LIST:COUNt?
Returned parameters	<NR3>
Related commands	LIST:CURR LIST:FREQ LIST:TTLT LIST:VOLT

LIST:CURRent

This command sets the sequence of peak output current list points. 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 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:CURRent <NRf+>{, <NRf+>}
Parameters	MINimum | MAXimum | 0 to 3.25 (6811C including all options) 0 to 6.50 (6812C including all options) 0 to 13.0 (6813C all options except Option 019) 0 to 16.0 (6813C Option 019)
Unit	*A* (peak amperes)
***RST value**	1
Examples	`LIST:CURR 2.5, 3.0, 3.5` `LIST:CURR MAX, 3.5, 2.5, MIN`
Query syntax	[SOURce:]LIST:CURRent?
Returned parameters	<NR3>{, <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:CURRent. Note that it returns only the total number of points, not the point values.

Query syntax	[SOURce:]LIST:CURRent:POINts?
Returned parameters	<NR1>
Examples	`LIST:CURR:POIN?`
Related commands	LIST:CURR

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 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	0 to 4.30349E5 | MINimum | MAXimum
Unit	*s* (seconds)
Examples	`LIST:DWEL 2.5, 1.5, .5`
Query syntax	[SOURce:]LIST:DWELl?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:CURR 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>
Examples	`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	MINimum | MAXimum | 0.001 to 1000 (6811C/6812C/6813C except Option 019) 0.001 to 100 (6813C Option 019)
Unit	*Hz* (hertz)
Examples	`LIST:FREQ 55, 60, 65`
Query syntax	[SOURce:]LIST:FREQuency[:LEVel]?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:FREQ:POIN? LIST:COUN LIST:DWEL LIST:STEP

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:FREQuency:POINts?
Returned parameters	<NR1>
Examples	`LIST:FREQ:POIN?`
Related commands	LIST:FREQ

LIST:FREQuency:SLEW

This command specifies the output frequency slew list points. 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 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+> | INF {, <NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*Hz* (hertz per second)
Examples	`LIST:FREQ:SLEW 10, 20, 1E2`
Query syntax	[SOURce:]LIST:FREQuency:SLEW?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:FREQ:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

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:FREQuency:SLEW:POINts?
Returned parameters	<NR1>
Examples	`LIST:FREQ:SLEW:POIN?`
Related commands	LIST:FREQ:SLEW

LIST:PHASe

This 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 (degrees) | MAXimum | MINimum
Examples	`LIST:PHAS 90, 120, 150`
Query syntax	[SOURce:]LIST:PHASe?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:FREQ: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	<NR1>
Examples	`LIST:PHAS:POIN?`
Related commands	LIST:PHAS

LIST:SHAPe

This command sets the sequence of the waveform shape entries. The order in which the shapes are given determines the sequence in which the list of shape will be output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt. The following shapes may be specified:

- **SINusoid**: A sine wave is output
- **SQUare**: A square wave is output
- **CSINusoid**: The output is a clipped sine wave. Both positive and negative peak amplitudes are clipped at a value determined by the FUNCtion:CSINusoid command.
- **<table>**: The output shape is described by one of the user-defined waveform tables.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]LIST[:SHAPe] <shape>{, <shape>}
Parameters	SINusoid | SQUare | CSINusoid | <table>
Examples	`LIST:SHAP SIN`
Query syntax	[SOURce:]LIST:SHAP?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:SHAP:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:VOLT LIST:VOLT:OFFS

LIST:SHAPe:POINts?

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

Query syntax	[SOURce:]LIST:SHAPe:POINts?
Returned parameters	<NR1>
Examples	`LIST:SHAP:POIN?`
Related commands	LIST:SHAP

LIST:STEP

This command specifies how the list sequencing responds to triggers. The following parameters may be specified:

- **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 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	[SOURce:]LIST:TTLTrg?
Returned parameters	0 | 1 {,0 | 1}
Related commands	LIST:TTLT:POIN? LIST:COUN LIST:DWEL LIST:STEP OUTP:TTLT 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>
Examples	`LIST:TTLT:POIN?`
Related commands	LIST:TTLT

LIST:VOLTage

This command specifies the output voltage points in a list. The voltage 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.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]LIST:VOLTage[:LEVel] <NRf+>{, <NRf+>}
Parameters	For sine waves: 0 to 300 | MAXimum | MINimum
Unit	*V* (rms voltage)
Examples	`LIST:VOLT 115, 126, 120` `LIST:VOLT MAX, 120, MIN`
Query syntax	[SOURce:]LIST:VOLTage[:LEVel]?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:VOLT:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:VOLT:SLEW LIST:VOLT:OFFS

LIST:VOLTage:POINts?

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

Query syntax	[SOURce:]LIST:VOLTage[:LEVel]:POINts?
Returned parameters	<NR1>
Examples	`LIST:VOLT:POIN?`
Related commands	LIST:VOLT

LIST:VOLTage:SLEW

This command specifies the output voltage slew list points. 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 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:VOLTage:SLEW <NRf+> | INF {, <NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
Examples	`LIST:VOLT:SLEW 10, 20, 1E2`
Query syntax	[SOURce:]LIST:VOLTage:SLEW?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:VOLT:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:VOLTage:SLEW:POINts?

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

Query syntax	[SOURce:]LIST:VOLTage:SLEW:POINts?
Returned parameters	<NR1>
Examples	`LIST:VOLT:SLEW:POIN?`
Related commands	LIST:VOLT:SLEW

LIST:VOLTage:OFFSet

This command specifies the DC offset points in a list. The offset 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.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]LIST:VOLTage:OFFSet <NRf+>{, <NRf+>}
Parameters	–425 to +425 | MAXimum | MINimum
Unit	*V* (DC voltage)
Examples	`LIST:VOLT:OFFS 50, 75, 100`
Query syntax	[SOURce:]LIST:VOLTage:OFFSet?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:VOLT:OFFS:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:VOLT:SLEW

LIST:VOLTage:OFFSet:POINts?

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

Query syntax	[SOURce:]LIST:VOLTage:OFFSet:POINts?
Returned parameters	<NR1>
Examples	`LIST:VOLT:OFFS:POIN?`
Related commands	LIST:VOLT:OFFS

LIST:VOLTage:OFFSet:SLEW

This command specifies the DC offset slew list points. 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 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:VOLTage:OFFSet:SLEW <NRf+> | INF {, <NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
Examples	`LIST:VOLT:OFFS:SLEW 10, 20, 1E2`
Query syntax	[SOURce:]LIST:VOLTage:OFFSet:SLEW?
Returned parameters	<NR3>{, <NR3>}
Related commands	LIST:VOLT:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:VOLTage:OFFSet:SLEW:POINts?

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

Query syntax	[SOURce:]LIST:VOLTage:OFFSet:SLEW:POINts?
Returned parameters	<NR1>
Examples	`LIST:VOLT:OFFS:SLEW:POIN?`
Related commands	LIST:VOLT:OFFS:SLEW

Source Subsystem (Phase) Commands

This subsystem programs the output phases. 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	
[:ADJust|:IMMediate] <n>	Sets the output phase
:AUXiliary	
[:ADJust|: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

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.

Command syntax	[SOURce:]PHASe[:ADJust|:IMMediate] <NRf+>
Parameters	–360 through +360 (degrees) | MAXimum | MINimum
***RST value**	0
Examples	`PHAS 90` `PHAS MAX`
Query syntax	[SOURce:]PHASe[:ADJust|:IMMediate]?
Returned parameters	<NR3>
Related commands	PHAS:MODE PHAS:TRIG

PHASe:AUXiliary

Keysight 6812C/6813C Option 026 only

This command sets the phase of the auxiliary 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.

Command syntax	[SOURce:]PHASe:AUXiliary[:ADJust|:IMMediate] <NRf+>
Parameters	–360 through +360 (degrees) | MAXimum | MINimum
***RST value**	0
Examples	`PHAS:AUX 90` `PHAS:AUX MAX`
Query syntax	[SOURce:]PHASe:AUXiliary[:ADJust|:IMMediate]?
Returned parameters	<NR3>
Related commands	VOLT:AUX

PHASe:MODE

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

- **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 output phase is controlled by the phase list when a triggered transient occurs.

Command syntax	[SOURce:]PHASe:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`PHAS:MODE LIST` `PHAS:MODE FIX`
Query syntax	[SOURce:]PHASe:MODE?
Returned parameters	<CRD>
Related commands	PHAS PHAS:TRIG

PHASe:TRIGgered

This command sets the output phase when a triggered step or pulse transient occurs. The phase of the output voltage 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.

Command syntax	[SOURce:]PHASe:TRIGgered <NRf+>
Parameters	–360 through +360 (degrees) | MAXimum | MINimum
***RST value**	0
Examples	`PHAS:TRIG 90` `PHAS:TRIG MAX`
Query syntax	[SOURce:]PHASe:TRIGgered?
Returned parameters	<NR3>
Related commands	PHAS:MODE PHAS

Source Subsystem (Pulse) Commands

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" on page 151 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 <parameter>	Selects parameter that is held constant (WIDTh | DCYCle)
: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 9.9E37. If INFinity or MAXimum is sent, the output pulse repeats indefinitely.

Command syntax	[SOURce:]PULSe:COUNt <Nrf+> | INFinity
Parameters	1 to 9.9E37 | MINimum | MAXimum | INFinity
***RST value**	1
Examples	`PULS:COUN 3` `PULS:COUN MIN` `PULS:COUN INF`
Query syntax	[SOURce:]PULSe: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 (percent) | 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

PULSe:HOLD = WIDTh

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

PULSe:HOLD = DCYCle

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

PULSe:PERiod

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

Command syntax	[SOURce:]PULSe:PERiod <NRf+>
Parameters	0 to 4.30349E5 | MINimum | MAXimum
Unit	*s* (seconds)
***RST value**	0.03333
Examples	`PULS:PER 0.001` `PULS:PER MIN`
Query syntax	[SOURce:]PULSe: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 model-dependent.

Command syntax	[SOURce:]PULSe:WIDTh <NRf+>
Parameters	0 to 4.30349E5 | 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:PER PULS:HOLD

Source Subsystem (Voltage) Commands

This subsystem programs the output voltage of the ac source.

Subsystem syntax

[SOURce:]	
VOLTage	
[:LEVel]	
[:IMMediate]	
[:AMPLitude] <n>	Sets the AC rms voltage amplitude of the ac source
:TRIGgered	
[:AMPLitude] <n>	Sets the transient voltage amplitude of the ac source
:AUXiliary	
[:LEVel]	
[:IMMediate]	
[:AMPLitude] <n>	Sets the AC rms voltage amplitude of the auxiliary output
:MODE <mode>	Sets the voltage mode (FIX | STEP | PULS | LIST)
:OFFSet	
[:IMMediate] <n>	Sets the DC offset voltage
:MODE <mode>	Sets the offset mode (FIX | STEP | PULS | LIST)
:TRIGgered <n>	Sets the transient DC offset voltage
:SLEW	
[:IMMediate] <n> | INFinity	Sets the voltage slew rate
:MODE <mode>	Sets voltage slew mode (FIX | STEP | PULS | LIST)
:TRIGgered <n> | INFinity	Sets the transient voltage slew rate
:PROTection	
[:LEVel] <n>	Sets the overvoltage protection threshold
:STATe <bool>	Sets the overvoltage protection state
:SENSe | :ALC	
:DETector <type>	Sets the sense detector (RTIM | RMS)
:SOURce <source>	Sets voltage sense source (INT | EXT)
:SLEW	
[:IMMediate] <n> | INFinity	Sets the voltage slew rate
:MODE <mode>	Sets voltage slew mode (FIX | STEP | PULS | LIST)
:TRIGgered <n> | INFinity	Sets the transient voltage slew rate

VOLTage

This command programs the AC rms output voltage level of the ac source.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>
Parameters	MINimum | MAXimum | For sine waves: 0 to 300 (6811C/6812C/6813C except Option 019) 0 to 308 (6813C Option 019)
Unit	*V* (rms voltage)
***RST value**	0
Examples	`VOLT 115` `VOLT:LEV 250`
Query syntax	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?
Returned parameters	<NR3>
Related commands	VOLT:MODE VOLT:TRIG VOLT:OFFS FUNC:SHAP

VOLTage:TRIGgered

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

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

Command syntax	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf+>
Parameters	For sine waves: 0 to 300 | MAXimum | MINimum
Unit	*V* (rms voltage)
***RST value**	0
Examples	`VOLT:TRIG 120` `VOLT:LEV:TRIG 150`
Query syntax	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
Returned parameters	<NR3> (if the trigger level is not programmed, the immediate level is returned)
Related commands	VOLT VOLT:MODE VOLT:OFFS FUNC:SHAP

VOLTage:AUXiliary

Keysight 6812C/6813C Option 026 only

This command programs the AC rms output voltage level of the auxiliary output. The auxiliary output waveform and frequency are always the same as the main output. The auxiliary output is enabled along with the main output.

The maximum peak voltage of the auxiliary output is 36.8 V_{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 sine wave, the maximum voltage that can be programmed is 26.02 V_{rms}.

Command syntax	[SOURce:]VOLTage:AUXiliary[:LEVel][:IMMediate][:AMPLitude] <NRf+>
Parameters	For sine waves: 0 to 26.02 | MAXimum | MINimum
Unit	*V* (rms voltage)
***RST value**	0
Examples	`VOLT:AUX 12` `VOLT:AUX:LEV 15`
Query syntax	[SOURce:]VOLTage:AUXiliary[:LEVel][:IMMediate][:AMPLitude]?
Returned parameters	<NR3>
Related commands	PHAS:AUX

VOLTage:MODE

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

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

Command syntax	[SOURce:]VOLTage:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`VOLT:MODE FIX` `VOLT:MODE:LIST`
Query syntax	[SOURce:]VOLTage:MODE?
Returned parameters	<CRD>
Related commands	VOLT VOLT:TRIG

VOLTage:OFFSet

This command programs the DC output voltage level of the ac source.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

NOTE The OUTPut:COUPling must be set to DC to get non-zero DC output.

Command syntax	[SOURce:]VOLTage:OFFSet[:IMMediate] <NRf+>
Parameters	−425 to +425 | MAXimum | MINimum
Unit	*V* (DC voltage)
***RST value**	0
Examples	`VOLT:OFFS 100`
Query syntax	[SOURce:]VOLTage:OFFSet[:IMMediate]?
Returned parameters	<NR3>
Related commands	VOLT:OFFS:MODE OUTP:COUP FUNC:SHAP

VOLTage:OFFSet:MODE

This command determines how the DC offset voltage is controlled during a triggered output transient. The choices are as follows:

- **FIXed**: The offset is unaffected by a triggered output transient.
- **STEP**: The offset is programmed to the value set by VOLTage:OFFSet:TRIGgered when a triggered transient occurs.
- **PULSe**: The offset is changed to the value set by VOLTage:OFFSet:TRIGgered for a duration determined by the pulse commands.
- **LIST**: The offset is controlled by the voltage list when a triggered transient occurs.

Command syntax	[SOURce:]VOLTage:OFFSet:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`VOLT:OFFS:MODE FIX` `VOLT:OFFS:MODE:LIST`
Query syntax	[SOURce:]VOLTage:OFFSet:MODE?
Returned parameters	<CRD>
Related commands	VOLT:OFFS VOLT:OFFS:TRIG

VOLTage:OFFSet:TRIGgered

This command selects the DC offset that the output waveform will be set to during a triggered step or pulse transient.

The maximum peak voltage that the ac source can output is 425 V_{peak}. This includes any combination of voltage, voltage offset, 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 sine wave, the maximum voltage that can be programmed is 300 V_{rms}.

NOTE **The OUTPut:COUPling must be set to DC to get non-zero DC output.**

Command syntax	[SOURce:]VOLTage:OFFSet:TRIGgered <NRf+>
Parameters	–425 to +425 | MAXimum | MINimum
Unit	*V* (DC voltage)
***RST value**	0
Examples	`VOLT:OFFS:TRIG 50` `VOLT:OFFS:TRIG INF`
Query syntax	[SOURce:]VOLTage:OFFSet:TRIGgered?
Returned parameters	<NR3>
Related commands	VOLT:OFFS:MODE OUTP:COUP

VOLTage:OFFSet:SLEW

This command sets the slew rate for all programmed changes in DC output voltage. A parameter of MAXimum or INFinity sets the slew to its maximum possible rate. The SCPI representation for INFinity is 9.9E37.

Command syntax	[SOURce:]VOLTage:OFFSet:SLEW[:IMMediate] <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
***RST value**	INFinity
Examples	`VOLT:OFFS:SLEW 50` `VOLT:OFFS:SLEW MAX`
Query syntax	[SOURce:]VOLTage:OFFSet:SLEW[:IMMediate]?
Returned parameters	<NR3>
Related commands	VOLT:OFFS:MODE OUTP:COUP

VOLTage:OFFSet:SLEW:MODE

This command determines how the DC offset slew rate is controlled during a triggered output transient. The choices are as follows:

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

Command syntax	[SOURce:]VOLTage:OFFSet:SLEW:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`VOLT:OFFS:SLEW:MODE STEP`
Query syntax	[SOURce:]VOLTage:OFFSet:SLEW:MODE?
Returned parameters	<CRD>
Related commands	VOLT:OFFS:SLEW VOLT:OFFS:SLEW:TRIG

VOLTage:OFFSet:SLEW:TRIGgered

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

Command syntax	[SOURce:]VOLTage:OFFSet:SLEW:TRIGgered <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
***RST value**	INFinity
Examples	`VOLT:OFFS:SLEW:TRIG 50` `VOLT:OFFS:SLEW:TRIG MAX`
Query syntax	[SOURce:]VOLTage:OFFSet:SLEW:TRIGgered?
Returned parameters	<NR3>
Related commands	VOLT:OFFS:SLEW VOLT:OFFS:SLEW:MODE

VOLTage:PROTection

This command sets the overvoltage protection (OVP) level of the ac source. If the peak output voltage exceeds the OVP level, then the output is disabled and the Questionable Condition status register OV bit is set (refer to “Programming the Status Registers” on page 259 in Chapter 5, "Programming Examples"). An overvoltage condition can be cleared with the OUTPut:PROTection:CLEar command after the condition that caused the OVP trip is removed. The OVP always trips with zero delay and is unaffected by the OUTPut:PROTection:DELay command.

Command syntax	[SOURce:]VOLTage:PROTection[:LEVel] <NRf+>
Parameters	0 to 500 | MAXimum | MINimum
Unit	*V* (peak voltage)
***RST value**	MAXimum
Examples	`VOLT:PROT 400` `VOLT:PROT:LEV MAX`
Query syntax	[SOURce:]VOLTage:PROTection[:LEVel]?
Returned parameters	<NR3>
Related commands	OUTP:PROT:CLE OUTP:PROT:DEL

VOLTage:PROTection:STATe

This command enables or disables the overvoltage protection feature.

Command syntax	[SOURce:]VOLTage:PROTection:STATe <bool>
Parameters	0 | 1 | OFF | ON
***RST value**	OFF
Examples	`VOLT:PROT:STAT 1` `VOLT:PROT:STAT ON`
Query syntax	[SOURce:]VOLTage:PROTection:STATe?
Returned parameters	<NR3>
Related commands	VOLT:PROT

VOLTage:SENSe:DETector

VOLTage:ALC:DETector

These commands select the type of closed-loop feedback (voltage control loop) that is used by the output power circuits of the ac source. The commands are interchangeable; they both perform the same function. The following closed-loop feedbacks can be selected:

- **RTIMe**: This feeds the instantaneous output voltage back to the error amplifier and compares it to the reference waveform.
- **RMS**: This converts the rms output voltage to DC and compares it to a DC reference.

Command syntax	[SOURce:]VOLTage:SENSe:DETector <type> [SOURce:]VOLTage:ALC:DETector <type>
Parameters	RTIMe | RMS
***RST value**	RTIMe
Examples	`VOLT:SENS:DET RTIM` `VOLT:ALC:DET RMS`
Query syntax	[SOURce:]VOLTage:SENSe:DETector? [SOURce:]VOLTage:ALC:DETector?
Returned parameters	<CRD>
Related commands	VOLT:SENS:SOUR

VOLTage:SENSe:SOURce

VOLTage:ALC:SOURce

These commands select the source from which the output voltage is sensed. The commands are interchangeable; they both perform the same function. 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 rear panel voltage 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` `VOLT:ALC:SOUR EXT`
Query syntax	[SOURce:]VOLTage:SENSe:SOURce? [SOURce:]VOLTage:ALC:SOURce?
Returned parameters	<CRD>
Related commands	VOLT:SENS:DET

VOLTage:SLEW

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

Command syntax	[SOURce:]VOLTage:SLEW[:IMMediate] <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
***RST value**	INFinity
Examples	`VOLT:SLEW 50` `VOLT:SLEW INF`
Query syntax	[SOURce:]VOLTage:SLEW[:IMMediate]?
Returned parameters	<NR3>
Related commands	VOLT:SLEW:MODE VOLT:SLEW:TRIG

VOLTage:SLEW:MODE

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

– **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 VOLTage:SLEW:TRIGgered for a duration determined by the pulse commands.

– **LIST**: The slew rate is controlled by the voltage list when a triggered transient occurs.

Command syntax	[SOURce:]VOLTage:SLEW:MODE <mode>
Parameters	FIXed | STEP | PULSe | LIST
***RST value**	FIXed
Examples	`VOLT:SLEW:MODE STEP`
Query syntax	[SOURce:]VOLTage:SLEW:MODE?
Returned parameters	<CRD>
Related commands	VOLT:SLEW VOLT:SLEW:TRIG

VOLTage:SLEW:TRIGgered

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

Command syntax	[SOURce:]VOLTage:SLEW:TRIGgered <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
***RST value**	INFinity
Examples	`VOLT:SLEW:TRIG 50` `VOLT:SLEW:TRIG MAX`
Query syntax	[SOURce:]VOLTage:SLEW:TRIGgered?
Returned parameters	<NR3>
Related commands	VOLT:SLEW VOLT:SLEW:MODE

Status Subsystem Commands

This subsystem programs the ac source status registers. The ac source has three groups of status registers; Operation, Questionable, and Standard Event. The Standard Event group is programmed with Common commands. The Operation and Questionable status groups each consist of the following five registers:

– Condition

– Enable

– Event

– NTR Filter

– PTR Filter

Refer to “Programming the Status Registers” on page 259 in Chapter 5, "Programming Examples" for more information.

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

STATus:PRESet

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

– **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 configuration of Operation Status registers

Bit position	15 – 9	8	7 – 6	5	4 – 1	0
Bit name	n.u.	CV	n.u.	WTG	n.u.	CAL
Bit weight		256		32		1

n.u. = Not used

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

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		
Examples	`STAT:OPER:EVEN?`		
Returned parameters	<NR1> (register value)		
Related commands	*CLS	STAT:OPER:NTR	STAT:OPER:PTR

STATus:OPERation:CONDition?

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

Query syntax	STATus:OPERation:CONDition?
Parameters	None
Examples	`STAT:OPER:COND?`
Returned parameters	<NR1> (register value)
Related commands	STAT:QUES:COND?

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 32767 | MAXimum | MINimum
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?

STATus:OPERation:NTRansition

STATus:OPERation:PTRansition

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 32767 | MAXimum | MINimum
Default value	0
Examples	`STAT:OPER:NTR 32` `STAT:OPER:PTR 1`
Query syntax	STATus:OPERation:NTRansition? STATus:OPERation:PTRansition?
Returned parameters	<NR1> (register value)
Related commands	STAT:OPER:ENAB

Bit configuration of Questionable Status registers

Bit position	15	14	13	12	11	10	9	8 – 5	4	3	2	1	0
Bit name	n.u.	Meas Ovld	n.u.	CL rms	Rail	CL peak	RI	n.u.	OT	UNR	SOA	OCP	OV
Bit weight		16384		4096	2048	1024	512		16	8	4	2	1

n.u. = Not used
OV = Overvoltage protection has tripped
OCP = Overcurrent protection has tripped
SOA = Safe operating area protection has tripped
UNR = Output is unregulated
OT = Overtemperature protection has tripped
RI = Remote inhibit is active
CL peak = Peak current limit is active
Rail = Rail protection tripped
CL rms = rms current limit is active
MeasOvld = Current measurement exceeded low current range capability

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.

Query syntax	STATus:QUEStionable[:EVENt]?
Parameters	None
Examples	`STAT:QUES:EVEN?`
Returned parameters	<NR1> (register value)
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?
Parameters	None
Examples	`STAT:QUES:COND?`
Returned parameters	<NR1> (register value)
Related commands	STAT:OPER:COND?

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 32767 | MAXimum | MINimum
Default value	0
Examples	`STAT:QUES:ENAB 32` `STAT:QUES:ENAB 1`
Query syntax	STATus:QUEStionable:ENABle?
Returned parameters	<NR1> (register value)
Related commands	STAT:QUES?

STATus:QUEStionable:NTRansition

STATus:QUEStionable:PTRansition

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

– When a bit in the Questionable NTR register is set to 1, then **a 1-to-0 transition** of the corresponding bit in 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 32767 | MAXimum | MINimum
Default value	0
Examples	`STAT:QUES:NTR 32` `STAT:QUES:PTR 1`
Query syntax	STATus:QUEStionable:NTRansition? STATus:QUEStionable:PTRansition?
Returned parameters	<NR1> (register value)
Related commands	STAT:QUES:ENAB

Bit configuration of Questionable Instrument Summary registers

Bit position	15 – 13	12	11	10	9	8 – 5	4	3	2	1	0
Bit name	n.u.	CL rms	Rail	n.u.	RI	n.u.	OT	UNR	n.u.	OCP	OV
Bit weight		4096	2048		512		16	8		2	1

n.u. = Not used
OV = Overvoltage protection has tripped
OCP = Overcurrent protection has tripped
UNR = Output is unregulated
OT = Overtemperature protection has tripped
RI = Remote inhibit is active
Rail = Rail protection tripped
CL rms = rms current limit is active

System Subsystem Commands

The system commands control the system-level functions of the ac source.

Subsystem syntax

```
SECurity
    [:IMMediate]
SYSTem
    :COMMunicate
        :LAN
            :CONTrol?                     Reads the initial control connection port number
            :DHCP                         Disables or enables instrument's use of DHCP
            :DNS[{1|2}]                   Assigns static IP addresses
            :DOMain?                      Returns the domain name assigned to the ac source
            :GATeway <quoted_address>     Assigns a default gateway for the ac source
            :HOSTname <quoted_string>     Assigns a hostname to the ac source
            :IPADdress <quoted_address>   Assigns a static Internet Protocol (IP) address
            :MAC?                         Returns the Media Access Control (MAC) address
            :SMASk <quoted_address>       Assigns a subnet mask for the ac source
            :TELNet
                :PROMpt<quoted_string>    Sets the command prompt
                :WMESsage <quoted_string> Sets the welcome message
            :UPDate                       Stores changes made to the LAN settings
        :RLSTate <mode>                   Sets the state of the interface
        :TCPip
            :CONTrol?                     Sets the overall operating mode of the ac source
    :ERRor?                               Returns the error number and error string
    :LANGuage <language>                  Sets the command language of the ac source to SCPI
    :PERSona
        :MANufacturer <quoted_string>     Sets the manufacturer string
            :DEFault                      Sets the manufacturer string to Keysight
                                          Technologies
        :MODel <quoted_string>            Sets the model string returned by *IDN? Query
            :DEFault                      Sets the model string to default
    :SECurity
        :COUNt?                           Returns the security count of the ac source
        [:IMMediate]                      Clears all user-accessible instrument memory
    :SET <binary_block>                   Sets up the ac source
```

SYSTem

:VERSion?	Returns the SCPI version number
:LOCal	Go to local mode
:REMote	Go to remote mode
:RWLock	Go to remote with lockout mode

SECurity[:IMMediate]

Sanitizes all user-accessible instrument memory except for the calibration constants and reboots the instrument. This feature clears all user-defined state information, measurement data, and user defined I/O settings such as the IP address. This feature is not recommended for use in routine applications because of the possibility of unintended data loss. The command is typically used before removing an instrument from a secure area. Increments the security count.

Command syntax	SECurity[:IMMediate]
Parameters	None
Examples	`SEC` `SEC:IMM`
Related commands	SYST:SEC:IMM

SYSTem:COMMunicate:LAN:CONTrol?

Reads the initial control connection port number for the communication of sockets. This connection is used to send and receive commands and queries. Use the control-socket connection to send a Device Clear (DCL) to the instrument or to detect pending SRQ events.

Query syntax	SYSTem:COMMunicate:LAN:CONTrol?
Parameters	None
Examples	`SYST:COMM:LAN:CONT?`
Returned parameters	<NR1>
Related commands	SYST:COMM:TCP:CONT

SYSTem:COMMunicate:LAN:DHCP

Disables or enables instrument's use of DHCP. The acronym DHCP stands for Dynamic Host Configuration Protocol, a protocol for assigning dynamic IP addresses to networked devices. With dynamic addressing, a device can have a different IP address every time it connects to the network.

- **ON**: Instrument tries to obtain an IP address from a DHCP server. If a DHCP server is found, it assigns a dynamic IP address, Subnet Mask, and Default Gateway to the instrument.
- **OFF or DHCP unavailable**: Instrument uses the static IP address, Subnet Mask, and Default Gateway during power-on.

Most corporate LANs have a DHCP server. The SYSTem:SECurity:IMMediate sets this parameter to its default value.This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:DHCP <bool>
Parameters	0 | 1 | OFF | ON
Examples	`SYST:COMM:LAN:DHCP 1`
Query syntax	SYSTem:COMMunicate:LAN:DHCP?
Returned parameters	0 | 1

SYSTem:COMMunicate:LAN:DNS[{1|2}]

Assigns static IP addresses of Domain Name System (DNS) servers. A primary and a secondary server address may be assigned. If DHCP is available and enabled, DHCP will auto-assign these server addresses. These auto-assigned server addresses take precedence over the static addresses assigned with this command. Contact your LAN administrator for details.

– **CURRent**: Returns address currently being used by the instrument.

– **STATic**: Returns static address from non-volatile memory. This address is used if DHCP is disabled or unavailable.

– SYSTem:SECurity:IMMediate sets this parameter to its default value.

The assigned DNS server addresses are used if DHCP is disabled or unavailable. Otherwise, the DNS server addresses are auto-assigned by DHCP. The DNS server addresses are stored in non-volatile memory. They do not change when power has been off, *RST, or SYSTem:PRESet.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:DNS <quoted_address>
Parameters	<quoted_address>
Examples	`SYST:COMM:LAN:DNS “192.168.10.1”`
	`SYST:COMM:LAN:DNS2 “141.183.10.1”`
Query syntax	SYSTem:COMMunicate:LAN:DNS[1|2]? [mode]
Parameters	[CURRent | STATic] (Default: CURRent)
Examples	`SYST:COMM:LAN:DNS1?`
	`SYST:COMM:LAN:DNS2? STAT`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:DOMain?

Returns the domain name assigned to the ac source. If Dynamic Domain Name System (DNS) is available on your network and your instrument uses DHCP, the domain name is assigned by the Dynamic DNS service at power-on. A null string (“”) indicates that no domain name is assigned.

Query syntax	SYSTem:COMMunicate:LAN:DOMain?
Parameters	None
Examples	`SYST:COMM:LAN:DOM?`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:GATeway

Assigns a default gateway for the ac source. The specified IP Address sets the default gateway, which allows the instrument to communicate with systems that are not on the local subnet. Thus, this is the default gateway where packets are sent that are destined for a device not on the local subnet, as determined by the Subnet Mask setting. Contact your LAN administrator for details.

– **CURRent**: returns address currently being used by the instrument.
– **STATic**: returns static address from non-volatile memory. This address is used if DHCP is disabled or unavailable.
– SYSTem:SECurity:IMMediate sets this parameter to its default value.

If DHCP is enabled (SYSTem:COMMunicate:LAN:DHCP ON), the specified default gateway is not used. However, if the DHCP server fails to assign a valid IP address, the currently configured default gateway is used. This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:GATeway <quoted_address>
Parameters	<quoted_address>
Examples	`SYST:COMM:LAN:GAT "192.168.10.1"`
Query syntax	SYSTem:COMMunicate:LAN:GATeway? [mode]
Parameters	[CURRent | STATic] (Default: CURRent)
Examples	`SYSTem:COMMunicate:LAN:GAT?`
	`SYSTem:COMMunicate:LAN:GAT? STAT`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:HOSTname

Assigns a hostname to the ac source. A hostname is the host portion of the domain name, which is translated into an IP address. If Dynamic Domain Name System (Dynamic DNS) is available on your network and your instrument uses DHCP, the hostname is registered with the Dynamic DNS service at power-on. If DHCP is enabled (SYSTem:COMMunicate:LAN:DHCP ON), the DHCP server can change the specified hostname.

If no hostname exists, a null string (“”) is returned. For the query form, specify “CURRent” (default) to read the value currently being used by the instrument. Specify “STATic” to read the host name currently stored in nonvolatile memory within the instrument (may not be the actual name used by the instrument if DHCP is enabled). This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet. This parameter is set to its default value when the instrument is shipped from the factory and after SYSTem:SECurity:IMMediate.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:HOSTname <quoted_string>
Parameters	<quoted_string>
Examples	`SYST:COMM:LAN:HOST “LAB-ACS”`
Query syntax	SYSTem:COMMunicate:LAN:HOSTname? [mode]
Parameters	[CURRent | STATic] (Default: CURRent)
Examples	`SYSTem:COMMunicate:LAN:HOST?`
	`SYSTem:COMMunicate:LAN:HOST? STAT`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:IPADdress

Assigns a static Internet Protocol (IP) address for the ac source. If DHCP is enabled (SYSTem:COMMunicate:LAN:DHCP ON), the specified static IP address is not used. Contact your LAN administrator for details.

SYSTem:SECurity:IMMediate sets this parameter to its default value. For the query form, specify “CURRent” (default) to read the value currently being used by the instrument. Specify “STATic” to read the value currently stored in nonvolatile memory within the instrument (may not be the actual address used by the instrument if DHCP is enabled). This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet. This parameter is set to its default value when the instrument is shipped from the factory and after SYSTem:SECurity:IMMediate.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:IPADdress <quoted_address>
Parameters	<quoted_address>
Examples	`SYST:COMM:LAN:IPAD “192.168.10.1”`
Query syntax	SYSTem:COMMunicate:LAN:IPADdress? [mode]
Parameters	[CURRent | STATic] (Default: CURRent)
Examples	`SYSTem:COMMunicate:LAN:IPAD?` `SYSTem:COMMunicate:LAN:IPAD? STAT`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:MAC?

Returns the ac source's Media Access Control (MAC) address as an ASCII string of 12 hexadecimal characters (0-9 and A-F) enclosed in quotation marks. The MAC address is also known as the link-layer address, the Ethernet (station) address, LANIC ID or Hardware Address. This is an unchangeable 48-bit address assigned by the manufacturer to each unique Internet device.

Your LAN administrator may need the MAC address to assign a static IP address for this device.

Query syntax	SYSTem:COMMunicate:LAN:MAC?
Parameters	None
Examples	`SYST:COMM:LAN:MAC?`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:SMASk

Assigns a subnet mask for the ac source to use in determining whether a client IP address is on the same local subnet. When a client IP address is on a different subnet, all packets must be sent to the Default Gateway. Contact your LAN administrator for details.

If DHCP is enabled (SYSTem:COMMunicate:LAN:DHCP ON), the specified subnet mask is not used. However, if the DHCP server fails to assign a valid IP address, the instrument uses the Auto-IP subnet mask. A value of “0.0.0.0” or “255.255.255.255” indicates that subnetting is not being used.

- **CURRent**: Returns address currently being used by the instrument.
- **STATic**: Returns static address from non-volatile memory. This address is used if DHCP is disabled or unavailable.
- SYSTem:SECurity:IMMediate sets this parameter to its default value.

This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet.

If you change this setting, you must send SYSTem:COMMunicate:LAN:UPDate to activate the new setting.

Command syntax	SYSTem:COMMunicate:LAN:SMASk <quoted_address>
Parameters	<quoted_address>
Examples	`SYST:COMM:LAN:SMAS “12.168.10.1”`
Query syntax	SYSTem:COMMunicate:LAN:SMASk?
Parameters	[CURRent | STATic] (Default: CURRent)
Examples	`SYSTem:COMMunicate:LAN:SMAS?`
Returned parameters	<SRD>

SYSTem:COMMunicate:LAN:TELNet:PROMpt

Sets the command prompt seen when communicating with the instrument via Telnet. The instrument uses LAN port 5024 for SCPI Telnet sessions and port 5025 for SCPI Socket sessions. Telnet sessions are typically started from a host computer shell: telnet <IP_address> <port>

For example:

telnet 169.254.4.10 5024
To exit a Telnet session, press <Ctrl-D>.

This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet.

SYSTem:SECurity:IMMediate sets this parameter to its default value.

Command syntax	SYSTem:COMMunicate:LAN:TELNet:PROMpt <quoted_string>
Parameters	<quoted_string>
Examples	`SYST:COMM:LAN:TELN:PROM "6811C >"`
Query syntax	SYSTem:COMMunicate:LAN:TELNet:PROMpt?
Returned parameters	<SRD>
Related commands	SYST:COMM:LAN:TELN:WMES

SYSTem:COMMunicate:LAN:TELNet:WMESsage

Sets the welcome message seen when communicating with the instrument via Telnet. The instrument uses LAN port 5024 for SCPI Telnet sessions and port 5025 for SCPI Socket sessions. This setting is non-volatile; it is not changed by power cycling, *RST, or SYSTem:PRESet.

SYSTem:SECurity:IMMediate sets this parameter to its default value.

Command syntax	SYSTem:COMMunicate:LAN:TELNet:WMESsage <quoted_string>
Parameters	<quoted_string>
Examples	`SYST:COMM:LAN:TELN:WMES "Welcome to Telnet"`
Query syntax	SYSTem:COMMunicate:LAN:TELNet:WMESsage?
Returned parameters	<SRD>
Related commands	SYST:COMM:LAN:TELN:PROM

SYSTem:COMMunicate:LAN:UPDate

Stores any changes made to the LAN settings into non-volatile memory and restarts the LAN driver with the updated settings. The command must be sent after changing the settings for DHCP, DNS, gateway, hostname, IP address, subnet mask. Make all changes to the LAN settings before sending this command.

The following example configures the instrument to use statically assigned LAN settings.

SYST:COMM:LAN:DHCP OFF
SYST:COMM:LAN:DNS "198.105.232.4"
SYST:COMM:LAN:DNS2 "198.105.232.5"
SYST:COMM:LAN:GAT "198.105.232.1"
SYST:COMM:LAN:HOST "LAB1-DMM"
SYST:COMM:LAN:IPAD "198.105.232.101"
SYST:COMM:LAN:SMAS "255.255.255.0"
SYST:COMM:LAN:UPD

The following example configures the instrument back to use DHCP.

SYST:COMM:LAN:DHCP ON

SYST:COMM:LAN:UPD

Command syntax	SYSTem:COMMunicate:LAN:UPDate
Parameters	None
Examples	`SYST:COMM:LAN:UPD`

SYSTem:COMMunicate:RLSTate

Sets the interface to the following states:

- **Local state**: Enables the front panel controls.
- **Remote state**: 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
- **Remote-lockout state**: Disables all front panel controls including the Local key. Use SYSTem:LOCal or SYSTem:COMMunicate:RLSTate LOCal to return the front panel to the Local state.

Command syntax	SYSTem:COMMunicate:RLSTate <mode>
Parameters	LOCal | REMote | RWLock
Examples	`SYST:COMM:RLST LOC`
Query syntax	SYSTem:COMMunicate:RLSTate?
Returned parameters	<CRD>
Related commands	SYST:LOC SYST:REM SYST:RWL

SYSTem:COMMunicate:TCPip:CONTrol?

Reads the initial control connection port number for sockets communications. This connection is used to send and receive commands and queries. Use the control-socket connection to send a Device Clear (DCL) to the ac source or to detect pending SRQ events.

NOTE Refer to the *Keysight Models 6811C, 6812C, and 6813C User's and Service Guide* under *"Using Sockets"* for more information

Query syntax	SYSTem:COMMunicate:TCPip:CONTrol?
Parameters	None
Examples	`SYST:COMM:TCP:CONT`
Returned parameters	<NR1>
Related commands	SYST:COMM:LAN:CONT

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
Examples	`SYST:ERR?`
Returned parameters	<NR1>, <SRD>

SYSTem:LANGuage

Sets the command language of the ac source to SCPI. 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
Examples	`SYST:LANG SCPI`
Query syntax	SYSTem:LANGuage?
Returned parameters	<CRD>

SYSTem:PERSona:MANufacturer

Sets the manufacturer string returned by *IDN? query.

Command syntax	SYSTem:PERSona:MANufacturer <quoted_string>
Parameters	<quoted_string>
Examples	`SYST:PERS:MAN “Keysight Technologies”`
Query syntax	SYSTem:PERSona:MANufacturer?
Returned parameters	<SRD>
Related commands	SYST:PERS:MAN:DEF

SYSTem:PERSona:MANufacturer:DEFault

Sets the manufacturer string to its default - Keysight Technologies.

Command syntax	SYSTem:PERSona:MANufacturer:DEFault
Parameters	None
Examples	`SYST:PERS:MAN:DEF`
Query syntax	SYSTem:PERSona:MANufacturer:DEFault?
Returned parameters	<SRD>
Related commands	SYST:PERS:MAN

SYSTem:PERSona:MODel

Sets the model string returned by *IDN? query. The models are 6811C, 6812C, or 6813C.

Command syntax	SYSTem:PERSona:MODel <quoted_string>
Parameters	<quoted_string>
Examples	`SYST:PERS:MOD “6811C”`
Query syntax	SYSTem:PERSona:MODel?
Returned parameters	<SRD>
Related commands	SYST:PERS:MOD:DEF

SYSTem:PERSona:MODel:DEFault

Sets the model string to default. This command depends on the ac source model 6811C, 6812C, or 6813C.

Command syntax	SYSTem:PERSona:MODel:DEFault
Parameters	None
Examples	`SYST:PERS:MOD:DEF`
Query syntax	SYSTem:PERSona:MODel:DEFault?
Returned parameters	<SRD>
Related commands	SYST:PERS:MOD

SYSTem:SECurity:COUNt?

Returns the security count of the ac source. The security count is incremented by one every time you perform an action, other than calibration, that requires the ac source to be unsecured. Be sure to read the security count upon receiving your instrument.

Query syntax	SYSTem:SECurity:COUNt?
Parameters	None
Examples	`SYST:SEC:COUN?`
Returned parameters	<NR1>
Related commands	CAL:COUN

SYSTem:SECurity

Sanitizes all user-accessible instrument memory except for the calibration constants and reboots the instrument. This feature clears all user-defined state information, measurement data, and user defined I/O settings such as the IP address. This feature is not recommended for use in routine applications because of the possibility of unintended data loss. The command is typically used before removing an instrument from a secure area. The security count is incremented by one each time you perform this command.

Command syntax	SYSTem:SECurity[:IMMediate]
Parameters	None
Examples	`SYST:SEC`
Related commands	SEC

SYSTem:SET

Sets up the ac source as defined by the data in the binary block of data from the computer. A binary block of data, consisting of bytes of setup information. The numbers of bytes is a dynamic numbers that is read and allocated by the instrument's firmware.SYSTem:SET query operates the same as the *LRN? query.

Command syntax	SYSTem:SET <binary_block>
Parameters	<binary_block>
Examples	`SYST:SET #516384…`
Query syntax	SYSTem:SET?
Returned parameters	<binary_block>
Related commands	*LRN

SYSTem:VERSion?

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

Query syntax	SYSTem:VERSion?
Parameters	None
Examples	`SYST:VERS?`
Returned parameters	<NR2>

SYSTem:LOCal

It sets the interface in Local state, which enables the front panel controls.

Command syntax	SYSTem:LOCal
Parameters	None
Examples	`SYST:LOC`
Related commands	SYST:REM SYST:RWL

SYSTem:REMote

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.

Command syntax	SYSTem:REMote
Parameters	None
Examples	`SYST:REM`
Related commands	SYST:LOC SYST:RWL

SYSTem:RWLock

It sets the interface in the Remote-lockout state, which disables all front panel controls including the **Local** key. Use SYSTem:LOCal to return the front panel to the Local state.

Command syntax	SYSTemRWLock
Parameters	None
Examples	`SYST:RWL`
Related commands	SYST:REM SYST:LOC

Trace Subsystem Commands

This subsystem programs the output waveform of the ac source. 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

TRACe:CATalog?

DATA:CATalog?

These queries 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?
Parameters	None
Examples	`TRAC:CAT?` `DATA:CAT?`
Returned parameters	<SRD>
Related commands	TRAC TRAC:DEL FUNC:SHAP

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.

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 voltage 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 non-volatile memory and is retained when input power is removed. Up to 12 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>
Examples	`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>{, <NR3>} (a total of 1024 data points)
Related commands	TRAC:DEF 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.

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>
Examples	`TRAC:DEF flattop`
Related commands	TRAC 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>
Examples	`TRAC:DEL flattop`
Related commands	TRAC TRAC:DEF FUNC:SHAP

Trigger Subsystem Commands

This subsystem controls the triggering of the ac source. Refer to “Triggering Output Changes” on page 244 in Chapter 5, "Programming Examples" for an explanation of the Trigger Subsystem. The INITiate commands control the initialization of both the transient and measurement trigger systems.

For Keysight 6811C/6812C/6813C Option 020 only, TRIGger:ACQuire and other measurement trigger commands apply to both measurement sources, regardless of which source is selected.

NOTE **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 <name>	Initiates a specific named sequence (TRANsient | ACQuire)
: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 <source>	Sets the synchronous source (PHAse | IMMediate)
: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

TRIGger	
: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 (refer to "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples"). ABORt is executed at power turn-on and upon execution of *RCL, RST, or any implied abort command (refer to "Source Subsystem (List) Commands" on page 134).

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 number	Description
1 (the 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 as follows:

- **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
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	0 to 4.30349E5 | 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**: GPIB device, *TRG, or <GET> (Group Execute Trigger)
- **EXTernal**: AC source's backplane Trigger In BNC
- **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**	IMMediate
Examples	`TRIG:SYNC:SOUR IMM` `TRIG:SEQ2:SOUR PHAS`
Query syntax	TRIGger:SEQuence2:SOURce? TRIGger:SYNChronize:SOURce?
Returned parameters	<CRD>
Related commands	ABOR TRIG TRIG:DEL TRIG:SOUR 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 when the TRIGger:SYNChronize:SOURce is set to PHASe. The range is from −360 to +360 degrees.

Command syntax	TRIGger:SEQuence2:PHASe <NRf+> TRIGger:SYNChronize:PHASe <NRf+>
Parameters	−360 to +360 (degrees) | MAXimum | MINimum
***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 TRIG:DEL TRIG:SYNC 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 voltage and current 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 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**: GPIB device, *TRG, or <GET> (Group Execute Trigger)
- **EXTernal**: AC source's backplane Trigger In BNC
- **TTLTrg**: The signal driving the Trigger Out BNC

Command syntax	TRIGger:SEQuence3:SOURce <CRD> TRIGger:ACQuire:SOURce <CRD>
Parameters	BUS | EXTernal | TTLTrg
***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 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, SYNChronize 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 SYNChronize TRIGger:SEQuence3:DEFine ACQuire
Parameters	TRANsient, SYNChronize, ACQuire
Examples	`TRIG:SEQ1:DEF TRAN` `TRIG:SEQ3:DEF ACQ`
Query syntax	TRIGger:SEQuence1:DEFine? TRIGger:SEQuence2:DEFine? TRIGger:SEQuence3:DEFine?
Returned parameters	<CRD>

Common Commands

Common commands begin with an * and consist of three letters (command) which is the IEEE 488.2 standard to perform some common interface functions. The Keysight ac sources respond to the required common commands that control status reporting, synchronization, and internal operations. The ac sources also respond 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 "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples", 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 enable
*ESR?	Return event status register
*IDN?	Return instrument identification
*LRN?	Return a binary string of all of the saved EEPROM values
*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 self test, then return result
*WAI	Hold off bus until all device commands done

*CLS

This command clears the following registers (refer to "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples" 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 (refer to "*ESR?" on page 208) 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. refer to "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples" for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

Bit configuration of Standard Event Status Enable register

Bit position	7	6	5	4	3	2	1	0
Bit name	PON	n.u.	CME	EXE	DDE	QYE	n.u.	OPC
Bit weight	128		32	16	8	4		1

n.u. = Not used
PON = Power-on
CME = Command error
EXE = Execution error
DDE = Device-dependent error
QYE = Query error
OPC = Operation complete

Command syntax	*ESE <NRf>
Parameters	0 to 255
Power-on value	Refer to *PSC
Examples	`*ESE 129`
Query syntax	*ESE?
Returned parameters	<NR1>
Related commands	*ESR? *PSC *STB?

*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 (refer to “*ESE” on page 207). Refer to “Programming the Status Registers” on page 259 in Chapter 5, "Programming Examples" for a detailed explanation of this register.

Query syntax	*ESR?
Parameters	None
Returned parameters	<NR1> (register value)
Related commands	*CLS *ESE *ESE? *OPC

*IDN?

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

Query syntax *IDN?

Parameters None

Returned parameters <AARD>

Field	Information
Keysight Technologies	Manufacturer
xxxxC	Model number
MYxxxxxxxx	Serial number or 0
x.x.x-x.x.x-x.x.x	Firmware revision

Examples `Keysight Technologies,6813C,MY00000000,1.0.0-1.0.0-1.0.0`

*LRN?

This query returns a binary string of all of the saved EEPROM values required to set the instrument into its current state. It is not compatible with the Telnet interface. The returned string includes “SYSTEM:SET” command header that can be used directly provided the receiver can accept binary data.

Query syntax *LRN?

Parameters None

Returned parameters <binary_block>

Related commands SYST:SET

*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. (refer to "*ESE" on page 207 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 ac 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

*OPT?

This query requests the ac source to identify any options that are installed. Options are identified by a number.

This query returns 0 if no options are installed.

For Keysight 6811C/6812C/6813C Option 020 only, this query returns 20.

For Keysight 6812C/6813C Option 026 only, this query returns 26.

Query syntax	*OPT?
Returned parameters	<AARD>

*PSC

This command controls the automatic clearing at power-on of the Service Request Enable and the Standard Event Status enable registers as follows (refer to "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples" for the register details):

- **1** or **ON**: Prevents the register contents from being saved, causing them to be cleared at power-on. This prevents a PON event from clearing SRQ at power-on.
- **0** or **OFF**: Saves the contents of the Service Request Enable and the Standard Event Status enable registers in non-volatile memory and recalls them at power-on. This allows a PON event to generate SRQ at power-on.

The query returns the current state of *PSC.

Command syntax	*PSC <bool>
Parameters	0 | 1 | OFF | ON
Examples	`*PSC 0` `*PSC 1`
Query syntax	*PSC?
Returned parameters	0 | 1
Related commands	*ESE *SRE

*RCL

WARNING

Recalling a previously stored state may place hazardous voltages at the ac source output.

Rappeler un état précédemment mémorisé peut placer des tensions dangereuses au niveau de la sortie de la source ac.

For standard Keysight 681xC series, 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)

NOTE

The device state stored in location 0 is automatically recalled at power turn-on when the OUTPut:PON:STATE is set to RCL0.

For Keysight 6812C/6813C Option 026 only, this command includes the auxiliary output voltage and phase.

Command syntax	*RCL <NRf>
Parameters	0 to 15
Examples	`*RCL 3`
Related commands	*PSC *RST *SAV

*RST

This command resets the ac source to the following factory-defined states:

CAL:STAT	OFF	[SOUR:]FUNC	SIN
DISP:STAT	ON	[SOUR:]FUNC:CSIN	100%
DISP:MODE	NORM	[SOUR:]LIST:COUN	1
INIT:CONT:SEQ	0	[SOUR:]LIST:STEP	AUTO
OUTP	OFF	[SOUR:]PHAS	0
OUTP:COUP	AC	[SOUR:]PHAS:AUX [a]	0
OUTP:DFI	OFF	[SOUR:]PHAS:MODE	FIX
OUTP:DFI:SOUR	OFF	[SOUR:]PHAS:TRIG	0
OUTP:IMP	OFF	[SOUR:]PULS:COUN	1
OUTP:IMP:REAL	0	[SOUR:]PULS:DCYC	50%
OUTP:IMP:REAC	0.00005	[SOUR:]PULS:HOLD	WIDT
OUTP:PROT:DEL	100 ms	[SOUR:]PULS:PER	0.03333 s
OUTP:RI:MODE	OFF	[SOUR:]PULS:WIDT	0.01667 s
OUTP:TTLT	OFF	[SOUR:]VOLT	0
OUTP:TTLT:SOUR	BOT	[SOUR:]VOLT:AUX[a]	0
SENS:CURR:ACDC:RANG	MAX	[SOUR:]VOLT:MODE	FIX
SENS: NSEL	1	[SOUR:]VOLT:OFFS	0
SENS:FREQ:SOUR	VOLT	[SOUR:]VOLT:OFFS:MODE	FIX
SENS:SWE:OFFS:POIN	0	[SOUR:]VOLT:OFFS:TRIG	0
SENS:SWE:TINT	25 µs	[SOUR:]VOLT:OFFS:SLEW	INF
SENS:WIND	KBES	[SOUR:]VOLT:OFFS:SLEW:MODE	FIX
[SOUR:]CURR	MAX	[SOUR:]VOLT:OFFS:SLEW:TRIG	INF
[SOUR:]CURR:PEAK	13 A/26 A/32 A[b]	[SOUR:]VOLT:PROT	MAX
[SOUR:]CURR:PEAK:MODE	FIX	[SOUR:]VOLT:PROT:STAT	OFF

[SOUR:]CURR:PEAK:TRIG	13 A/26 A/32 A[b]	[SOUR:]VOLT:RANG	MAX
[SOUR:]CURR:PROT:STAT	OFF	[SOUR:]VOLT:SENS:DET	RTIM
[SOUR:]FREQ	60 Hz	[SOUR:]VOLT:SENS:SOUR	INT
[SOUR:]FREQ:MODE	FIX	[SOUR:]VOLT:SLEW	INF
[SOUR:]FREQ:SLEW	INF	[SOUR:]VOLT:SLEW:MODE	FIX
[SOUR:]FREQ:SLEW:MODE	FIX	[SOUR:]VOLT:SLEW:TRIG	INF
[SOUR:]FREQ:SLEW:TRIG	INF	[SOUR:]VOLT:TRIG	0
[SOUR:]FREQ:TRIG	60 Hz	TRIG:DEL	0
[SOUR:]FM	OFF	TRIG:SOUR	BUS
[SOUR:]FM:DEV	0	TRIG:SEQ2:SOUR	IMM
[SOUR:]FM:FREQ	0.1	TRIG:SEQ2:PHAS	0
[SOUR:]FUNC:MODE	FIX	TRIG:SEQ3:SOUR	BUS
[SOUR:]FUNC:TRIG	SIN		

[a] This command is for Keysight 6812C/6813C Option 026 only

[b] 13 A for Keysight 6811C and 6812C; 26 A for Keysight 6813C; and 32 A for Keysight 6813C Option 019

NOTE

- *RST does not clear any of the status registers or the error queue, and ac source 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.

For Keysight 6812C/6813C Option 026 only, this command sets the auxiliary output voltage and phase to zero.

Command syntax	*RST	
Parameters	None	
Related commands	*PSC	*SAV

*SAV

This command stores the present state of the ac source to a specified location in memory. Up to 16 states can be stored in non-volatile 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.

For Keysight 6812C/6813C Option 026 only, this command includes the auxiliary output voltage and phase.

Command syntax	*SAV <NRf>
Parameters	0 to 15
Example	`*SAV 3`
Related commands	*PSC *RST *RCL

*SRE

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (refer to "*STB?" on page 217 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. Refer to "Programming the Status Registers" on page 259 in Chapter 5, "Programming Examples" 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	Refer to *PSC
Example	`*SRE 128`
Query syntax	*SRE?
Returned parameters	<NR1> (register binary value)
Related commands	*ESE *ESR *PSC

*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 (refer to “Programming the Status Registers” on page 259 in Chapter 5, "Programming Examples" 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 configuration of Status Byte register

Bit position	7	6	5	4	3	2 - 0
Bit name	OPER	MSS RQS	ESB	MAV	QUES	n.u.
Bit weight	128	64	32	16	8	

n.u. = Not used
OPER = Operation status summary
MSS = Master status summary
RQS = Request for service
ESB = Event status byte summary
MAV = Message available
QUES = Questionable status summary

Query syntax	*STB?
Parameters	None
Returned parameters	<NR1> (register value)
Related commands	*SRE *ESR *ESE

*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

*TST?

This query causes the ac source to do a self-test and report any errors.

Query syntax	*TST?	
Parameters	None	
Returned parameters	<NR1>	0 indicates the ac source has passed self-test. Non-zero indicates an error code (refer to Appendix C, "Error Messages")

*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 ac source a GPIB DCL (Device Clear) command.

Command syntax	*WAI
Parameters	None
Related commands	*OPC

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

4 Compatibility of the B and C Versions of the AC Power Solutions

This chapter provides the compatibility of the B and C versions of the ac source. The Persona commands described in this section are provided for compatibility with the existing 6811B/6812B/6813B and 6811C/6812C/6813C ac source. This section also compares the new 6811C/6812C/6813C features to the previous 6811B/6812B/6813B ac source.

Persona Commands

Persona commands are provided to allow users to change the programmatic identity of the 6811C/6812C/6813C mainframes back to the “B” version mainframes for code compatibility. The manufacturer string can also be changed for code compatibility. The persona commands are non-volatile.

Model number

The following SCPI allows you to change the model number ID of an 6811C/6812C/6813C to 6811B/6812B/6813B:

SYST:PERS:MOD “6811B”

To return the model number to the factory shipped setting:

SYST:PERS:MOD:DEF

Manufacturer ID

To change the manufacturer ID to Agilent Technologies:

SYST:PERS:MAN "Agilent"

To return the manufacturer to the factory shipped setting:

SYST:PERS:MAN:DEF

New Feature Comparison with B and C AC Source

Feature	6811C/6812C/6813C implementation	6811B/6812B/6813B implementation
Color	Phantom gray	Quartz gray
Start-up time[a]	Approximately 11 seconds	Approximately 12 seconds
Programming language	SCPI only	SCPI and Elgar
Front panel page	Added System Info, LAN Menu, and LAN Reset	Not available
IO connectivity	No selection required: GPIB, RS232, USB, and LAN	Requires selection: GPIB or RS232
Compliance	LXI 1.4 Core	Not available
RS232 baud rate	1200, 2400, 4800, 9600	300, 600, 1200, 2400, 4800, 9600
Ethernet connectivity	Up to 100 Mbps	Not available
***IDN response**	Keysight Technologies 6811C/6812C/6813C with serial number	Agilent Technologies 6811B/6812B/6813B without serial number
Firmware update method	Firmware update utility only	Require service engineer to change ROM
ID strings – LAN, USB, GPIB, and Web page	Keysight, K-<hostname>	Agilent, A-<hostname>
RI mode	OFF	LATChing
Web control	Available	Not available
Calibration count	CALibration:COUNt?	Returns -113 Undefined header
Calibration date	CALibration:DATE	Returns -113 Undefined header
Calibration password	CALibration:SECure:CODE	Returns -113 Undefined header
Calibration state	CALibration:SECure:STATe	Returns -113 Undefined header
Calibration string	CALibration:STRing	Returns -113 Undefined header
Clear front panel display text	DISPlay[:WINDow]:TEXT:CLEar	Returns -113 Undefined header
LXI identify and mDNS commands	LXI:IDENtify[:STATe] LXI:MDNS[:STATe]	Returns -113 Undefined header

Feature	6811C/6812C/6813C implementation	6811B/6812B/6813B implementation
Query resolved mDNS hostname	LXI:MDNS:HNAMe[:RESolved]?	Returns -113 Undefined header
Set and query desired mDNS service name	LXI:MDNS:SNAMe:DESired LXI:MDNS:SNAMe[:RESolved]?	Returns -113 Undefined header
Reset LAN settings to known operating state	LXI:RESet	Returns -113 Undefined header
Restart LAN with specified settings	LXI:RESTart	Returns -113 Undefined header
Configure and query LAN settings	SYSTem:COMMunicate:LAN:CONTrol? SYSTem:COMMunicate:LAN:DHCP SYSTem:COMMunicate:LAN:DNS SYSTem:COMMunicate:LAN:DOMain? SYSTem:COMMunicate:LAN:GATeway SYSTem:COMMunicate:LAN:HOSTname SYSTem:COMMunicate:LAN:IPADdress SYSTem:COMMunicate:LAN:MAC? SYSTem:COMMunicate:LAN:SMASk	Returns -113 Undefined header
Set and query command prompt and welcome message for Telnet	SYSTem:COMMunicate:LAN:TELNet:PROMpt SYSTem:COMMunicate:LAN:TELNet:WMESsage	Returns -113 Undefined header
Set the state of front panel	SYSTem:COMMunicate:RLSTate	Returns -113 Undefined header
Reads initial control connections port number	SYSTem:COMMunicate:TCPip:CONTrol?	Returns -113 Undefined header
PERSona commands switch manufacturer and model number for backward compatibility	SYSTem:PERSona:MANufacturer SYSTem:PERSona:MANufacturer:DEFault SYSTem:PERSona:MODel SYSTem:PERSona:MODel:DEFault	Returns -113 Undefined header
Security count	SYST:SECurity:COUNt?	Returns -113 Undefined header
Clears all user-accessible instrument memory	SECurity:IMMediate SYSTem:SECurity:IMMediate	Returns -113 Undefined header

Feature	6811C/6812C/6813C implementation	6811B/6812B/6813B implementation
Set the instrument state in binary format	SYSTem:SET	Returns -113 Undefined header
Returns binary string on EEPROM	*LRN?	Returns -113 Undefined header

[a] Start-up time from power on to when front panel display shows rms voltage and frequency readings.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

5 Programming Examples

This chapter provides an introduction to programming the ac source with SCPI commands.

Introduction

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

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

NOTE These examples in this chapter are generic SCPI commands. Refer to Chapter 2, "Introduction to Programming" for information about encoding the commands as language strings. Where appropriate, optional commands are shown for clarity in the examples.

Programming the Output

Power-on initialization

When the ac source is first turned on, it wakes up with the output state set to 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 Chapter 3, "Language Dictionary" to see how each programmable parameter is set by *RST. Refer to the *PSC command in Chapter 3, "Language Dictionary" 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 voltage and frequency

The AC rms output voltage is controlled with the VOLTage command. For example, to set the AC output voltage to 125 volts rms, use:

```
VOLTage 125
```

The ac source can be programmed to turn off its output if the AC output voltage exceeds a preset peak voltage limit. This protection feature is implemented with the VOLTage:PROTection command as explained in Chapter 3, "Language Dictionary".

Maximum voltage

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

```
VOLTage? MAX
```

The maximum voltage that the ac source can output is limited by the peak value of the waveform. This is 425 V peak on all models. Since the user programs output voltage in units of rms volts, the maximum value that can be programmed is dependent on the peak-to-rms ratio (crest factor) of the selected waveform. For a sine waveform, the maximum AC voltage that can be programmed is 300 V.

NOTE **Because voltage commands are coupled with the waveform shape and voltage offset commands, changing voltages without changing the waveform shape or voltage offset may result in an error if the resulting peak voltage amplitude exceeds the maximum voltage rating of the ac source. Refer to "Coupled commands" on page 235 for more information.**

Frequency

The output frequency is controlled with the FREQuency command. To set the output frequency to 50 Hz, use:

```
FREQuency 50
```

Voltage and frequency slew rates

Voltage slew

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

```
VOLTage:SLEW 20
```

At `*RST` the slew rate is set to INFinity, which means that AC voltage 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. Refer to "Programming Output Transients" on page 236.

NOTE Output voltage changes caused by the OUTPut:STATe or VOLTage:OFFSet commands, 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. Refer to "Programming Output Transients" on page 236.

Waveform shapes

At *RST, the ac source supplies a sine waveform, but other shapes can be selected. There are built-in tables for sine, square and clipped sine waveforms. In addition, the user can define arbitrary waveshapes by creating a 1024 point table of amplitudes for a single cycle.

As shown in the following examples, the FUNCtion[:SHAPe] command selects the output waveform.

Square waveform

To select the square output waveform, use:

```
FUNCtion:SHAPe SQUare
```

Clipped waveform

To select a clipped sine waveform use:

```
FUNCtion:SHAPe CSINusoid
```

To set the clipping level to 50%, use:

```
FUNCtion:SHAPe:CSINusoid 50
```

The clipping level is the percentage of the peak amplitude at which clipping occurs.

The clipping level can also be specified in terms of the percent total harmonic distortion in the clipped sine waveform by adding a THD suffix to the command. For example,

```
FUNCtion:SHAPe:CSINusoid 10 THD
```

sets the clipping level so that the clipped sine has 10% distortion.

User-defined waveform

To create a user-defined waveform, use TRACe:DEFine command to create a name for the waveform, then use the TRACe[:DATA] command to send the list of 1024 amplitude points. The waveform can then be selected using the FUNCtion command. For example, a waveform named “Distortion” can be created with:

```
TRACe:DEFine DISTORTION
TRACe:DATA DISTORTION, n1, n2, n3, ..., n1024
```

where n1 ... n1024 are the data points that define the relative amplitudes of exactly one cycle of the waveform. The first data point defines the amplitude that will be output at 0 degrees phase reference.

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 voltage is output when the waveform is selected. When queried, trace data is returned as normalized values in the range of 1. Waveform data is stored in non-volatile memory and is retained when input power is removed. Up to 12 user defined waveforms may be created and stored.

NOTE

Because waveform shape commands are coupled with the voltage commands, changing waveforms without changing the programmed voltage may result in an error if the resulting peak voltage amplitude exceeds the maximum voltage rating of the ac source. Refer to “Coupled commands” on page 235 for more information.

Programming the output phase

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

```
PHASe <n>
```

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

In the single-phase models, the only discernible effect of using the PHASe command is to cause an instantaneous shift in output waveform phase. This is because the internal phase reference is not accessible externally.

Current limit

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

```
CURRent <n>
```

where <n> is the rms current limit in amperes.

If the load attempts to draw more current than the programmed limit, the output voltage is reduced to keep the rms current 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 current limit is not instantaneous. When the output voltage 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 current limit is reached. This protection feature is implemented with the CURRent:PROTection:STATe command as explained in Chapter 3, "Language Dictionary".

Peak current limit

The Keysight 6811C, 6812C, and 6813C have a programmable peak current limit which limits the instantaneous current. The command to set this limit is:

```
CURRent:PEAK <n>
```

where <n> is the peak current in amperes.

Since instantaneous current tends to be highest when the output voltage is highest, this current limit tends to clip the peaks of the output voltage waveform.

DC output

The Keysight 6811C, 6812C, and 6813C single-phase models have DC output capability which lets you independently control the DC and AC components of the output voltage. At *RST, these models default to the AC output mode. When DC offset is desired, this mode must be changed using the OUTPut:COUPling command.

To enable the DC output, use:

```
OUTPut:COUPling DC
```

To set the DC output voltage to 5 volts, use:

```
VOLTage:OFFSet 5
```

When the command

```
OUTPut:COUPling AC
```

is sent, the ac source regulates the DC output voltage to 0, regardless of any programmed voltage offset.

NOTE

Because the voltage offset commands are coupled with the voltage commands, the DC output voltage will affect the maximum output voltage and vice-versa. When the DC output is non-zero, the maximum AC voltage that can be programmed is reduced to a value that limits the total AC+DC peak amplitude to 425 V. Similarly, when the AC output is non-zero the maximum DC offset that can be programmed is subject to the same limitation. Refer to “Coupled commands” on page 235 for more information.

Coupled commands

This section describes how to avoid programming errors that may be caused because of the error checking done for coupled commands.

VOLTage:LEVel, VOLTage:OFFSet, and FUNCtion:SHAPe

When using these commands, assume the present state of the ac source has AC amplitude set to 240 volts rms and DC offset set to 0, and a new state is desired with AC amplitude of 0 and DC offset of 300 volts. If the commands

```
VOLTage:OFFSet 300
VOLTage 0
```

are sent individually, an error will be generated because the first command requests an output state that exceeds the peak voltage capability. The error can be avoided by reversing the order in which the commands are sent.

Another way to avoid this type of error with coupled commands is to give the commands together as part of one terminated message as in

```
VOLTage:OFFSet 300;:VOLTage 0
```

When coupled commands are sent this way, the couplings are deferred and resolved when the newline terminator is received.

Other commands that are coupled to the VOLTage, VOLTage:OFFSet, and SHAPe commands are the output transient commands that control step, pulse and list generation. When an output transient is initiated (ready to receive a trigger), the error checking that takes place for maximum peak output voltage includes any combination of voltage, voltage offset, or function shape that can occur during the transient.

Programming Output Transients

Output transients are used to:

- Synchronize output changes with a particular phase of the voltage 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 voltage

Frequency

Phase

Waveform shape

AC voltage slew rate

Frequency slew rate

DC output voltage

Peak current limit

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.

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

Transient system model

Figure 5-1 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. 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.

NOTE The AC voltage, waveform shape, and voltage slew functions cannot be set to Step or Pulse mode if one of them is set to List mode.

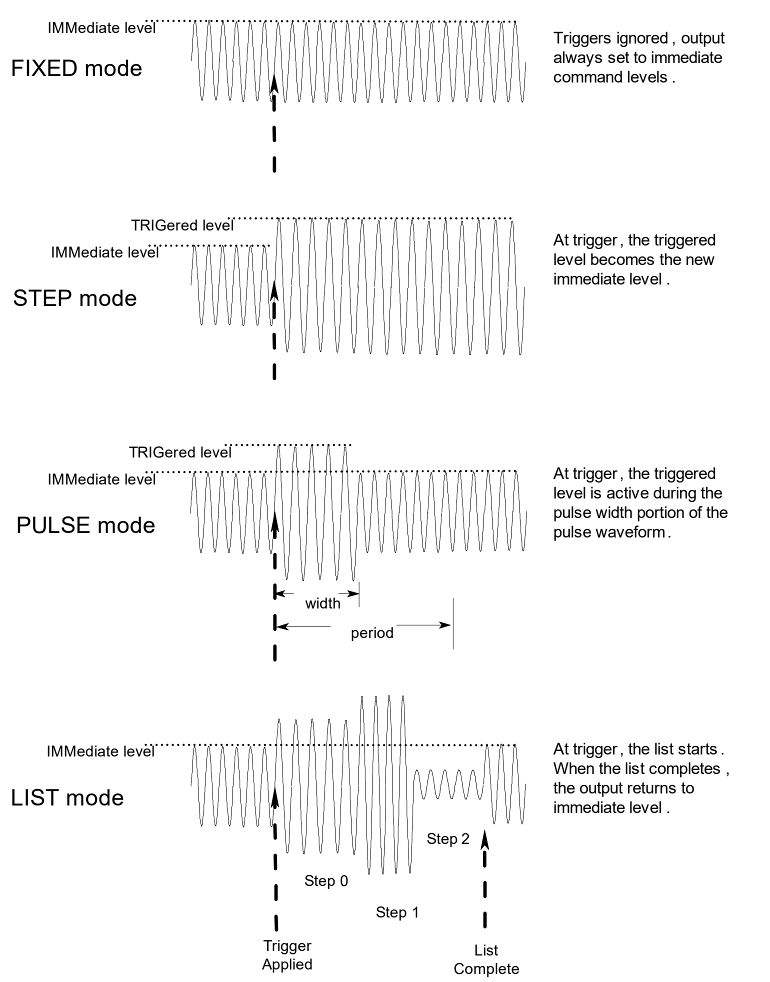

Figure 5-1 Model of transient system

Step and pulse transients

Step 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
VOLTage:MODE STEP
```

Step 2 Set the triggered level of the function that will generate the transient. For example, if the previously programmed voltage function is going to step the output voltage amplitude to 150 volts upon receipt of a trigger, use:

```
VOLTage:TRIGger 150
```

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

```
TRIGger:TRANsient:SOURce EXTernal
```

Trigger sources are discussed in detail under “Triggering Output Changes” on page 244.

Step 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)

Step 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
```

To enable the transient system indefinitely use:

```
INITiate:CONTinuous:SEQuence1 ON
```

Step 6 Trigger the transient. This is described in detail under "Triggering Output Changes" on page 244.

Example

The following example programs a voltage dropout for two cycles of a 120 volt, 60 Hz output. The dropout begins at the positive peak of the output voltage waveform (90 degrees phase) and is triggered by GPIB bus trigger.

`*RST`	Begin at power-on state
`VOLT 120`	Set initial output voltage (immediate-level)
`FREQ 60`	Set initial output frequency
`OUTP ON`	Enable the output
`VOLT:MODE PULS`	Enable output to generate pulses when triggered
`VOLT:TRIG 0`	Set the voltage dropout (triggered level)
`PULS:WIDT .03333`	Set pulse width for 2 periods
`TRIG:SOUR BUS`	Respond to GPIB 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 GPIB bus trigger

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 non-volatile 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 voltage and frequency changes.

Step 1 Set the mode of each function that will participate in the output sequence to LIST. For example:

```
VOLTage:MODE LIST
FREQuency:MODE LIST
```

Step 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 that includes nominal line, high line, and low line, a list may include the following values:

```
LIST:VOLTage 120, 132, 108, 120, 132, 108, 120, 132, 108
```

You can specify lists for more than one function. For example, to synchronize the previous voltage list with another list that varies the output frequency from nominal, to high, to low, the lists may include the following values:

```
LIST:VOLTage 120, 132, 108, 120, 132, 108, 120, 132, 108
LIST:FREQuency 60, 60, 60, 63, 63, 63, 57, 57, 57
```

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:VOLTage 120, 130, 140, 150;FREQuency 60
```

is the same as:

```
LIST:VOLTage 120, 130, 140, 150
LIST:FREQuency 60, 60, 60, 60
```

Step 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.

Step 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.

Step 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.

Step 6 Use the transient trigger system to trigger the list. This is described in detail under “Triggering Output Changes” on page 244.

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" on page 254.

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 GPIB bus triggers.
EXTernal	selects the external Trigger In BNC connector.

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

Output trigger system model

Figure 5-2 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.

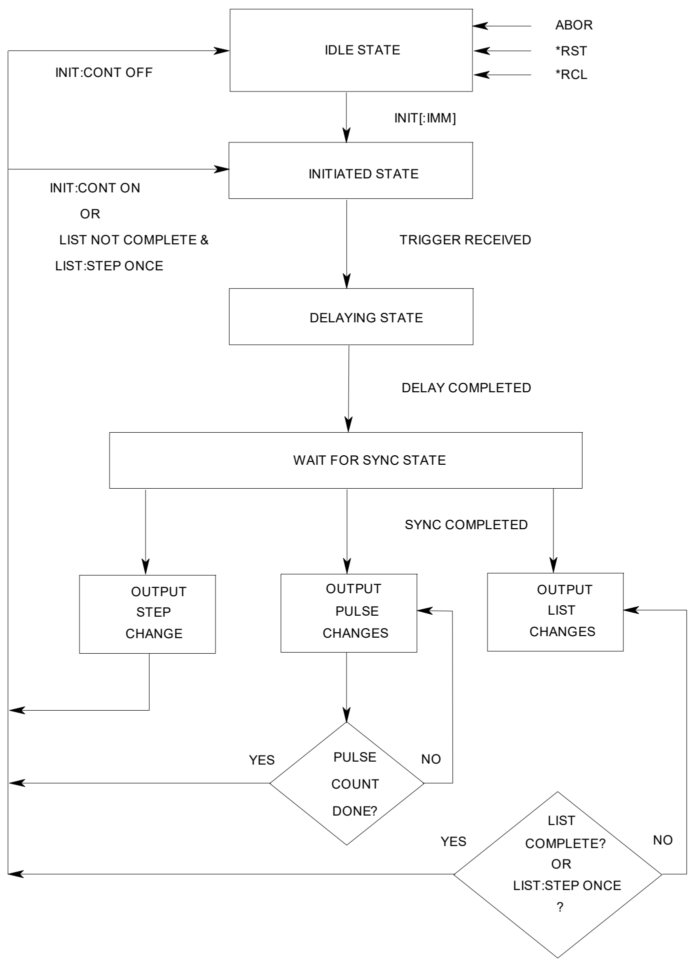

Figure 5-2 Model of output triggers

Initiating the output 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 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`

NOTE The SEQuence2 (or SYNChronize) trigger sequence does not have an INITiate command. It is always initiated.

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 In BNC as the source, use:

`TRIGger:SEQuence1:SOURce EXTernal` or

`TRIGger:TRANsient:SOURce EXTernal`

To select GPIB 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`

NOTE **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.

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.

NOTE **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.

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.

Generating output 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 GPIB:

  `TRIGger:SEQuence1:IMMediate`

  `<GET>` (Group Execute Trigger)

- By applying a signal with a high-to-low transition to the Trig In BNC connector.

- By pressing the front panel Trigger key when the unit is operating in local mode.

Continuous triggers

- By sending the following commands over the GPIB:

```
TRIGger:SEQuence1:SOURce IMMediate
```

```
INITiate:CONTinuous:SEQuence1 ON
```

When the trigger system enters the Output Change state upon receipt of a trigger (refer to Figure 5-2), the triggered functions are set to their programmed trigger levels. When the triggered actions are completed, the trigger system returns to the Idle state.

Specifying a dwell time for each list point

Each voltage 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 (refer to Figure 5-2).

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.

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” on page 254.

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.

Voltage and current measurements

The SCPI interface provides a number of MEASure and FETCh queries that return various components of rms voltage and current. For example, to read the AC component of the rms voltage or current, use:

`MEASure:VOLTage:AC?` or

`MEASure:CURRent:AC?`

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 and the current crest factor, use:

`MEASure:CURRent:AMPLitude:MAXimum?`

`MEASure:CURRent:CREStfactor?`

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

Harmonic measurements

The MEASure and FETCh queries can return the amplitude and phase of up to the 50th harmonic of voltage and current. They can also return the total harmonic distortion in the output voltage or current. For example, to return readings for an individual harmonic component, use the following commands:

```
MEASure:CURRent:HARMonic:AMPLitude? <harmonic_number>
MEASure:CURRent:HARMonic:PHASe? <harmonic_number>
MEASure:VOLTage:HARMonic:AMPLitude? <harmonic_number>
MEASure:VOLTage:HARMonic:PHASe? <harmonic_number>
```

Harmonic numbers are related to the programmed frequency of output voltage. Queries sent with an argument of 0 return the DC component. An argument of 1 indicates the fundamental frequency, 2 indicates the second harmonic, 3 indicates the third, and so on. The maximum harmonic component that can be read is limited by the fundamental measurement bandwidth, which is 12.5 kHz. An error is generated if a query is sent for a harmonic that has a frequency greater than 12.5 kHz. To return all the harmonic components with a single query, use the following commands:

```
MEASure:ARRay:CURRent:HARMonic:AMPLitude?
MEASure:ARRay:CURRent:HARMonic:PHASe?
MEASure:ARRay:VOLTage:HARMonic:AMPLitude?
MEASure:ARRay:VOLTage:HARMonic:PHASe?
```

These queries always return 51 data values, from the DC component up to the 50th harmonic. Any harmonics that represent frequencies greater than 12.5 kHz are returned as the value 0. 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?
```

Returning voltage and current data from the data buffer

The MEASure and FETch queries can also return all 4096 data values of the instantaneous voltage and current buffers. These are as follows:

```
MEASure:ARRay:CURRent?
MEASure:ARRay:VOLTage?
```

Regulatory-compliant measurement of quasi-stationary harmonics

In order to meet regulatory requirements of IEC-555 and other standards that specify how quasi-stationary harmonics are to be measured, the ac source has a command that alters both the output frequency control and the harmonic measurement algorithms to meet these requirements. The command is

```
SENSe:WINDow KBESsel | RECTangular
```

Triggering Measurements

You can use the data acquisition trigger system to synchronize the timing of the voltage and current data acquisition with an external trigger source. Then use the FETCh commands to return different calculations from the data acquired by the measurement trigger. The following measurement trigger sources can be selected:

BUS	selects GPIB bus triggers.
EXTernal	selects the external Trigger In BNC connector.
TTLTrg	selects the signal driving the Trigger Out BNC connector.

SCPI triggering nomenclature

As previously explained under "Triggering Output Changes" on page 244, 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

Measurement trigger system model

Figure 5-3 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.

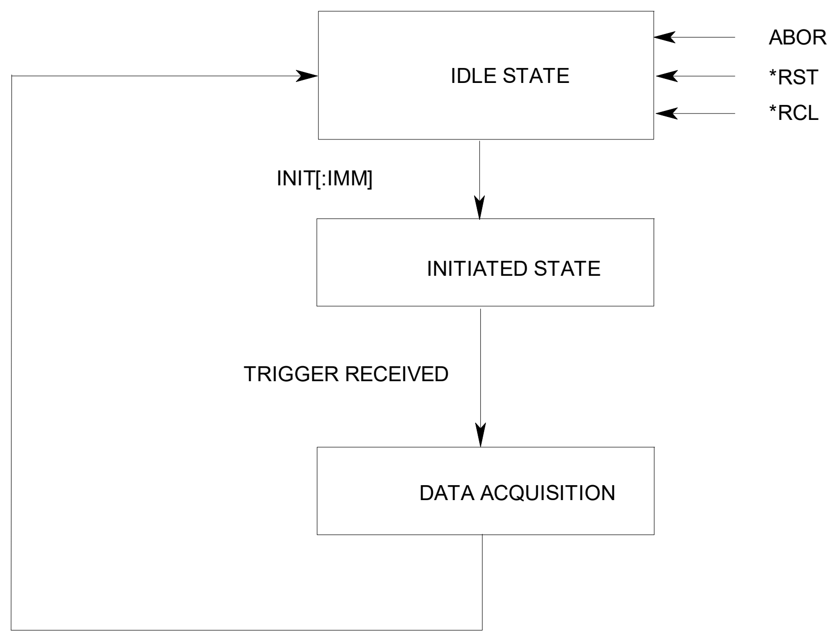

Figure 5-3 Model of measurement triggers

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.**

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 In BNC as the source, use:

`TRIGger:SEQuence3:SOURce EXTernal` or

`TRIGger:ACQuire:SOURce EXTernal`

To select GPIB 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 Out BNC connector, use:

`TRIGger:SEQuence3:SOURce TTLTrg` or

`TRIGger:ACQuire:SOURce TTLTrg`

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 GPIB:

```
TRIGger:SEQuence3:IMMediate
TRIGger:ACQuire:IMMediate
*TRG
<GET> (Group Execute Trigger)
```

– By applying a signal with a high-to-low transition to the Trig In BNC connector.

– By generating an output transient that causes the Trig Out BNC connector to output a pulse.

– By pressing the front panel Trigger key when the unit is operating in local mode.

Controlling the instantaneous voltage and current data buffers

Varying the voltage and current sampling rate

At *RST, the output voltage and current sampling rate is 40 kHz (period = 25 s). This means that it takes about 100 ms 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 25 μs (the default), to 250 μs in 25 μs steps.

Pre-event and post-event triggering

The ac source continuously samples the instantaneous output voltage and current. While this is happening, you can move the block of data that is being read into the voltage and current buffers with respect to the data acquisition trigger. This permits pre-event or post-event data sampling. To offset the starting point of the data buffer relative to the acquisition trigger, use:

```
SENSe:SWEep:OFFSet:POINts <offset>
```

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 X Sample period).

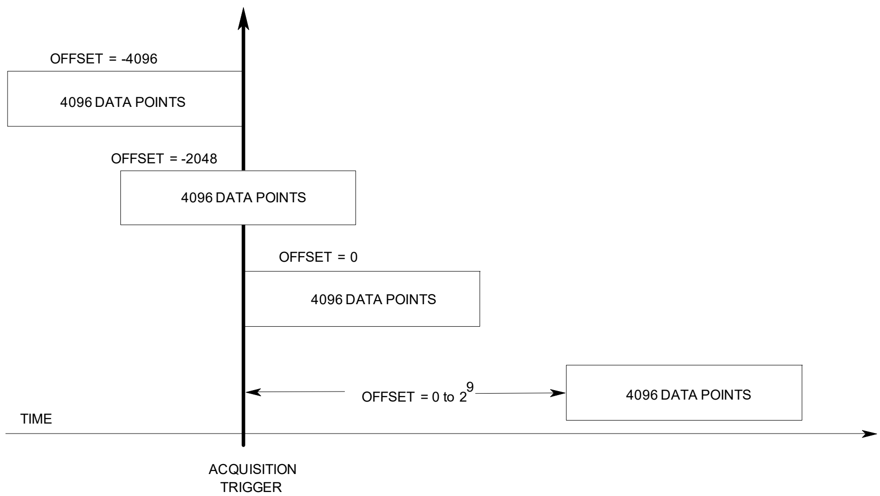

Figure 5-4 Pre-event and post-event triggering

Programming the Status 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 5-5 shows the status register structure of the ac source. Table 5-1 defines the status bits. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard GPIB 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.

Power-on conditions

Refer to the *RST command description in Chapter 3, "Language Dictionary" for the power-on conditions of the status registers.

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 3, "Language Dictionary". 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 3, "Language Dictionary". It is a read/write register.

Register	Command	Description
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.

The outputs of the Operation Status register group are logically-ORed into the OPER(ation) summary bit (7) of the Status Byte register.

Table 5-1 Bit configuration of status registers

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
2	SOA	The safe operating area protection has tripped
3	UNR	The output is unregulated
4	OT	An overtemperature condition has occurred
9	RI	The remote inhibit state is active
10	CLpeak	The peak current limit circuit is active
11	Rail	The rail protection circuit has tripped
12	CLrms	The rms current limit circuit is active
14	MeasOvld	Current measurement exceeded capability of low range

Table 5-1 Bit configuration of status registers (continued)

Bit	Signal	Meaning
		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

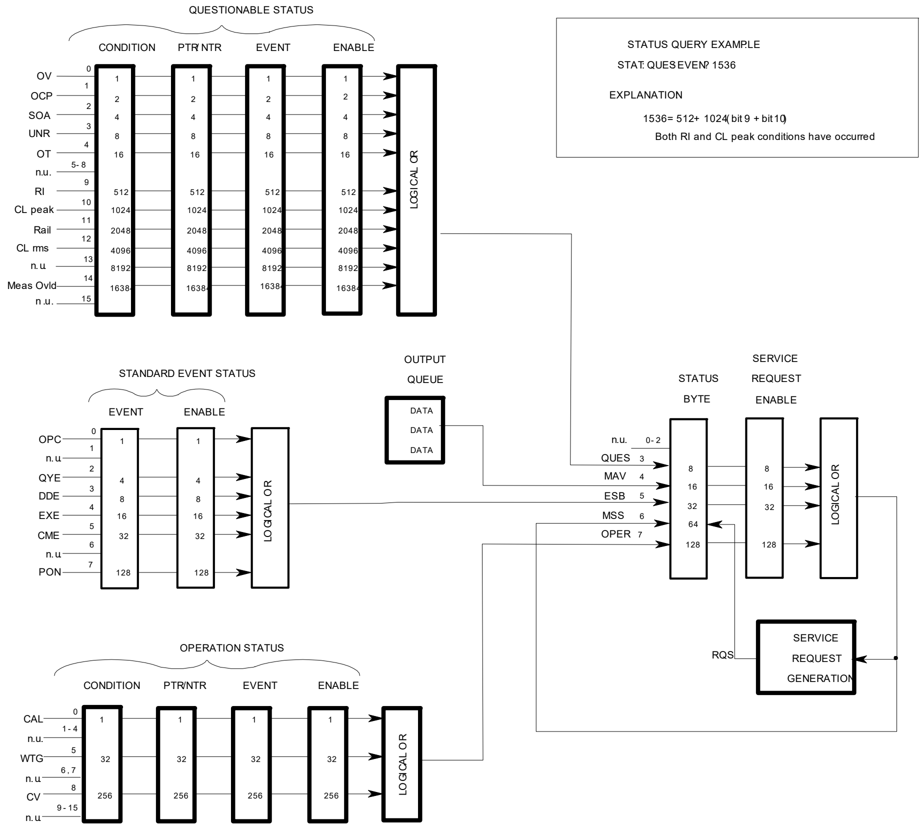

Figure 5-5 AC source status model

Questionable Status Group

The Questionable Status registers record signals that indicate abnormal operation of the ac source. As shown in the Figure 5-5, 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 3, "Language Dictionary". 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 3, "Language Dictionary". 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.

The outputs of the Questionable Status group are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register.

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 (refer to Figure 5-5). 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 as follows:

```
*PSC OFF
*ESE 128
*SRE 32
```

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 5-1.

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 (4) of the Status Byte register.

Examples

Determining the cause of a service interrupt

You can determine the cause for an SRQ by the following actions:

Step 1 Determine which summary bits are active. Use:

`*STB?` or

Serial poll

Step 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?`

When an Event register is read, it is cleared. This also clears the corresponding summary bit.

Step 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.

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 5-5, 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:

Step 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)

Step 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`

Step 3 Program the Service Request Enable register to allow the QUES summary bit from the Status Byte register to generate RQS. Use:

`*SRE 8`

Step 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?`

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.

Programming the Trigger In and Trigger Out BNC Connectors

The ac source has two BNC connectors labeled Trigger In and Trigger Out (refer to Figure 5-6). Refer to *"Trigger in/Trigger out characteristics" in Appendix A of the User's Guide* for the electrical parameters.

Trigger in BNC

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 BNC

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 TTLTor
TRIGger:ACQuire:SOURce TTLT
```

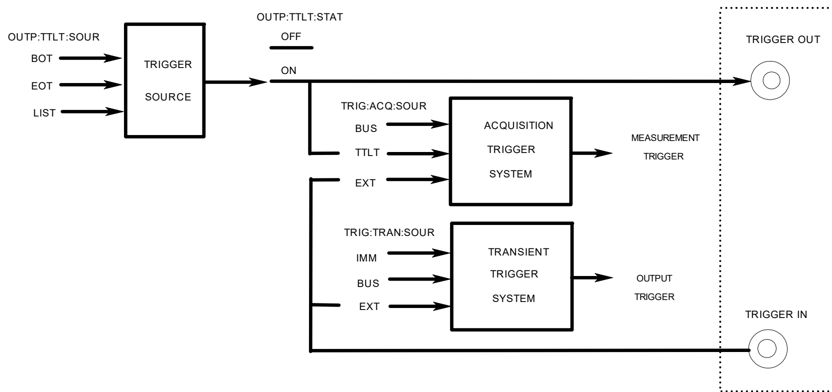

Figure 5-6 BNC connector trigger model

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 *"INH/FLT characteristics" in Appendix A of the User's Guide* for the electrical parameters.

Remote inhibit (RI)

Remote inhibit is an external 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.

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
```

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 3, "Language Dictionary". 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.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

A SCPI Command Tree

This appendix provides the SCPI command tree.

Command Syntax

The following subsystem commands are listed in this appendix:

CALibrate | CALibration

:COUNt?
:CURRent
    :AC
    :EXTernal
    :MEASure
:DATA <n>
:DATA?
:DATE <quoted_string>
:IMPedance
:LEVel <level>
:PASSword <n>
:PHASe
    :AUXiliary
:PWM
    :FREQuency <n>
    :RAMP <n>
:SAVE
:SECure
    :CODE <quoted_string>
    :STATe <bool>[, <n>]
:SHUNt <n>
:STATe <bool>[, <n>]
:STRing <quoted_string>
:VOLTage
    :AC
    :AUXiliary
        [:AC]
    :DC
    :EXTernal
    :OFFSet
    :PROTection

DISPlay

[:WINDow]

[:STATe] <bool>

:MODE <mode>

:TEXT

[:DATA] <quoted_string>

:CLEar

LXI

- :IDENTify
  - [:STATe] <bool>
- :MDNS
  - [:STATe] <bool>
  - :ENABle
  - :HNAMe
    - [:RESolved]?
  - :SNAMe
    - :DESired <quoted_string>
    - [:RESolved]?
  - :RESet
  - :RESTart

MEASure | FETCh

```
:ARRay
    :CURRent
        [:DC]?
        :HARMonic
            [:AMPLitude]?
            :PHASe?
            [:DC]?
    :VOLTage
        [:DC]?
        :HARMonic
            [:AMPLitude]?
            :PHASe?
```

MEASure | FETCh

```
[:SCALar]
    :CURRent
        [:DC]?
        :AC?
        :ACDC?
        :AMPLitude
            :MAXimum?
        :CREStfactor?
        :HARMonic
            [:AMPLitude]? <n>
            :PHASe? <n>
            :THD?
    :FREQuency?
    :POWer
        [:DC]?
        :AC
            [:REAL]?
            :APParent?
            :REACtive?
            :PFACtor?
    :TVOLt? <level>, <occurrence>
    :TVOL
        :ABSolute? <level>, <start_time>, <min_pluse_width>
    :VOLTage
        [:DC]?
        :AC?
        :ACDC?
        :HARMonic
            [:AMPLitude]? <n>
            :PHASe? <n>
            :THD?
        :RANKed? <n>
```

OUTPut

```
[:STATe] <bool>
:COUPling <coupling>
:DFI
    [:STATe] <bool>
    :SOURce <source>
:IMPedance
    [:STATe] <bool>
    :REAL <n>
    :REACtive <n>
:PON
    :STATe <state>
:PROTection
    :CLEar
    :DELay <n>
:RI
    :MODE <mode>
:TTLTrg
    [:STATe] <bool>
    :SOURce <source>
```

SENSe

:CURRent
    :ACDC
        :RANGe
            [:UPPEr] <n>
:FREQuency
    :SOURce <source>
:NSELect <n>
:SWEep
    :OFFSet
        :POINts <n>
    :TINTerval <n>
:WINDow
    [:TYPE] <type>

[SOURce:]

- CURRent
  - [:LEVel]
    - [:IMMediate]
      - [:AMPLitude] <n>
  - :PEAK
    - [:IMMediate] <n>
    - :MODE <mode>
    - :TRIGgered <n>
  - :PROTection
    - :STATe <bool>

- FREQuency
  - [:CW | :IMMediate] <n>
  - :MODE <mode>
  - :SLEW
    - [:IMMediate] <n> | INFinity
    - :MODE <mode>
    - :TRIGgered <n> | INFinity
  - :TRIGgered <n>

- FM
  - [:STATe] <bool>
    - :DEViation <n>
    - :FREQuency <n>

- FUNCtion
  - [:SHAPe]
    - [:IMMediate] <shape>
    - :MODE <mode>
    - :TRIGgered <shape>
    - :CSINusoid <n> [THD]

[SOURce:]

```
LIST
    :COUNt <n> | INFinity
    :CURRent <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 <step>
    :TTLTrg <bool>{, <bool>}
        :POINts?
    :VOLTage
        [:LEVel] <n>{, <n>}
            :POINts?
        :SLEW <n>{, <n>}
            :POINts?
        :OFFSet <n>{, <n>}
            :POINts?
            :SLEW <n>{, <n>}
                :POINts?
```

[SOURce:]

PHASe

    [:ADJust | :IMMediate] <n>
    :AUXiliary
        [:ADJust | :IMMediate] <n>
    :MODE <mode>
    :TRIGgered <n>

PULSe

    :COUNt <n> | INFinity
    :DCYCle <n>
    :HOLD <parameter>
    :PERiod <n>
    :WIDTh <n>

[SOURce:]

```
VOLTage
    [:LEVel]
        [:IMMediate]
            [:AMPLitude] <n>
        :TRIGgered
            [:AMPLitude] <n>
    :AUXiliary
        [:LEVel]
            [:IMMediate]
                [:AMPLitude] <n>
    :MODE <mode>
    :OFFSet
        [:IMMediate] <n>
        :MODE <mode>
        :TRIGgered <n>
        :SLEW
            [:IMMediate] <n> | INFinity
            :MODE   <mode>
            :TRIGgered <n> | INFinity
    :PROTection
        [:LEVel] <n>
        :STATe <bool>
    :SENSe | :ALC
        :DETector <type>
        :SOURce <source>
    :SLEW
        [:IMMediate] <n> | INFinity
        :MODE   <mode>
        :TRIGgered <n> | INFinity
```

STATus

- :PRESet
- :OPERation
    - [:EVENt]?
    - :CONDition?
    - :ENABle <n>
    - :NTRansition <n>
    - :PTRansition <n>
- :QUEStionable
    - [:EVENt]?
    - :CONDition?
    - :ENABle <n>
    - :NTRansition <n>
    - :PTRansition <n>

SECurity

[:IMMediate]

SYSTem

```
:COMMunicate
    :LAN
        :CONTrol?
        :DHCP
        :DNS[{1|2}]
        :DOMain?
        :GATeway <quoted_address>
        :HOSTname <quoted_string>
        :IPADdress <quoted_address>
        :MAC?
        :SMASk <quoted_address>
        :TELNet
            :PROMpt<quoted_string>
            :WMESsage <quoted_string>
        :UPDate
    :RLSTate <mode>
    :TCPip
        :CONTrol?
    :ERRor?
    :LANGuage <language>
    :PERSona
        :MANufacturer <quoted_string>
            :DEFault
        :MODel <quoted_string>
            :DEFault
:SECurity
    :COUNt?
    [:IMMediate]
:SET <binary_block>
```

SYSTem

:VERSion?

:LOCal

:REMote

:RWLock

TRACe | DATA

:CATalog?

[:DATA] <waveform_name>, <n>{, <n>}

:DEFine <waveform_name>[, <waveform_name> | 1024]

:DELete

    [:NAME] <waveform_name>

ABORt

INITiate

[:IMMediate]
    :SEQuence[1|3]
    :NAME <name>
:CONTinuous
    :SEQuence[1] <bool>
    :NAME TRANsient <bool>

TRIGger

[:SEQuence1 | :TRANsient]
    [:IMMediate]
    :DELay <n>
    :SOURce <source>
:SEQuence2 | :SYNCronize
    :SOURce <source>
    :PHASe <n>
:SEQuence3 | :ACQuire
    [:IMMediate]
    :SOURce <source>
:SEQuence1
    :DEFine TRANsient
:SEQuence2
    :DEFine SYNChronize
:SEQuence3
    :DEFine ACQuire

SCPI Common Commands

*CLS

*ESE <n>

*ESE?

*ESR?

*IDN?

*LRN?

*OPC

*OPC?

*OPT?

*PSC <bool>

*PSC?

*RCL <n>

*RST

*SAV <n>

*SRE <n>

*SRE?

*STB?

*TRG

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

B SCPI Conformance Information

This appendix provides the SCPI conformance information. The ac source responds to SCPI Version 1992.0.

SCPI Confirmed Commands

ABOR

CAL:DATA

CAL:STAT

DISP[:WIND][:STAT]

DISP[:WIND]:TE‘XT[:DATA]

INIT[:IMM]

INIT[:IMM]:SEQ | NAME

INIT:CONT:SEQ | NAME TRAN

MEAS | FETC:ARR:CURR[:DC]?

MEAS | FETC:ARR:VOLT[:DC]?

MEAS | FETC[:SCAL]:CURR[:DC]?

MEAS | FETC[:SCAL]:CURR:AC?

MEAS | FETC[:SCAL]:FREQ?

MEAS | FETC[:SCAL]:POW[:DC]?

MEAS | FETC[:SCAL]:POW:AC[:REAL]?

MEAS | FETC[:SCAL]:VOLT[:DC]?

MEAS | FETC[:SCAL]:VOLT:AC?

OUTP[:STAT]

OUTP:COUP

OUTP:IMP[:STAT]

OUTP:PROT:CLE

OUTP:PROT:DEL

OUTP:TTLT[:STAT]

SENS:CURR:ACDC:RANG[:UPP]

SENS:SWE:OFFS:POIN

[SOUR:]CURR:PROT:STAT

[SOUR:]FREQ[:CW | :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[:ADJ | :IMM]

[SOUR:]PULS:COUN

[SOUR:]PULS:DCYC

[SOUR:]PULS:HOLD

[SOUR:]PULS:PER

[SOUR:]PULS:WIDT

[SOUR:]VOLT[:LEV][:IMM][:AMPL]

[SOUR:]VOLT:MODE

[SOUR:]VOLT:RANG

[SOUR:]VOLT:SENS | ALC:SOUR

TRAC | DATA:CAT?

TRAC | DATA[:DATA]

TRAC | DATA:DEF

STAT:PRES	TRAC | DATA:DEL[:NAME]		
STAT:OPER[:EVEN]?	TRIG[:SEQ1 | :TRAN][:IMM]		
STAT:OPER:COND?	TRIG[:SEQ1 | :TRAN]:DEL		
STAT:OPER:ENAB	TRIG[:SEQ1 | :TRAN]:SOUR		
STAT:OPER:NTR	TRIG:SEQ2 | SYNC:SOUR		
STAT:OPER:PTR	TRIG:SEQ3 | ACQ[:IMM]		
STAT:QUES[:EVEN]?	TRIG:SEQ3 | ACQ:SOUR		
STAT:QUES:COND?	*CLS		
STAT:QUES:ENAB	*ESE	*ESE?	*ESR?
STAT:QUES:NTR	*IDN?		
STAT:QUES:PTR	*OPC	*OPC?	*OPT?
SYST:ERR?	*PSC	*PSC?	
SYST:VERS?	*RCL	*RST	
SENS:SWE:TINT	*SAV	*SRE	*SRE?
SENS:WIND[:TYPE]	*STB?	*TRG	*TST?
[SOUR:]CURR[:LEV][:IMM][:AMPL]	*WAI		

Non SCPI Commands

CAL:COUN?
CAL:CURR:AC
CAL:CURR:EXT
CAL:CURR:MEAS
CAL:DATA?
CAL:DATE
CAL:IMP
CAL:LEV
CAL:PASS
CAL:PHAS:AUX
CAL:PWM:FREQ
CAL:PWM:RAMP
CAL:SAVE
CAL:SEC:CODE
CAL:SEC:STATE
CAL:SHUN
CAL:STR
CAL:VOLT:AC
CAL:VOLT:AUX[:AC]
CAL:VOLT:DC
CAL:VOLT:EXT
CAL:VOLT:OFFS
CAL:VOLT:PROT
DISP[:WIND]:MODE
DISP:TEXT:CLE
LXI:IDEN[:STAT]
LXI:MDNS[:STAT]
LXI:MDNS:ENAB
LXI:MDNS:HNAM[:RES]?
LXI:MDNS:SNAM:DES
LXI:MDNS:SNAM[:RES]?
LXI:MDNS:RES
LXI:MDNS:REST
MEAS | FETC:ARR:CURR:HARM[:AMPL]?
MEAS | FETC:ARR:CURR:HARM:PHAS?
MEAS | FETC:ARR:CURR:NEUT[:DC]?
MEAS | FETC:ARR:CURR:NEUT:HARM[:AMPL]?
MEAS | FETC:ARR:CURR:NEUT:HARM:PHAS?
MEAS | FETC:ARR:VOLT:HARM[:AMPL]?
MEAS | FETC:ARR:VOLT:HARM:PHAS?
MEAS | FETC[:SCAL]:CURR:ACDC?
MEAS | FETC[:SCAL]:CURR:AMPL:MAX?
MEAS | FETC[:SCAL]:CURR:CRES?
MEAS | FETC[:SCAL]:CURR:HARM[:AMPL]?
MEAS | FETC[:SCAL]:CURR:HARM:PHAS?
MEAS | FETC[:SCAL]:CURR:HARM:THD?
MEAS | FETC[:SCAL]:FREQ?
MEAS | FETC[:SCAL]:POW:AC:APP?
MEAS | FETC[:SCAL]:POW:AC:REAC?
MEAS | FETC[:SCAL]:POW:AC:PFAC?

MEAS | FETC[:SCAL]:TVOL?
MEAS | FETC[:SCAL]:TVOL:ABS?
MEAS | FETC[:SCAL]:VOLT:ACDC?
MEAS | FETC[:SCAL]:VOLT:HARM[:AMPL]?
MEAS | FETC[:SCAL]:VOLT:HARM:PHAS?
MEAS | FETC[:SCAL]:VOLT:HARM:THD?
MEAS | FETC[:SCAL]:VOLT:RANK?
OUTP:DFI[:STAT]
OUTP:DFI:SOUR
OUTP:IMP:REAL
OUTP:IMP:REAC
OUTP:PON:STAT
OUTP:RI:MODE
OUTP:TTLT:SOUR
SENS:FREQ:SOURC
SENS:NSEL
[SOUR:]CURR:PEAK[:IMM]
[SOUR:]CURR:PEAK:MODE
[SOUR:]CURR:PEAK:TRIG
[SOUR:]FREQ:SLEW[:IMM]
[SOUR:]FREQ:SLEW:MODE
[SOUR:]FREQ:SLEW:TRIG
[SOUR:]FREQ:TRIG
[SOUR:]FM[:STAT]
[SOUR:]FM[:STAT]:DEV
[SOUR:]FM[:STAT]:FREQ
[SOUR:]FUNC[:SHAP][:IMM]
[SOUR:]FUNC[:SHAP]:MODE
[SOUR:]FUNC[:SHAP]:TRIG
[SOUR:]FUNC[:SHAP]:CSIN
[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:VOLT:SLEW
[SOUR:]LIST:VOLT:SLEW:POIN?
[SOUR:]LIST:VOLT:OFFS
[SOUR:]LIST:VOLT:OFFS:POIN?
[SOUR:]LIST:VOLT:OFFS:SLEW
[SOUR:]LIST:VOLT:OFFS:SLEW:POIN?
[SOUR:]PHAS:AUX[:ADJ | :IMM]
[SOUR:]PHAS:MODE
[SOUR:]PHAS:TRIG
[SOUR:]VOLT:AUX[:LEV][:IMM][:AMP]
[SOUR:]VOLT:AUX:TRIG[:AMP]
[SOUR:]VOLT:OFFS[:IMM]
[SOUR:]VOLT:OFFS:MODE
[SOUR:]VOLT:OFFS:TRIG
[SOUR:]VOLT:SLEW:[:IMM]
[SOUR:]VOLT:SLEW:MODE
[SOUR:]VOLT:SLEW:TRIG

[SOUR:]VOLT:PROT[:LEV]
[SOUR:]VOLT:PROT:STAT
[SOUR:]VOLT:SENS | ALC:DET
[SOUR:]VOLT:SLEW[:IMM]
[SOUR:]VOLT:SLEW:MODE
[SOUR:]VOLT:SLEW:TRIG
SEC[:IMM]
SYST:COMM:LAN:CONT?
SYST:COMM:LAN:DHCP
SYST:COMM:LAN:DNS[{1 | 2}]
SYST:COMM:LAN:DOM?
SYST:COMM:LAN:GAT
SYST:COMM:LAN:IPAD
SYST:COMM:LAN:MAC?
SYST:COMM:LAN:SMAS
SYST:COMM:LAN:TELN:PROM
SYST:COMM:LAN:TELN:WMES
SYST:COMM:LAN:UPD
SYST:COMM:RLST
SYST:COMM:TCP:CONT?
SYST:COMM:PERS:MAN
SYST:COMM:PERS:MAN:DEF
SYST:COMM:PERS:MOD
SYST:COMM:PERS:MOD:DEF
SYST:SEC:COUN?
SYST:SEC:[IMM]
SYST:SET
SYST:LOC
SYST:REM
SYST:RWL
TRIG:SEQ2 | SYNC:PHAS
TRIG:SEQ1:DEF TRAN
TRIG:SEQ2:DEF SYNC
TRIG:SEQ3:DEF ACQ
*LRN

C Error Messages

This appendix provides a list of error messages.

Error Number List

This appendix gives the error numbers and descriptions that are returned by the ac 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.

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. When errors occur, the Standard Event Status register records them in bit 2, 3, 4, or 5:

Table C-1 Error numbers

Error #	Error string [description/explanation/examples]
	Command errors –100 through –199 (sets Standard Event Status Register bit #5)
–100	Command error [generic]
–101	Invalid character
–102	Syntax error [unrecognized command or data type]
–103	Invalid separator
–104	Data type error [e.g., “numeric or string expected, got block data”]
–105	GET not allowed
–108	Parameter not allowed [too many parameters]
–109	Missing parameter [too few parameters]
–112	Program mnemonic too long [maximum 12 characters]
–113	Undefined header [operation not allowed for this device]
–121	Invalid character in number [includes “9” in octal data, etc.]
–123	Numeric overflow [exponent too large; exponent magnitude >32 k]
–124	Too many digits [number too long; more than 255 digits received]
–128	Numeric data not allowed

Table C-1 Error numbers (continued)

Error #	Error string [description/explanation/examples]
–131	Invalid suffix [unrecognized units, or units not appropriate]
–138	Suffix not allowed
–141	Invalid character data [bad character, or unrecognized]
–144	Character data too long
–148	Character data not allowed
–150	String data error
–151	Invalid string data [e.g., END received before close quote]
–158	String data not allowed
–160	Block data error
–161	Invalid block data [e.g., END received before length satisfied]
–168	Block data not allowed
–170	Expression error
–171	Invalid expression
–178	Expression data not allowed
	Execution errors –200 through –299 (sets Standard Event Status Register bit #4)
–200	Execution error [generic]
–221	Settings conflict [check current device state]
–222	Data out of range [e.g., too large for this device]
–223	Too much data [out of memory; block, string, or expression too long]
–224	Illegal parameter value [device-specific]
–225	Out of memory
–270	Macro error
–272	Macro execution error
–273	Illegal macro label
–276	Macro recursion error
–277	Macro redefinition not allowed

Table C-1 Error numbers (continued)

Error #	Error string [description/explanation/examples]
	System errors –300 through –399 (sets Standard Event Status Register bit #3)
–310	System error [generic]
–350	Too many errors [errors beyond 9 lost due to queue overflow]
	Query errors –400 through –499 (sets Standard Event Status Register bit #2)
–400	Query error [generic]
–410	Query INTERRUPTED [query followed by DAB or GET before response complete]
–420	Query UNTERMINATED [addressed to talk, incomplete programming message received]
–430	Query DEADLOCKED [too many queries in command string]
–440	Query UNTERMINATED [after indefinite response]
	Self-test errors 0 through 99 (sets Standard Event Status Register bit #3)
0	No error
1	Non-volatile RAM RD0 section checksum failed
2	Non-volatile RAM CONFIG section checksum failed
3	Non-volatile RAM CAL section checksum failed
4	Non-volatile RAM WAVEFORM section checksum failed
5	Non-volatile RAM STATE section checksum failed
6	Non-volatile RAM LIST section checksum failed
7	Non-volatile RAM RST section checksum failed
10	RAM self-test
11 - 31	DAC self-test error, expected <n>, read <reading>
40	Voltage self-test error, output 1
41	Voltage self-test error, output 2
42	Voltage self-test error, output 3
43	Current self-test error, output 1
44	Current self-test error, output 2
45	Current self-test error, output 3

Table C-1 Error numbers (continued)

Error #	Error string [description/explanation/examples]
60	FPGA JTAG error
61	FPGA image invalid
62	Model detection FPGA version invalid
63	Analog voltage rail failed
64	Model detection FPGA model bit invalid
65	Model detection EEPROM model checksum failed
66	Model detection between FPGA and EEPROM incompatible
67	EEPROM version is higher than expected
70	Fan voltage failure
71	3.3 V voltage failed
72	5 V voltage failed
80	Digital I/O self-test error
90	Non-volatile RAM IO DATA section checksum failed
91	Non-volatile RAM ROM VERS section checksum failed
92	Non-volatile RAM SERIAL section checksum failed
93	Non-volatile RAM MAC ADDR section checksum failed
94	Non-volatile RAM LAN section checksum failed
95	Non-volatile RAM HTTP section checksum failed
96	Non-volatile RAM PERSONA section checksum failed
97	Non-volatile RAM CAL INFO section checksum failed

Table C-1 Error numbers (continued)

Error #	Error string [description/explanation/examples]
	Device-dependent errors 100 through 32767 (sets Standard Event Status Register bit #3)
200	Outgrd not responding
201	Front panel not responding
210	Ingrd receiver framing error
211	Ingrd uart overrun status
212	Ingrd received bad token
213	Ingrd receiver buffer overrun
214	Ingrd input buffer overrun
215	Outgrd output buffer overrun
216	RS-232 receiver framing error
217	RS-232 receiver parity error
218	RS-232 receiver overrun error
219	Ingrd inbuf count sync error
220	Front panel uart overrun
221	Front panel uart framing
222	Front panel uart parity
223	Front panel buffer overrun
224	Front panel timeout
401	CAL switch prevents calibration
402	CAL password is incorrect
403	CAL not enabled
404	Computed read back cal constants are incorrect
405	Computed programming cal constants are incorrect
406	Incorrect sequence of calibration commands
481	Model bit detection failed
501	LAN disconnect

Table C-1 Error numbers (continued)

Error #	Error string [description/explanation/examples]
502	LAN invalid IP address
503	LAN duplicate IP address
504	LAN failed to renew DHCP lease
505	LAN failed to configure
506	LAN failed to initialize
507	LAN VXI-11 fault
600	Systems in MODE:LIST have different list lengths
601	Requested voltage and waveform exceeds peak voltage capability
602	Requested voltage and waveform exceeds transformer volt-second rating
603	Command only applies to RS-232 interface
604	Trigger received before requested number of pre-trigger readings
605	Requested rms current too high for voltage range
606	Waveform data not defined
607	VOLT, VOLT:SLEW, and FUNC:SHAP modes incompatible
608	Measurement over-range
609	Output buffer overrun
610	Command cannot be given with present SYST:CONF setting
614	Measurement error occurred
900	Firmware update failed

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Index

R

S

T

U

V

W

This information is subject to change without notice. Always refer to the Keysight website for the latest revision.

Edition 1, October 2, 2017

Printed in Malaysia

06811-90005

www.keysight.com