



**Revision D**

**September 1999**

Copyright © 1996,1997,1999

California Instruments

All rights reserved

**P/N 5001-969**

**iL-Series AC Power  
Source / Analyzers  
SCPI Programming Manual**

Programming Manual

Refers to iL Series AC Power Source/Analyzers

Models: 3000iL, 4500iL, 4801iL

Copyright 1996,1997, 1999 California Instruments Company

September 1999

**Safety Summary**

The beginning of the AC source User's Manual has a Safety Summary page. Be sure you are familiar with the information on this page before programming the AC source from a controller.

---



***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.***

---

---

## Table of Contents

---

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>1-1</b>
1.1	DOCUMENTATION SUMMARY.....	1-1
1.1.1	External References.....	1-1
1.2	INTRODUCTION TO PROGRAMMING.....	1-2
1.2.1	IEEE-488 Capabilities of the AC source.....	1-2
1.2.2	IEEE-488 Address.....	1-2
<b>2.</b>	<b>INTRODUCTION TO SCPI.....</b>	<b>2-1</b>
2.1	CONVENTIONS USED IN THIS MANUAL.....	2-1
2.2	THE SCPI COMMANDS AND MESSAGES.....	2-1
2.2.1	Types of SCPI Commands.....	2-1
2.2.2	Types of SCPI Messages.....	2-1
2.2.3	The SCPI Command Tree.....	2-2
2.3	USING QUERIES.....	2-4
2.4	COUPLED COMMANDS.....	2-4
2.5	STRUCTURE OF A SCPI MESSAGE.....	2-4
2.5.1	The Message Unit.....	2-4
2.5.2	Combining Message Units.....	2-5
2.5.3	Headers.....	2-5
2.5.4	Query Indicator.....	2-6
2.5.5	Message Unit Separator.....	2-6
2.5.6	Root Specifier.....	2-6
2.5.7	Message Terminator.....	2-6
2.6	SCPI DATA FORMATS.....	2-7
2.6.1	Numerical Data Formats.....	2-7
2.6.2	Character Data.....	2-7
<b>3.</b>	<b>SYSTEM CONSIDERATIONS.....</b>	<b>3-1</b>
3.1	ASSIGNING THE IEEE-488 ADDRESS IN PROGRAMS.....	3-1
3.2	TYPES OF DOS DRIVERS.....	3-1
3.2.1	HP 82335A Driver.....	3-1
3.2.2	National Instruments GP-IB Driver.....	3-2
<b>4.</b>	<b>SCPI COMMAND REFERENCE.....</b>	<b>4-1</b>
4.1	INTRODUCTION.....	4-1
4.2	SUBSYSTEM COMMANDS.....	4-2
4.2.1	Calibration Subsystem.....	4-3
4.2.2	Diagnostic Subsystem.....	4-7
4.2.3	Display Subsystem.....	4-10
4.2.4	Instrument Subsystem.....	4-12
4.2.5	Arrays Measurement Subsystem.....	4-14
4.2.6	Current Measurement Subsystem.....	4-19
4.2.7	Frequency Measurement Subsystem.....	4-25
4.2.8	Power Measurement Subsystem.....	4-26
4.2.9	Voltage Measurement Subsystem.....	4-29
4.2.10	Output Subsystem.....	4-32
4.2.11	Sense Subsystem - Sweep.....	4-37
4.2.12	Source Subsystem - Current.....	4-40
4.2.13	Source Subsystem - Frequency.....	4-42
4.2.14	Source Subsystem - Function.....	4-46

4.2.15	Source Subsystem - List.....	4-49
4.2.16	Source Subsystem - Phase.....	4-57
4.2.17	Source Subsystem - Pulse.....	4-59
4.2.18	Source Subsystem - Voltage.....	4-62
4.2.19	Status Subsystem Commands.....	4-69
4.3	SYSTEM COMMANDS.....	4-78
4.3.1	Trace Subsystem Commands.....	4-81
4.3.2	Trigger Subsystem.....	4-83
4.4	COMMON COMMANDS.....	4-90
4.5	IEC 1000-2 COMMANDS.....	4-99
<b>5.</b>	<b>PROGRAMMING EXAMPLES .....</b>	<b>5-1</b>
5.1	INTRODUCTION.....	5-1
5.2	PROGRAMMING THE OUTPUT .....	5-2
5.2.1	Waveform Shapes.....	5-3
5.2.2	Individual Phases.....	5-5
5.2.3	Current Limit.....	5-6
5.3	COUPLED COMMANDS.....	5-7
5.4	PROGRAMMING OUTPUT TRANSIENTS.....	5-8
5.4.1	Transient System Model.....	5-9
5.5	STEP AND PULSE TRANSIENTS.....	5-10
5.6	LIST TRANSIENTS.....	5-11
5.7	TRIGGERING OUTPUT CHANGES .....	5-13
5.7.1	SCPI Triggering Nomenclature.....	5-13
5.7.2	Output Trigger System Model.....	5-14
5.7.3	Initiating the Output Trigger System.....	5-15
5.7.4	Selecting the Output Trigger Source.....	5-15
5.7.5	Specifying a Trigger Delay.....	5-16
5.7.6	Synchronizing Output Changes to a Reference Phase Angle.....	5-16
5.7.7	Generating Triggers .....	5-16
5.7.8	Specifying a Dwell Time for Each List Point.....	5-17
5.8	MAKING MEASUREMENTS.....	5-18
5.8.1	Voltage and Current Measurements.....	5-18
5.8.2	Power Measurements.....	5-19
5.8.3	Harmonic Measurements .....	5-19
5.8.4	Simultaneous Output Phase Measurements.....	5-20
5.8.5	Returning Voltage and Current Data From the Data Buffer .....	5-20
5.8.6	Triggering Measurements.....	5-20
5.8.7	SCPI Measurement Triggering Nomenclature .....	5-21
5.8.8	Measurement Trigger System Model .....	5-21
5.8.9	Initiating the Measurement Trigger System.....	5-21
5.8.10	Selecting the Measurement Trigger Source.....	5-22
5.8.11	Generating Measurement Triggers.....	5-22
5.8.12	Using the DFI Output to Indicate Error Conditions .....	5-22
5.9	CONTROLLING THE INSTANTANEOUS VOLTAGE AND CURRENT DATA BUFFERS.....	5-23
5.9.1	Varying the Voltage and Current Sampling Rate.....	5-23
5.9.2	Pre-event and Post-event Triggering.....	5-24
5.9.3	Regulatory-Compliant Measurement of Quasi-Stationary Harmonics.....	5-24
<b>6.</b>	<b>PROGRAMMING THE STATUS AND EVENT REGISTERS.....</b>	<b>6-1</b>
6.1	POWER-ON CONDITIONS.....	6-1
6.2	OPERATION STATUS GROUP .....	6-1
6.3	QUESTIONABLE STATUS GROUP .....	6-4
6.4	QUESTIONABLE INSTRUMENT ISUMMARY STATUS GROUP .....	6-5

6.5	STANDARD EVENT STATUS GROUP .....	6-6
6.6	STATUS BYTE REGISTER .....	6-7
6.7	EXAMPLES .....	6-8
6.7.1	<i>Determining the Cause of a Service Interrupt</i> .....	6-8
6.7.2	<i>Servicing Questionable Status Events</i> .....	6-8
6.7.3	<i>Monitoring Both Phases of a Status Transition</i> .....	6-9
6.7.4	<i>Programming the Trigger In and Trigger Out BNC Connectors</i> .....	6-9
6.8	REMOTE INHIBIT AND DISCRETE FAULT INDICATOR .....	6-11
6.8.1	<i>Remote Inhibit (RI)</i> .....	6-11
6.8.2	<i>Discrete Fault Indicator (DFI)</i> .....	6-11
6.9	SCPI COMMAND COMPLETION .....	6-12
<b>APPENDIX A: SCPI COMMAND TREE</b> .....		<b>A-1</b>
<b>APPENDIX B: SCPI CONFORMANCE INFORMATION</b> .....		<b>B-1</b>
<b>APPENDIX C: ERROR MESSAGES</b> .....		<b>C-1</b>

---

## Table of Figures

FIGURE 2-1 : PARTIAL COMMAND TREE .....	2-2
FIGURE 2-2 : COMMAND MESSAGE STRUCTURE .....	2-5
FIGURE 5-1 : MODEL OF TRANSIENT SYSTEM .....	5-9
FIGURE 5-2 : MODEL OF OUTPUT TRIGGERS .....	5-14
FIGURE 5-3 : MODEL OF MEASUREMENT TRIGGERS .....	5-21
FIGURE 5-4 : PRE-EVENT AND POST-EVENT TRIGGERING .....	5-24
FIGURE 6-1 : AC SOURCE STATUS MODEL .....	6-2
FIGURE 6-2 : BNC CONNECTOR TRIGGER MODEL .....	6-10

## Table of Tables

TABLE 2-1 : COMMAND PARAMETERS SUFFIXES AND MULTIPLIERS .....	2-7
TABLE 4-1 : PULSE:HOLD = WIDTH PARAMETERS .....	4-60
TABLE 4-2 : PULSE:HOLD = DCYCLE PARAMETERS .....	4-60
TABLE 4-3 : BIT CONFIGURATION OF STATUS OPERATION REGISTERS .....	4-70
TABLE 4-4 : BIT CONFIGURATION OF QUESTIONABLE REGISTERS .....	4-72
TABLE 4-5 : BIT CONFIGURATION OF QUESTIONABLE INSTRUMENT SUMMARY REGISTERS .....	4-74
TABLE 4-6 : BIT CONFIGURATION OF STANDARD EVENT STATUS ENABLE REGISTER .....	4-91
TABLE 4-7 : FACTORY-DEFINED *RST STATES .....	4-95
TABLE 4-8 : BIT CONFIGURATION OF STATUS BYTE REGISTER .....	4-97
TABLE 6-1 : OPERATION STATUS REGISTERS .....	6-1
TABLE 6-2 : BIT CONFIGURATIONS OF STATUS REGISTERS .....	6-3
TABLE 6-3 : QUESTIONABLE STATUS REGISTERS .....	6-4
TABLE 6-4 : QUESTIONABLE INSTRUMENT ISUMMARY STATUS REGISTERS .....	6-5

## 1. Introduction

---

This manual contains programming information for the iL Series AC source/Analyzers. The expression "AC source" as used in the manual also applies to the same series. You will find the following information in the rest of this manual:

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

### 1.1 Documentation Summary

---

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

- User's Manual. Includes specifications and supplemental characteristics, how to use the front panel, how to connect to the instrument, and calibration procedures.

#### 1.1.1 External References

---

##### SCPI References

The following documents will assist you with programming in SCPI:

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

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

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

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

## **1.2 Introduction to Programming**

---

This section provides some general information regarding programming IEEE-488 bus instrumentation.

### **1.2.1 IEEE-488 Capabilities of the AC source**

---

All AC source functions except for setting the IEEE-488 address are programmable over the IEEE-488. The IEEE 488.2 capabilities of the AC source are listed in Chapter 5 of the User's Manual.

### **1.2.2 IEEE-488 Address**

---

The AC source operates from an IEEE-488 address that is set from the front panel. To set the IEEE-488 address, press the Address key on the front panel and enter the address using the Entry keys. Press the ENTER key to confirm your selection.



## 2. Introduction to SCPI

---

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over the IEEE-488. SCPI is layered on top of the hardware-portion of IEEE 488.2. The same SCPI commands and parameters control the same functions in different classes of instruments. For example, you would use the same DISPlay command to control the AC source display and the display of a SCPI-compatible multimeter.

### 2.1 Conventions Used in This Manual

---

Angle brackets	<>	Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.
Vertical bar		Vertical bars separate alternative parameters. For example, NORM   TEXT indicates that either "TEXT" or "NORM" can be used as a parameter.
Square Brackets	[]	Items within square brackets are optional. The representation [SOURce:]LIST means that SOURce: may be omitted.
Braces	{}	Braces indicate parameters that may be repeated zero or more times. It is used especially for showing arrays. The notation <A> <,B> shows that parameter "A" must be entered, while parameter "B" may be omitted or may be entered one or more times.

### 2.2 The SCPI Commands and Messages

---

#### 2.2.1 Types of SCPI Commands

---

SCPI has two types of commands, common and subsystem.

- Common commands generally are not related to specific operation but to controlling overall AC source functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: \*RST, \*IDN?, \*SRE 8
- Subsystem commands perform specific AC source functions. They are organized into an inverted tree structure with the "root" at the top. Some are single commands while others are grouped within specific subsystems.

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

#### 2.2.2 Types of SCPI Messages

---

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

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

### 2.2.3 The SCPI Command Tree

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

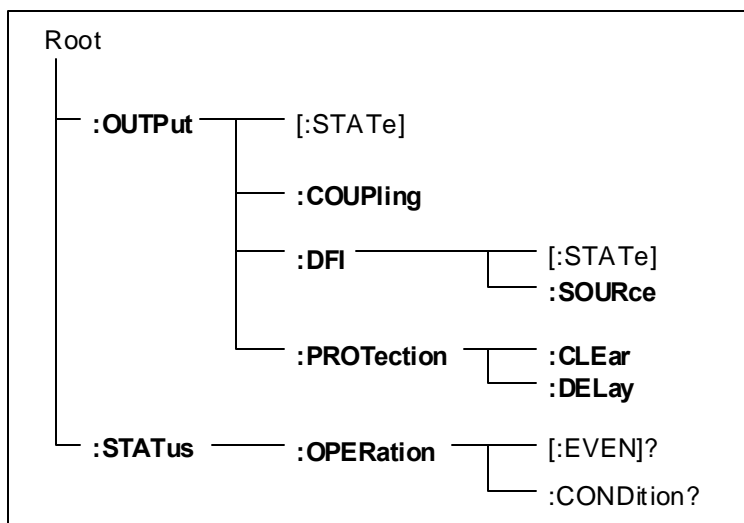


Figure 2-1 : Partial Command Tree

#### The Root Level

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

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

#### Active Header Path

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

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

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

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

1. OUTPUT:PROTECTION:CLEAR and
2. OUTPUT:PROTECTION:DELAY 20.

The entire message would be:

```
OUTPUT:PROTECTION:CLEAR;DELAY 20
```

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

### The Effect of Optional Headers

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

```
OUTPUT:STATE OFF;PROTECTION:CLEAR
```

in one message. If you tried to send

```
OUTPUT OFF;PROTECTION:CLEAR
```

the header path would return to :OUTPUT instead of :PROTECTION.

The optional header [SOURCE] precedes the current, frequency, function, phase, pulse, list, and voltage subsystems. This effectively makes :CURRENT,:FREQUENCY, :FUNCTION, :PHASE, :PULSE, :LIST, and :VOLTAGE root-level commands.

### Moving Among Subsystems

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

```
OUTPUT:PROTECTION:CLEAR
```

```
STATUS:OPERATION:CONDITION?
```

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

```
OUTPUT:PROTECTION:CLEAR::STATUS:OPERATION:CONDITION?
```

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

```
VOLTAGE:LEVEL 70;PROTECTION 80::CURRENT:LEVEL 3;PROTECTION:STATE ON
```

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

---

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

---

### Including Common Commands

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

```
VOLTAGE:TRIGGER 7.5;INITIALIZE;*TRG  
OUTPUT OFF;*RCL 2;OUTPUT ON
```

## 2.3 Using Queries

---

Observe the following precautions with queries:

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

## 2.4 Coupled Commands

---

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

- the voltage and function shape commands
- the step, pulse, and list commands that control output voltages and function shapes
- the pulse commands that program the width, duty cycle, period, and the hold parameter
- the voltage range and current limit commands

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

## 2.5 Structure of a SCPI Message

---

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

### 2.5.1 The Message Unit

---

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

ABORt<newline>

VOLTage?<newline>

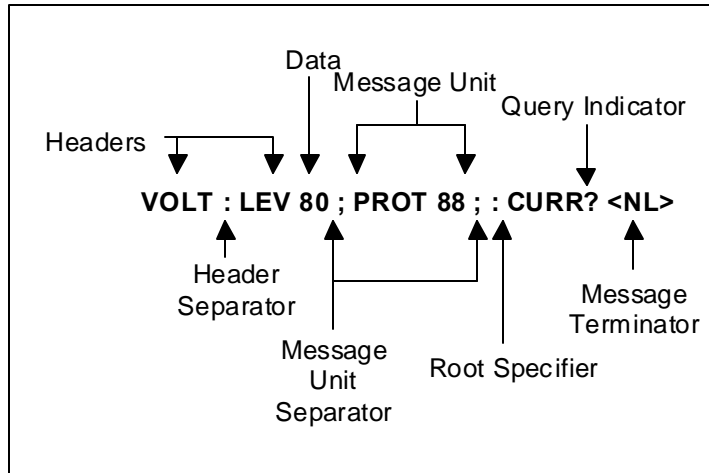
The message unit may include a parameter after the header. The parameter usually is numeric, but it can be a string:

VOLTage 20<newline>

VOLTage MAX<newline>

## 2.5.2 Combining Message Units

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



**Figure 2-2 : Command Message Structure**

The basic parts of the above message are:

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

## 2.5.3 Headers

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

Long Form	The header is completely spelled out, such as VOLTAGE, STATUS, and DELAY.
Short Form	The header has only the first three or four letters, such as VOLT, STAT, and DEL.

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

**Header Convention**

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

**Header Separator**

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

**Optional Headers**

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

---

**2.5.4 Query Indicator**

---

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

---

**2.5.5 Message Unit Separator**

---

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

---

**2.5.6 Root Specifier**

---

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

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

---

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

---

---

**2.5.7 Message Terminator**

---

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

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

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

## 2.6 SCPI Data Formats

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

### 2.6.1 Numerical Data Formats

Symbol	Data Form
<b>Talking Formats</b>	
<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 ON   OFF
<b>Listening Formats</b>	
<Nrf>	Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273 273. 2.73E2
<Nrf+>	Expanded decimal format that includes <Nrf> and MIN MAX. Examples: 273 273. 2.73E2 MAX. MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter.
<Bool>	Boolean Data. Example: 0   1

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

Table 2-1 : Command parameters Suffixes and Multipliers

### 2.6.2 Character Data

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

- <CRD> Character Response Data. Permits the return of character strings.
- <AARD> Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
- <SRD> String Response Data. Returns string parameters enclosed in double quotes.





### 3. System Considerations

---

This chapter addresses some system issues concerning programming. Specifically, these are AC source addressing and the use of the following IEEE-488 system interfaces:

- PC controller with National Instruments GPIB-PCII Interface/Handler
- HP82335A PC Interface controller card

#### 3.1 Assigning the IEEE-488 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 IEEE-488 select code is 7, and the AC source will be assigned to the variable ACS.

```
1070 ACS 706                !HP82335A Interface
1070 ASSIGN @ACS TO 706     !HP 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 (see the National Instruments GP-IB documentation).

#### 3.2 Types of DOS Drivers

---

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

##### 3.2.1 HP 82335A Driver

---

For GW-BASIC programming, the IEEE-488 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 IEEE-488 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.

### 3.2.2 National Instruments GP-IB 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 IEEE-488 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 power source expects a message termination on EOI or line feed, so set EOI w/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.**

---

## 4. SCPI Command Reference

---

### 4.1 Introduction

---

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

#### **Phases**

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

#### **Related Commands**

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

This chapter is organized as follows:

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

## 4.2 Subsystem Commands

---

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

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

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

## 4.2.1 Calibration Subsystem

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 nonvolatile memory.

### Subsystem Syntax

#### CALibrate

:CURRent	
:AC	Begin ac current programming calibration sequence
:MEASure	Begin current measurement calibration sequence
:DATA <n>	Input a calibration measurement
:IMPedance	Begin output impedance calibration sequence
:LEVel <level>	Advance to next calibration step (P1 P2 P3 P4)
:PASSword <n>	Set calibration password
:SAVE	Save new cal constants in non-volatile memory
:STATE <bool> [,<n>]	Enable or disable calibration mode
:VOLTage	
:AC	Begin ac voltage calibration sequence
:PROTection	Begin voltage protection calibration sequence
:RTIME	Begin realtime voltage calibration sequence

#### **CALibrate:CURRent:AC**

##### Phase Selectable

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		
Parameters	None		
Examples	CAL:CURR:AC		
Related Commands	CAL:STAT	CAL:SAVE	CAL:LEV

#### **CALibrate:CURRent:MEASure**

##### 4801iL Only

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

Command Syntax	CALibrate:CURRent:MEASure		
Parameters	None		
Examples	CAL:CURR:MEAS		
Related Commands	CAL:STAT	CAL:SAVE	CAL:LEV

**CALibrate:DATA**

Phase Selectable

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 nonvolatile 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>	
Parameters	<external reading>	
Unit	A (amperes)	
Examples	CAL:DATA 3222.3 MA	CAL:DATA 5.000
Related Commands	CAL:SAVE	CAL:STAT

**CALibrate:IMPedance**

4801iL Only

This command can only be used in calibration mode. It calibrates the output impedance circuits. The AC source automatically performs the calibration and stores the impedance constant in nonvolatile memory. CALibrate:IMPedance is a sequential command that takes several seconds to complete.

Command Syntax	CALibrate:IMPedance	
Parameters	None	
Examples	CAL:IMP	
Related Commands	CAL:SAVE	CAL:STAT

**CALibrate:LEVel**

Phase Selectable

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>	
Parameters	P1   P2   P3   P4	
Examples	CAL:LEV P2	
Related Commands	CAL:STAT	CAL:SAVE

**CALibrate: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 nonvolatile 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:PASScode<NRf>
Parameters	<model number> (default)
Examples	CAL:PASS 4500          CAL:PASS 06.1994
Related Commands	CAL:STAT

**CALibrate:SAVE**

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

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

**CALibrate: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>]
Parameters	0 1 OFF ON[,<password>]
*RST Value	OFF
Examples	CAL:STAT 1,4500          CAL:STAT OFF
Query Syntax	CALibrate:STATe?
Returned Parameters	<NR1>
Related Commands	CAL:PASS          CAL:SAVE

**CALibrate:VOLTage:AC**

Phase Selectable

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
Parameters	None
Examples	CAL:VOLT:AC
Related Commands	CAL:SAVE    CAL:STAT

**CALibrate:VOLTage:PROTection**

This command can only be used in calibration mode. It calibrates the overvoltage protection (OV) circuit. The AC source automatically performs the calibration and stores the new OV constant in nonvolatile memory. CALibrate:VOLTage:PROTection is a sequential command that takes several seconds to complete.

Command Syntax:	CALibrate:VOLTage:PROTection
Parameters	None
Example	CAL:VOLT:PROT
Related Commands	CAL:STAT    CAL:SAVE

**CALibrate:VOLTage:RTIME**

4801iL Only

This command can only be used in calibration mode. It calibrates the realtime voltage programming circuit.

Command Syntax	CALibrate:VOLTage:RTIME
Parameters	None
Example	CAL:VOLT:RTIM
Related Commands	CAL:STAT    CAL:SAVE



## 4.2.2 Diagnostic Subsystem

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

### Subsystem Syntax

DIAGnostic	
:CURRent?	Returns low resolution rms current
:EEPROM<n>,<n>,<n>[,INTeger   FLOat]	Write address, size, value, format
:FAN	
:MODE AUTO   MANual	Sets fan speed mode
:SPEed	
:AUTO? Returns auto fan speed	
:MANUAL <n>	Sets manual fan speed
:NOP	Null execution function
:RESet	Force power-on reset
:Rwlongform <n>	Set value of dummy function
:TEMPerature	
:AMBient?	Returns ambient temperature in °C

### DIAGnostic:CURRent?

#### Phase Selectable

This query returns the current measured by the detector used in the rms current limit circuit. The rms converter output is monitored with an 8-bit DAC.

Query Syntax	DIAGnostic:CURRent?
Returned Parameters	<NR3>
Examples	DIAG:CURR?
Related Commands	None

### DIAGnostic:EEPROM

This command sets or reads the values in nonvolatile memory. The first argument is the nonvolatile memory address of the data, the second argument is the size of the data in units of basic machine storage, and the third argument is the value. The third argument is not used in the query. The only allowable value for the size argument is 1. The optional INTeger or FLOat argument describes the internal storage format of the number. INTeger is assumed if no format is specified.

---

**Note:** Access to this command is password protected. The instrument must be in calibration mode before this command can be given.

---

Command Syntax	DIAGnostic:EEPROM <NRf>,<NRf>,<NRf>[,INTeger   FLOat]
Parameters	<start_address>,<size>,<value>   INTeger   FLOat
Query Syntax	DIAGnostic:EEPROM? <NRf>,<NRf>[,INTeger   FLOat]
Returned Parameters	<NR3>
Examples	DIAG:EEPROM 24, 1, 1024, INT
Related Commands	CAL:STATE

**DIAGnostic:FAN:MODE**

This command sets or reads the mode of fan speed control. When AUTO is selected, the fan speed is a function of load current and internal temperature. When MANUAL is selected, the fan speed is set by the DIAGnostic:FAN:SPEED:MANUAL command.

Command Syntax	DIAGnostic:FAN:MODE <mode>
Parameters	AUTO   MANUAL
*RST Value	AUTO
Examples	DIAG:FAN:MODE AUTO
Query Syntax	DIAGnostic:FAN:MODE?
Returned Parameters	<CRD>
Related Commands	None

**DIAGnostic:FAN:SPEED:MANUAL**

This command sets the fan speed control when the fan mode is MANUAL. The fan speed is expressed as a percentage of its maximum speed. The range of values is 0 through 100.

---

**Note:** The AC Source requires sufficient air flow to prevent overheating. When manual fan speed is selected, it becomes the user's responsibility to set the fan speed high enough to prevent the unit from overheating.

---

Command Syntax	DIAGnostic:FAN:SPEED:MANUAL <NRf>
Parameters	0 to 100
Unit	% of maximum speed
*RST Value	50
Examples	DIAG:FAN:SPEED:MAN 85
Query Syntax	DIAGnostic:FAN:SPEED:MAN?
Returned Parameters	<NR3>
Related Commands	None

**DIAGnostic:FAN:SPEED:AUTO?**

This query returns the value of the automatic fan speed control. The fan speed is returned as a percentage of its maximum speed.

Query Syntax	DIAGnostic:FAN:SPEED:AUTO?
Returned Parameters	<NR3>
Unit	% of maximum speed
Examples	DIAG:FAN:SPEED:AUTO?
Related Commands	None

**DIAGnostic:NOP**

This command is a null function which does absolutely nothing. It is used for test purposes. The query form of this command always returns a zero.

Command Syntax	DIAGnostic:NOP
Parameters	None
Examples	DIAG:NOP
Query Syntax	DIAGnostic:NOP?
Returned Parameters	<NR1>

**DIAGnostic:RESet**

This commands forces a power-on reset.

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

**DIAGnostic:RWlongform**

This command stores its NRf argument and allows it to be read back using the query form. It is used to test the GPIB driver, parser and floating point numeric functions in the firmware. It has no effect on the output of the AC source.

Command Syntax	DIAGnostic:RWlongform <NRf>
Parameters	any NRf value
Examples	DIAG:RW 342
Query Syntax	DIAGnostic:RWlongform?
Returned Parameters	<NR3>

**DIAGnostic:TEMPerature:AMBient?**

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

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

### 4.2.3 Display Subsystem

This subsystem programs the front panel display of the iL Series.

#### Subsystem Syntax

##### DISPlay

[:WINDow]		
[:STATe] <bool>		Enable/disable front panel display
:MODE NORMAl   TEXT		Set display mode
:TEXT		
[:DATA] <display_string>		Set text displayed in text mode

#### DISPlay

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

Command Syntax	DISPlay[:WINDow]:STAT<bool>
Parameters	0 1 OFF ON
*RST Value	ON
Examples	DISP:STAT 1    DISP:STAT OFF
Query Syntax	DISPlay[:WINDow]:STAT?
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]<display_string>
Parameters	<display string>
*RST Value	null string
Examples	DISP:TEXT "DO TEST1"
Query Syntax	DISPlay[:WINDow]:MODE?
Returned Parameters	<SRD> (the last programmed string)
Related Commands	DISP DISP:TEXT

## 4.2.4 Instrument Subsystem

This subsystem programs the three-phase output capability of the 3000iL and the 4500iL.

### Subsystem Syntax

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

### INSTrument:COUPle

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

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

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

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

**INSTRument:NSElect****INSTRument:SElect**

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

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

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

## 4.2.5 Arrays Measurement Subsystem

---

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

Individual outputs of a three-phase source are specified by the setting of INSTRument:NSElect.

### Subsystem Syntax

MEASure | FETCh

:ARRay

:CURRent

[:DC]?

Returns the digitized instantaneous current

:HARMonic

[:AMPLitude]?

Returns amplitudes of the first 50 harmonics

:PHASe?

Returns phase angles of the first 50 harmonics

:NEUTral

[:DC]?

Returns the neutral digitized instantaneous current (3-phase only)

:HARMonic

[:AMPLitude]?

Returns neutral current harmonic amplitude

:PHASe?

Returns neutral current harmonic phase

:VOLTage

[:DC]?

Returns the digitized instantaneous voltage

:HARMonic

[:AMPLitude]?

Returns amplitudes of the first 50 harmonics

:PHASe?

Returns phase angles of the first 50 harmonics



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

Phase Selectable

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

The 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	INST:NSEL    SENS:SWE

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

Phase Selectable

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

The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

**MEASure:ARRay:CURRent:HARMonic:PHASe?**

**FETCh:ARRay:CURRent:HARMonic:PHASe?**

Phase Selectable

These queries return an array of harmonic phases of output current in degrees, referenced to the positive zero crossing of the fundamental component.

The first value returned is the dc component (always returned as 0 degrees phase) , the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

**MEASure:ARRay:CURRent:NEUTral?**

**FETCh:ARRay:CURRent:NEUTral?**

These queries return an array containing the instantaneous output current of the neutral output terminal in amperes.

The output voltage and current are digitized whenever a measure command is given or whenever an 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:NEUTral[:DC]? FETCh:ARRay:CURRent:NEUTral[:DC]?
Parameters	None
Examples	MEAS:ARR:CURR:NEUT?      FETC:ARR:CURR:NEUT?
Returned Parameters	4096 NR3 values
Related Commands	INST:NSEL      SENS:SWE

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

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

The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

These queries return an array of harmonic phases of output current of the neutral output terminal in degrees, referenced to the positive zero crossing of the fundamental component.

The first value returned is the dc component (always returned as 0 degrees phase), the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

Phase Selectable

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:SWEp:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSE:SWEp: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	INST:NSEL      SENS:SWE

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

Phase Selectable

These queries return an array of harmonic amplitudes of output voltage in rms volts.

The first value returned is the dc component, the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus, the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

## 4.2.6 Current Measurement Subsystem

This subsystem programs the current measurement capability of the 3000iL and the 4500iL. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new measurement data before returning a reading. FETCh returns a reading computed from previously acquired data.

Individual outputs of a three-phase source are specified by the setting of INSTRument:NSElect.

### Subsystem Syntax

#### MEASure | FETCh

[:SCALar]	
:CURRent	
[:DC]?	Returns dc component of the current
:AC?	Returns ac rms current
:ACDC?	Returns ac+dc rms current
:AMPLitude	
:MAX?	Returns peak current
:CREStfactor?	Returns current crestfactor
:HARMonic	
[:AMPLitude]? <n>	Returns amplitude of the Nth harmonic of current
:PHASe? <n>	Returns phase of the Nth harmonic of current
:THD?	Returns % of total harmonic distortion of current
:NEUTral	
[:DC]?	Returns neutral dc current (3-phase only)
:AC?	Returns neutral ac rms current (3-phase only)
:ACDC?	Returns neutral ac+dc rms current (3-phase only)
:HARMonic	
[:AMPLitude]? <n>	Returns neutral current harmonic amplitude
:PHASe? <n>	Returns neutral current harmonic phase

#### MEASure:CURRent?

#### FETCh:CURRent?

#### Phase Selectable

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

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

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

Phase Selectable

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

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

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

Phase Selectable

These queries return the ac and dc components of the rms current being sourced at the output terminals.

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

**MEASure:CURRent:AMPLitude:MAXimum?****FETCh:CURRent:AMPLitude:MAXimum?**

Phase Selectable

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	INST:NSEL

**MEASure:CURRent:CREStfactor?****FETCh:CURRent:CREStfactor?**

Phase Selectable

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:CRESt? FETC:CURR:CRESt?
Returned Parameters	<NR3>
Related Commands	INST:NSEL

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

Phase Selectable

These queries return the rms amplitude of the Nth harmonic of output current.

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

Phase Selectable

These queries return the phase angle of the Nth harmonic of output current, referenced to the positive zero crossing of the fundamental component.

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

Phase Selectable

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

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



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

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

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

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

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

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

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

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

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

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

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

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

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

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

## 4.2.7 Frequency Measurement Subsystem

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

Subsystem Syntax

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

**MEASure:FREQuency?**

**FETCh:FREQuency?**

This query returns the output frequency in Hertz.

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

## 4.2.8 Power Measurement Subsystem

---

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

Individual outputs of a three-phase source are specified by the setting of INSTRument:NSElect.

### Subsystem Syntax

```

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

### MEASure:POWer?

### FETCh:POWer?

### Phase Selectable

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

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

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

Phase Selectable

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

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

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

Phase Selectable

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

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

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

Phase Selectable

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

$$\text{VAR} = \sqrt{(\text{square}(\text{apparent power}) - \text{square}(\text{real power}))}$$

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

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

Phase Selectable

These queries return the output power factor. The power factor is computed as:  
 $\text{pfactor} = \text{real power}/\text{apparent power}$

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	INST:NSEL

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

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

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

## 4.2.9 Voltage Measurement Subsystem

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

Individual outputs of a three-phase source are specified by the setting of INSTRument:NSElect.

Subsystem Syntax

```
MEASure | FETCh
  [:SCALar]
    :VOLTage
      [:DC]?           Returns the 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
```

**MEASure:VOLTage?**

**FETCh:VOLTage?**

Phase Selectable

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

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

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

Phase Selectable

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

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

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

Phase Selectable

These queries return the ac or dc rms voltage being sourced at the output terminals.

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

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

Phase Selectable

These queries return the rms amplitude of the Nth harmonic of output voltage.

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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



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

Phase Selectable

These queries return the phase angle of the Nth harmonic of output voltage, referenced to the positive zero crossing of the fundamental component.

The parameter is the desired harmonic number. Queries sent with a value of 0 return the dc component. A value of 1 returns the fundamental output frequency. Harmonic orders can be queried up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. Thus the maximum harmonic that can be measured is dependent on the output frequency. Any harmonics that represent frequencies greater than 12.6kHz are returned as 0.

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

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

Phase Selectable

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

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

## 4.2.10 Output Subsystem

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

### Subsystem Syntax

#### OUTPut

[:STATe] <bool>	Enable/disable output voltage, current, power, etc.
:DFI	
[:STATE] <bool>	Enable/disable DFI output
:SOURce <source>	Selects an event source (QUES OPER ESB RQS OFF)
: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 RST   RCL0	Set power-on state to *RST or *RCL0
:PROtection	
:CLEar	Reset latched protection
:DELay <n>	Delay after programming/before protection
:RI	
:MODE <mode>	set remote inhibit input (LATC LIVE OFF)
:TTLTrg	
[: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.

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

**OUTPut:DFI**

This command enables or disables the discrete fault indicator (DFI) signal to the iL Series.

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

**OUTPut:DFI:SOURce**

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

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

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

**OUTPut:IMPedance**

4801iL Only

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 OFF
Query Syntax	OUTPut:IMP[:STATe]?
Returned Parameters	0   1
Related Commands	OUTP:IMP:REAL    OUTP:IMP:REAC

**OUTPut:IMP:REAL**

4801iL Only

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	OUTP:IMP:REAL<NRf>
Parameters	0 to 1 (ohms)
*RST Value	0 ohms
Examples	OUTP:IMP:REAL 0.25
Query Syntax	OUTPut:IMP:REAL?
Returned Parameters	<NR3>
Related Commands	OUTP:IMP      OUTP:IMP:REAC

**OUTPut:IMP:REActive**

4801iL Only

This command sets the reactive 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	OUTP:IMP:REActive<NRf>
Parameters	0.00002 to 0.001 (Henrys)
*RST Value	0.00005 Henrys
Examples	OUTP:IMP:REAC 100E-6
Query Syntax	OUTPut:IMP:REActive?
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), or remote inhibit (RI) fault condition is detected. All conditions that generated the fault must be removed before the latch can be cleared. The output is then restored to the state it was in before the fault condition occurred.

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

**OUTPut:PROTEction:DELay**

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

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

**OUTPut:RI:MODE**

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

LATChing	A TTL low at the RI input latches the output in the protection shutdown state, which can only be cleared by OUTPut:PROTEction:CLEar.
LIVE	The output state follows the state of the RI input. A TTL low at the RI input turns the output off; a TTL high turns the output on.
OFF	The instrument ignores the RI input.

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 iL Series units.

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

**OUTPut:TTLTrg:SOURce**

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

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

When an event becomes true at the selected TTLTrg source, a pulse is sent to the 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

## 4.2.11 Sense Subsystem - Sweep

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

### Subsystem Syntax

SENSE		
:CURRent		
:ACDC		
:RANGe		
[:UPPer]<n>		Sets measurement current range
: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] KBESsel   RECTangular		Sets measurement window type

### SENSE:CURRent:ACDC:RANGe

#### 4801iL Only

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

High Range:	0 through 57.1342 A <sub>rms</sub> ( -80.8 A <sub>peak</sub> through + 80.8 A <sub>peak</sub> )
Low Range:	0 through 5.71342 A <sub>rms</sub> ( -8.08 A <sub>peak</sub> through + 8.08 A <sub>peak</sub> )

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 SENSE: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<sub>rms</sub>.

Command Syntax	SENSE:CURRent:ACDC:RANGe[:UPPer]<NRf+>
Parameters	0 through 80.8   MINimum   MAXimum
*RST Value	MAX (high range)
Examples	SENS:CURR:ACDC:RANGE 10
Query Syntax	SENSE:CURR:ACDC:RANGe?
Returned Parameters	<NR3>
Related Commands	SENS:SWE:TINT      MEAS:ARR

**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
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.

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 (microseconds)
*RST Value	25.037 s
Examples	SENS:SWE:TINT 100
Query Syntax	SENSe:SWEep:TINTerval?
Returned Parameters	<NR3>
Related Commands	SENS:SWE:OFFS:POIN            MEAS:ARR



**SENSe:WINDow**

Phase Selectable

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]<window>	
Parameters	RECTangular or KBESsel	
*RST Value	KBESsel	
Examples	SENS:WIN KBE	
Query Syntax	SENSe:WINDow?	
Returned Parameters	<CRD>	
Related Commands	MEAS:ARR:CURR:HARM	MEAS:ARRAY:VOLT:HARM

## 4.2.12 Source Subsystem - Current

This subsystem programs the output current of the AC source.

### Subsystem Syntax

```
[SOURce:]
  CURRent
    [:LEVel]
      [:IMMEDIATE]
        [:AMPLitude] <n>      Sets the rms current limit
      :PROTection
        :STATe <bool>        Enable/Disable rms current limit protection
```

### CURRent

#### Phase Selectable

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 within 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.

Note that the CURRent command is coupled with the VOLTage:RANGe. This means that the maximum current limit that can be programmed at a given time depends on the voltage range setting in which the unit is presently operating. Refer to Section 5.3 under "Coupled Commands" for more information.

Command Syntax	[SOURce:]CURRent[:LEVel] [:IMMEDIATE][:AMPLitude]<NRf+>
Parameters	
Unit	A (rms amperes)
*RST	1
Examples	CURR 5 CURR:LEV .5
Query Syntax	[SOURce:]CURRent[:LEVel] [:IMMEDIATE][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	CURR:PROT:STAT    VOLT:RANG

**CURRent:PROTection:STATe**

This command enables or disables the AC source 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 (see Chapter 6). An overcurrent condition can be cleared with OUTPut:PROTection:CLEar after the cause of the condition is removed.

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 0 CURR:PROT:STAT OFF
Query Syntax	[SOURce:]CURRent:PROTection:STATe?
Returned Parameters	0   1
Related Commands	OUTP:PROT:CLE      OUTP:PROT:DEL

### 4.2.13 Source Subsystem - Frequency

This subsystem programs the output frequency of the AC source.

#### Subsystem Syntax

[SOURce:]		
FREQUency		
[:CW   :IMMEDIATE] <n>		Sets the frequency
:MODE <mode>		Sets frequency mode (FIX STEP PULS LIST)
:SLEW		
[:IMMEDIATE] <n>   INFINity		Sets the frequency slew rate
:MODE <mode>		Sets frequency slew mode (FIX STEP PULS LIST)
:TRIGgered <n>   INFINity		Sets the triggered frequency slew rate
:TRIGgered <n>		Sets the triggered frequency

#### FREQUency

This command sets the frequency of the output waveform.

Command Syntax	[SOURce:]FREQUency[:CW :IMMEDIATE]<NRf+>
Parameters	45 to 5000
Unit	HZ (Hertz)
*RST Value	60 Hz
Examples	FREQ 50
Query Syntax	[SOURce:]FREQUency?
Returned Parameters	<NR3>
Related Commands	FREQ:MODE FREQ:SLEW

**FREQuency:MODE**

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

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   INFinity
Unit	HZ (Hertz per second)
*RST Value	MAXimum
Examples	FREQ:SLEW:IMM 75 FREQ:SLEW MAX
Query Syntax	[SOURce:]FREQuency:SLEW?
Returned Parameters	<NR3>
Related Commands	FREQ:SLEW:MODE FREQ

**FREQuency:SLEW:MODE**

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

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

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

**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   INFINITY
Unit	HZ (Hertz per second)
*RST Value	MAXimum
Examples	FREQ:SLEW:TRIG 75 FREQ:SLEW:TRIG MAX
Query Syntax	[SOURce:]FREQuency:SLEW:TRIG?
Returned Parameters	<NR3>
Related Commands	FREQ:SLEW:MODE FREQ

**FREQuency:TRIGgered**

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

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

## 4.2.14 Source Subsystem - Function

This subsystem programs the output function of the AC source.

Subsystem Syntax

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

### FUNCTION

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

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

The maximum peak voltage that the AC source can output is 425 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 sinewave, the maximum voltage that can be programmed is 300 V rms.

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

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sinewave.

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



**FUNCTION:MODE**

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

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

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

**FUNCTION:TRIGgered**

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

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

The maximum peak voltage that the AC source can output is 425 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 sinewave, the maximum voltage that can be programmed is 300 V rms.

---

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sinewave.

---

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

**FUNCTION:CSINusoid**

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

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

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

## 4.2.15 Source Subsystem - List

This subsystem controls the generation of complex sequences of output changes with rapid, precise timing and synchronized with internal or external signals. Each subsystem command for which lists can be generated has an associated list of values that specify the output at each list step. LIST:COUNT determines how many times the AC source sequences through a list before that list is completed. LIST:DWELI 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 nonvolatile memory.

MODE commands such as VOLTage:MODE LIST are used to activate lists for specific functions (See . However, the LIST:DWELI command is active whenever any function is set to list mode. Therefore, LIST:DWELI 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
:DWELI <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
:SHAPE <shape>{,<shape>}	Sets the waveform shape list
:POINTs?	Returns the number of shape list points
:STEP ONCE   AUTO	Defines whether list is dwell- or trigger-paced
:TTLTrg <bool>{,<bool>}	Defines the output marker list
:POINTs?	Returns the number of output marker list points
: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

**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:CURRLIST:FREQ LIST:TTLT LIST:VOLT

**LIST:DWELI**

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

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

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

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

**LIST:DWEL:POINts?**

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

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

**LIST:FREQuency**

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

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

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

**LIST:FREQuency:POINts?**

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

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

**LIST:FREQUENCY:SLEW**

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

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

**LIST:FREQUENCY:SLEW:POINTS?**

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

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

**LIST:PHASe**

Phase Selectable

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

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

**LIST:PHASe:POINts?**

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

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

**LIST: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 transient is triggered. Changing list data while a subsystem is in list mode generates an implied ABORT. The following shapes may be specified:

SINusoid	A sinewave is output
SQUare	A squarewave is output
CSINusoid	The output is a clipped sine waveform. Both positive and negative peak amplitudes are clipped at a value determined by the SOURce:FUNcTION:SHAPe:CSINusoid setting.
<waveform_name>	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 and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

---

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sinewave.

---

Command Syntax	[SOURce:]LIST:SHAPe<shape>{,<shape>}
Parameters	SINusoid SQUare CSINusoid <waveform_name>
Examples	LIST:SHAP
Query Syntax	[SOURce:]LIST:SHAP?
Returned Parameters	<CRD>
Related Commands	LIST:SHAP:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:VOLT

**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>
Example	LIST:SHAP:POIN?
Related Commands	LIST:SHAP

**LIST:STEP**

This command specifies how the list sequencing responds to triggers.

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

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

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

**LIST:TTLTrg**

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

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



**LIST:TTLTrg:POINts?**

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

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

**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 the list will be output when a list transient is triggered. Changing list data while a subsystem is in list mode generates an implied ABORT.

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 sinewave, the maximum voltage that can be programmed is 300 V rms.

---

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sinewave.

---

Command Syntax	[SOURce:]LIST:VOLTage[:LEVel] <NRf+>{,<NRf+>}
Parameters	0 to 300 (for sinewaves)
Unit	V (rms voltage)
Examples	LIST:VOLT 2.0,2.5,3.0 LIST:VOLT MAX,2.5,MIN
Query Syntax	[SOURce:]LIST:VOLTage[:LEVel]?
Returned Parameters	<NR3>
Related Commands	LIST:VOLT:POIN? LIST:COUN LIST:DWEL LIST:STEP LIST:SHAP LIST:VOLT:OFFS

**LIST:VOLTage:POINts?**

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

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

**LIST:VOLTage:SLEW**

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

Command Syntax	[SOURce:]LIST:VOLTage:SLEW <NRf+>{,<NRf+>}
Parameters	0 to 9.9E37   INFINITY
Unit	V/S (volts per second)
Example	LIST:VOLT:SLEW 10, 1E2, INF
Query Syntax	[SOURce:]LIST:VOLTage:SLEW?
Returned Parameters	<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>
Example	LIST:VOLT:SLEW:POIN?
Related Commands	LIST:VOLT:SLEW

## 4.2.16 Source Subsystem - Phase

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

### Subsystem Syntax

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

### PHASe

#### Phase Selectable

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

The PHASe command is not influenced by INSTRUMENT:COUPLE ALL. It applies only to the current output phase selected by INSTRUMENT:NSELECT.

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

**PHASe:MODE**

Phase Selectable

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

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

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

**PHASe:TRIGgered**

Phase Selectable

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

The PHASe command is not influenced by INSTRument:COUPlE ALL. It applies only to the current output phase selected by INSTRument:NSElect.

Command Syntax	[SOURce:]PHASe:TRIGgered<NRf+>
Parameters	360° through +360°
*RST Value	triggered phase $\varnothing 1 = 0^\circ$ , triggered phase $\varnothing 2 = 120^\circ$ , triggered phase $\varnothing 3 = 240^\circ$
Examples	PHAS:TRIG 120 PHASE MAX
Query Syntax	[SOURce:]PHASe:TRIGgered?
Returned Parameters	<NR3>
Related Commands	PHAS:MODE      PHAS

## 4.2.17 Source Subsystem - Pulse

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

### Subsystem Syntax

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

### PULSe:COUNT

This command sets the number of pulses that are output when a triggered output transient occurs. The command accepts parameters in the range 1 through 9.9E37. If INFIinity or MAXimum is sent, the output pulse repeats indefinitely.

Command Syntax	[SOURce:]PULSe:COUNT<NRf+>   INFIinity
Parameters	1 to 9.9E37   MINimum   MAXimum   INFIinity
*RST Value	1
Examples	PULS:COUN 3 PULS:COUN MIN PULS:COUN INF
Query Syntax	[SOURce:]PULS:COUNT?
Returned Parameters	<NR3>
Related Commands	PULS:DCYC PULS:HOLD PULS:PER PULS:WIDT

### PULSe:DCYClE

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

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

**PULSe:HOLD**

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

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

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

**Table 4-1 : PULSe:HOLD = WIDTh parameters**

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

**Table 4-2 : PULSe:HOLD = DCYClE parameters**

**PULSe:PERiod**

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

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

**PULSe:WIDTh**

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

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

## 4.2.18 Source Subsystem - Voltage

This subsystem programs the output voltage of the 3000iL and the 4500iL.

### Subsystem Syntax

[SOURce:]		
VOLTage		
[:LEVel]		
[:IMMEDIATE]		
[:AMPLitude] <n>		Sets the ac rms voltage amplitude
:TRIGgered		
[:AMPLitude] <n>		Sets the transient voltage amplitude
:MODE <mode>		Sets the voltage mode (FIX STEP PULS LIST)
:PROTection		
[:LEVel] <n>		Sets the overvoltage protection threshold
:STATe <bool>		Sets the overvoltage protection state
:RANGe <n>		Sets the voltage range
:SENSe   ALC		
:DETeCTOR RTIME   RMS		Sets sense detector for the voltage control loop
:SOURce INTernal   EXTeRnal		Sets voltage sense source
:SLEW		
[:IMMEDIATE] <n>   INFIInity		Sets the voltage slew rate
:MODE <mode>		Sets voltage slew mode (FIX STEP PULS LIST)
:TRIGgered <n>   INFIInity		Sets the transient voltage slew rate

### VOLTage

#### Phase Selectable

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 and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sinewave.

Command Syntax	[SOURce:]VOLTage[:LEVel] [:IMMEDIATE][:AMPLitude]<NRf+>
Parameters	0 to 300 (for sinewaves)
Unit	V (rms voltage)
*RST Value	1 volt
Examples	VOLT 250      VOLT:LEV 25
Query Syntax	[SOURce:]VOLTage[:LEVel] [:IMMEDIATE][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	VOLT:MODE    VOLT:TRIG    FUNC:SHAP



**VOLTage:TRIGgered**

Phase Selectable

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, 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:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sine wave.

---

Command Syntax	[SOURce:]VOLTage[:LEVel]:TRIGgered :AMPLitude]<NRf+>
Parameters	0 to 300 (for sine waves)
Unit	V (rms voltage)
*RST Value	0 volt
Examples	VOLT:TRIG 120 VOLT:LEV:TRIG 120
Query Syntax	SOURce:]VOLTage[:LEVel]:TRIGgered:AMPLitude]?
Returned Parameters	<NR3> If the TRIG level is not programmed, the IMM level is returned.
Related Commands	VOLT VOLT:MODE FUNC:SHAP

**VOLTage:MODE**

Phase Selectable

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

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	FIX
Examples	VOLT:MODE LIST VOLT:MODE FIX
Query Syntax	[SOURce:]VOLTage:MODE?
Returned Parameters	<CRD>
Related Commands	VOLT:TRG VOLT

---

**Note:**

Affects: VOLT:MODE command on 3000iL and 4500iL only.

Version of the iL Series firmware A.00.03 or older have a firmware bug that causes improper execution of the VOLT:MODE FIX command when in three phase mode.

Thus, when one or more phases is set to VOLT:MODE LIST and the remaining phase(s) to VOLT:MODE FIX, the VOLT:MODE FIX is not implemented. Instead, all three phases will operate in LIST mode.

To work around this problem, it is recommended to program those phases that should not be part of the transient list execution to VOLT:MODE LIST instead of VOLT:MODE FIX and set the LIST:VOLT and LIST:VOLT:SLEW to the nominal voltage for the phase in question. By setting only one list point for the affected phase, the iL firmware automatically extends this point to the length of the list programmed for the phases that are programmed with a complete list.

**Example:**

Instead of :

'Transient V drop on phase A only. Also results in drop on B and C

```
ABORT
VOLT 120
FREQ 60
OUTP ON
INST:COUP NONE
FUNC:MODE FIX
FREQ:MODE FIX
INST:NSEL 2
VOLT:MODE FIX
INST:NSEL 3
VOLT:MODE FIX
INST:NSEL 1
VOLT:MODE LIST
LIST:VOLT 0,120
LIST:VOLT:SLEW INF,INF
LIST:DWELL 0.02,1.00
LIST:COUN 1
TRIG:TRAN:SOUR IMM
INIT:NAME TRAN
```

Use :

'Workaround for transient V drop on phase A only. Results in no drop on B and C

```

ABORT
VOLT 120
FREQ 60
OUTP ON
INST:COUP ALL
FUNC:MODE LIST          /* sets all phases to list mode
INST:COUP NONE
INST:NSEL 2
LIST:VOLT 120           /* sets all list points for phase B to 120 V
LIST:VOLT:SLEW INF
INST:NSEL 3
LIST:VOLT 120           /* sets all list points for phase C to 120 V
LIST:VOLT:SLEW INF
INST:NSEL 1
VOLT:MODE LIST
LIST:VOLT 0,120         /* programs a voltage drop on phase A
LIST:VOLT:SLEW INF,INF
LIST:DWELL 0.02,1.00   /* drops for 20 msec
LIST:COUN 1
TRIG:TRAN:SOUR IMM
INIT:NAME TRAN

```

---

### **VOLTage:PROTection**

Phase Selectable

This command sets the overvoltage protection (OVP) level of the AC source. If the peak output voltage exceeds the OVP level, then the AC source output is disabled and the Questionable Condition status register OV bit is set (see Section 6 under Programming the Status and Event Registers). 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
Unit	V (peak voltage)
*RST Value	MAX
Examples	VOLT:PROT 400            VOLT:PROT:LEV MAX
Query Syntax	[SOURce:]VOLTage:PROTection[:LEVel]?
Returned Parameters	<NR3>
Related Commands	OUTP:PROT:CLE

**VOLTage:RANGe**

Phase Selectable

This command sets the voltage range of the AC source. Two voltage ranges are available: a 150 volt range and a 300 volt range. Sending a parameter greater than 150 selects the 300 volt range, otherwise the 150 volt range is selected.

When the range is set to 150, the maximum rms voltage that can be programmed for a sine wave is 150 volts. For other waveshapes, the maximum programmable voltage may be different, depending on the waveform crest factor.

The VOLTage:RANGe command is coupled with the CURRent command. This means that the maximum current limit that can be programmed at a given time depends on the voltage range setting in which the unit is presently operating. Refer to chapter 4 under "Coupled Commands" for more information.

Command Syntax	[SOURce:]VOLTage:RANGe<NRf+>
Parameters	150   300
*RST Value	MAX
Examples	VOLT:RANG 150            VOLT:RANG MIN
Query Syntax	[SOURce:]VOLTage:RANGe?
Returned Parameters	<NR3>
Related Commands	VOLT

**VOLTage:SENSe:DETEctor    VOLTage:ALC:DETEctor**

4801iL Only

These commands select the type of closed loop feedback 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 user's sense terminals, which allows remote voltage sensing at the load.

Command Syntax	[SOURce:]VOLTage:SENSe:SOURce<source> [SOURce:]VOLTage:ALC:SOURce<source>
Parameters	INTernal   EXTernal
*RST Value	INTernal
Examples	VOLT:SENSe:SOUR INT VOLT:ALC:SOUR EXT
Query Syntax	[SOURce:]VOLTage:SENSe:SOURce? [SOURce:]VOLTage:ALC:SOURce?
Returned Parameters	<CRD>
Related Commands	VOLT:SENSe: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 will set the slew to its maximum possible rate. The SCPI representation for INFInity is 9.9E37. This command does not affect the rate at which programmed dc offset changes occur.

Command Syntax	[SOURce:]VOLTage:SLEW[:IMMEDIATE]<NRf+> INFInity
Parameters	0 to 9.9E37   INFInity
Unit	V/S (volts per second)
*RST Value	INFInity
Examples	VOLT:SLEW 1 VOLT:SLEW MAX VOLT:SLEW INF
Query Syntax	[SOURce:]VOLTage:SLEW[:IMMEDIATE]?
Returned Parameters	<NR3>
Related Commands	VOLT:SLEW:MODE VOLT:SLEW:TRIG

**VOLTage:SLEW:MODE**

Phase Selectable

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

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

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

**VOLTage:SLEW:TRIGgered**

Phase Selectable

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

Command Syntax	[SOURce:]VOLTage:SLEW:TRIGgered<NRf+> INFinity
Parameters	0 to 9.9E37   INFinity
Unit	V/S (volts per second)
*RST Value	INFinity
Examples	VOLT:SLEW:TRIG 1      VOLT:SLEW:TRIG MAX VOLT:SLEW:TRIG INF
Query Syntax	[SOURce:]VOLTage:SLEW:TRIGgered?
Returned Parameters	<NR3>
Related Commands	VOLT:SLEW:MODE      VOLT:SLEW

## 4.2.19 Status Subsystem Commands

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

Condition	Enable	Event	NTR Filter	PTR Filter
-----------	--------	-------	------------	------------

Refer to Chapter 6 for more information about the status registers.

### Subsystem Syntax

#### STATus

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

**STATus:PRESet**

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

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

Command Syntax	STATus:PRESet
Parameters	None
Examples	STAT:PRES

Bit Position	15-9	8	7-6	5	4-1	0
Bit Name	not used	CV	not used	WTG	not used	CAL
Bit Weight		256		32		1
CAL	Interface is computing new calibration constants					
WTG	Interface is waiting for a trigger.					
CV	Output voltage is regulated.					

**Table 4-3 : Bit Configuration of Status Operation Registers**

**STATus:OPERation?**

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

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

**STATus:OPERation:CONDition?**

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

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



**STATus:OPERation:ENABLE**

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

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

**STATus:OPERation:NTR STATus:OPERation:PTR**

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

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

---

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

---

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

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

**Table 4-4 : Bit Configuration of Questionable Registers**

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

**STATus:QUESTionable?**

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

On the 3000iL and the 4500iL, each signal that is fed into the Questionable Status Condition register is logically-ORed from three corresponding status signals that originate from each phase.

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

**STATus:QUESTionable:CONDition?**

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

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

**STATus:QUEStionable:ENABle**

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

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

**STATus:QUEStionable:NTR****STATus:QUEStionable:PTR**

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

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

---

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

---

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

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

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

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

**STATus:QUEStionable:INSTrument:ISUMmary?**

Phase Selectable

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

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

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

**STATus:QUESTionable:INSTrument:ISUMmary:CONDition?**

Phase Selectable

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

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

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

**STATus:QUESTionable:INSTrument:ISUMmary:ENABLE**

Phase Selectable

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

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

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

**STATus:QUESTionable:INSTrument:ISUMmary:NTR**  
**STATus:QUESTionable:INSTrument:ISUMmary:PTR**

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

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

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

---

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

---

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

## 4.3 System Commands

The system commands control the system-level functions of the iL Series.

### Subsystem Syntax

SYSTem	
:CONFIguration	
:NOUTputs <n>	Selects the number of output phases
:ERRor?	Returns the error number and error string
:VERSIon?	Returns the SCPI version number
:LANGUage	Sets the programming language
:LOCal	Go to local mode (RS-232 only)
:REMote	Go to remote mode (RS-232 only)
:RWLock	Go to remote with lockout mode (RS-232 only)

### SYSTem:CONFIgure:NOUTputs

This command selects the number of outputs phases for the AC source. This requires that the AC source is capable of switching between single and three phase mode. Both the 4500iL and the 3000iL have this capability.

---

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

---

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

---

**Note:** The AC source *must* be calibrated in the three phase mode to properly execute this command.

---

This bus command corresponds to the "NOUTPUTS" item found behind the ADDRESS key on the front panel.

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



**SYSTem:ERRor?**

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

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

**SYSTem:VERSion?**

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

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

**SYSTem:LANGuage**

Sets the command language of the AC Power Source to either SCPI or Elgar Model 9012 PIP. 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 or E9012
Example	SYST:LANG:SCPI
Query Syntax	SYSTem LANGuage?
Returned Parameters	<CRD>

**SYSTem:LOCal**

This command can only be used with the RS-232 interface. It sets the interface in Local state, which enables the front panel controls.

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

**SYSTem:REMOte**

This command can only be used with the RS-232 interface. 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
Example	SYST:LOC
Related Commands	SYST:LOC SYST:RWL

**SYSTem:RWLock**

This command can only be used with the RS-232 interface. 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	SYSTem:RWLock
Parameters	None
Example	SYST:RWL
Related Commands	SYST:LOC SYST:REM

### 4.3.1 Trace Subsystem Commands

This subsystem programs the output waveform of the 3000iL and the 4500iL. 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

#### 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  $\pm 1$ . You can query the predefined SINusoid, SQUare, or CSINusoid waveform shapes, but you cannot use the predefined names as names for your waveform.

Waveform data is stored in nonvolatile memory and is retained when input power is removed. Up to 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>
Example	TRAC flattop,0.1,0.3,0.7,.....-0.7,-0.3,-0.1
Query Syntax:	TRACe[:DATA]?<waveform_name>
	DATA[:DATA]?<waveform_name>
Returned Parameters	<NR3> (a total of 1024 data points)
Related Commands	TRAC:DATA TRAC:DEL FUNC:SHAP

**TRACe:CATalog****DATA:CATalog**

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

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

**TRACe:DEFine****DATA:DEFine**

These commands define a new waveform with the name <waveform\_name> and allocates storage for its data. The waveform name can then be referenced by the TRACe:DATA command to define its data values.

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

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

**TRACe:DELeTe****DATA:DELeTe**

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

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

## 4.3.2 Trigger Subsystem

This subsystem controls the triggering of the AC source. See chapter 5.7 under Triggering Output Changes for an explanation of the Trigger Subsystem. The INITiate commands control the initialization of both the transient and measurement trigger systems.

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

### Subsystem Syntax

ABORT	Resets the trigger system to the Idle state
INITiate	
[:IMMEDIATE]	Initiates the system for one trigger
:SEQUENCE[1 3]	Initiates a specific numbered sequence
:NAME TRANSient   ACQUIRE	Initiates a specific named sequence
:CONTINUOUS	
:SEQUENCE[1] <bool>	Sets continuous initialization
:NAME TRANSient <bool>	Sets continuous initialization
TRIGGER	
[:SEQUENCE1   :TRANSient]	
[:IMMEDIATE]	Triggers the output immediately
:DELAY <n>	Sets the trigger delay time
:SOURCE <source>	Sets the trigger source (BUS EXT IMM)
:SEQUENCE2   :SYNCRONIZE	
:SOURCE PHASE   IMMEDIATE	Sets the synchronous source
:PHASE <n>	Sets the synchronous phase reference
:SEQUENCE3   :ACQUIRE	
[:IMMEDIATE]	Triggers the measurement immediately
:SOURCE <source>	Sets the trigger source (BUS EXT TTL)
:SEQUENCE1	
:DEFINE TRANSient	Sets or queries the SEQ1 name
:SEQUENCE2	
:DEFINE SYNCRONIZE	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 (see ). ABORt is executed at power turn-on and upon execution of \*RCL, RST, or any implied abort command (see List Subsystem).

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

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

## INITiate:SEQuence

### INITiate:NAME

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

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

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

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

**INITiate:CONTInuous:SEQuence****INITiate:CONTInuous:NAME**

These commands control the transient generator trigger system.

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

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

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

**TRIGger**

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

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

### TRIGger:DELay

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

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

### TRIGger:SOURce

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

- BUS IEEE-488 device, \*TRG, or <GET> (Group Execute Trigger)
- EXTernal AC source's backpanel Trigger 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	IMM
Examples	TRIG:SYNC:SOUR IMM TRIG:SEQ2:SOUR PHAS
Query Syntax	TRIGger:SEQuence2:SOURce? TRIGger:SYNChronize:SOURce?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

**TRIGger:SEQuence2:PHASe****TRIGger:SYNChronize:PHASe**

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

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

**TRIGger:SEQuence3**

**TRIGger:ACQuire**

When the trigger subsystem has been initiated, these commands generate a measurement trigger regardless of the selected trigger source. The measurement trigger causes the AC source to digitize the instantaneous output 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:SOUR TRIG:DEL TRIG:SYNC TRIG:SYNC:PHAS INIT INIT:CONT *TRG *WAI

**TRIGger:SEQuence3:SOURce**

**TRIGger:ACQuire:SOURce**

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

- BUS IEEE-488 device, \*TRG, or <GET> (Group Execute Trigger)
- EXTernal backpanel Trigger In BNC
- TTLTrg the signal driving the Trigger Out BNC

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

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

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

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

## 4.4 Common Commands

---

Common commands begin with an \* and consist of three letters (command) or three letters and a ? (query). Common commands are defined by the IEEE 488.2 standard to perform some common interface functions. The 3000iL and the 4500iL respond to the required common commands that control status reporting, synchronization, and internal operations. The 3000iL and the 4500iL 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 chapter 6 under Programming the Status and Event Registers which explains how to read specific register bits and use the information that they return.

### Common Commands Syntax

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

**\*CLS**

This command clears the following registers (see chapter 6 under Programming the Status and Event Registers for descriptions of all registers):

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

Command Syntax	*CLS
Parameters	None

**\*ESE**

This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event register (see \*ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Event Register are logically ORed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. See section 6.5 for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

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

**Table 4-6 : Bit Configuration of Standard Event Status Enable Register**

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

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

**\*ESR?**

This query reads the Standard Event Status Event register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see \*ESE). See section 6.5 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?	
Returned Parameters	<AARD>	
	<u>Field</u>	<u>Information</u>
	California Instruments	Manufacturer
	4500iL	6-digit model number
	nnnnA-nnnn	10-character serial number or 0
	<R>.xx.xx	Revision levels of firmware
Example	CALIFORNIA INSTRUMENTS,4500IL,0,A.00.01	

**\*OPC**

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

- all commands sent before \*OPC have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect output voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the iL Series. 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 to identify any options that are installed. Options are identified by number. A 0 indicates no options are installed.

Query Syntax	*OPT?
Returned Parameters	<AARD>

**\*PSC**

This command controls the automatic clearing at power-on of the Service Request Enable and the Standard Event Status Enable registers (see chapter 6 under Programming the Status and Event Registers for register details):

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

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

**\*RCL**


---

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

---

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

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

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

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



**\*RST**

This command resets the AC source to the factory-defined states shown in Table 4-7.

Item	Value	Item	Value
CAL:STAT	OFF	[SOUR:]FUNC:MODE	FIX
DIAG:FAN:SPE:MAN	50%	[SOUR:]FUNC:TRIG	SIN
DIAG:FAN:MODE	AUTO	[SOUR:]FUNC:CSIN	100%
DISP:STAT	ON	[SOUR:]LIST:COUN	1
DISP:MODE	TEXT	[SOUR:]LIST:STEP	AUTO
INIT:CONT	OFF	[SOUR:]PHAS	$\phi 1 = 0^\circ$ , $\phi 2 = 240^\circ$ , $\phi 3 = 120^\circ$
INST:COUP	ALL	[SOUR:]PHAS:TRIG	$\phi 1 = 0^\circ$ , $\phi 2 = 240^\circ$ , $\phi 3 = 120^\circ$
INST:NSEL	1	[SOUR:]PHAS:MODE	FIX
OUTP	OFF	[SOUR:]PULS:COUN	1
OUTP:DFI	OFF	[SOUR:]PULS:DCYC	50%
OUTP:DFI:SOUR	OFF	[SOUR:]PULS:HOLD	WIDT
OUTP:PROT:DEL	100ms	[SOUR:]PULS:PER	1
OUTP:RI:MODE	OFF	[SOUR:]PULS:WIDT	0.01667s
OUTP:TTLT	OFF	[SOUR:]VOLT	1
OUTP:TTLT:SOUR	BOT	[SOUR:]VOLT:TRIG	1
SENS:SWE:OFFS:POIN	0	[SOUR:]VOLT:MODE	FIX
SENS:SWE:TINT	25 $\mu$ s	[SOUR:]VOLT:PROT	MAX
SENS:WIND	KBESsel	[SOUR:]VOLT:RANG	MAX
[SOUR:]CURR	1A	[SOUR:]VOLT:SENS:SOUR	INT
[SOUR:]CURR:PROT:STAT	OFF	[SOUR:]VOLT:SLEW	INF
[SOUR:]FREQ	60Hz	[SOUR:]VOLT:SLEW:MODE	FIX
[SOUR:]FREQ:MODE	FIX	[SOUR:]VOLT:SLEW:TRIG	INF
[SOUR:]FREQ:SLEW	INF	TRIG:DEL	0
[SOUR:]FREQ:SLEW:MODE	FIX	TRIG:SOUR	BUS
[SOUR:]FREQ:SLEW:TRIG	INF	TRIG:SEQ2:SOUR	IMM
[SOUR:]FREQ:TRIG	60Hz	TRIG:SEQ2:PHAS	0
[SOUR:]FUNC	SIN	TRIG:SEQ3:SOUR	BUS

**Table 4-7 : factory-defined \*RST states**

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

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

**\*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 nonvolatile memory. If a particular state is desired at power-on, it should be stored in location 0. It then will be recalled at power-on if the OUTPUT:PON:STATE command is set to RCL0. Use \*RCL to retrieve instrument states.

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

**\*SRE**

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

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

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

**\*STB?**

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

Bit Position	7	6	5	4	3	2 - 0
Bit Name	OPER	MSS RQS	ESB	MAV	QUES	not used
Bit Weight	128	64	32	16	8	

**Table 4-8 : Bit Configuration of Status Byte Register**

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

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

**\*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 to do a self-test and report any errors.

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

### **\*WAI**

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

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

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

Command Syntax	*WAI
Parameters	None
Related Commands	*OPC

## 4.5 IEC 1000-2 Commands

---

### Introduction

This section pertains to the 4801iL AC source only.

The California Instruments 4801iL Harmonic/Flicker Test System is designed to operate in Normal as well as IEC555 mode. In normal mode, the Harmonic/Flicker Test System responds to all of the commands that program AC source operation. In IEC555 mode, the Harmonic/Flicker Test System provides full IEC555-2 and IEC555-3 compliance test capability. The System Mode command details the differences between Normal and IEC555 mode.

When the HFT System is being used in IEC555 mode, the HFT software handles all of the communication between the user interface and the HFT System. The HFT software must be loaded and running on a personal computer that is connected to the HFT System.

The IEC555 commands that are described in this section are for those users who need to directly program the IEC555 functions of the HFT System without using the HFT software. Be aware that these commands will return “raw” IEC555 data from the HFT system. It is the programmer’s responsibility to interpret the data according to the IEC555 standards.

## Command Syntax

## CALCulate

:INTEgral

:TIME &lt;Nrf+&gt;

selects the Pst integration time for flicker measurements

:LIMit

UPPer

[:DATA] &lt;Nrf+&gt;

sets various limits associated with rms voltage fluctuation testing for IEC 1000-3-3

:SMOothing &lt;bool&gt;

turns the smoothing filter on or off

## FORMat

[:DATA] &lt;type&gt;

specifies the response data format (ASCIi | REAL)

:BORder &lt;type&gt;

sets the byte order of the floating point values returned (NORMal | SWAPPed)

## MEASure

:ARRay

:CURRent

:HARMonic? &lt;Nrf+&gt;

returns an array of current harmonic magnitudes

:VOLTage

:FLUCtuations

:FLICKer? &lt;Nrf+&gt;

measures voltage fluctuations in accordance with the IEC868 standard

:PST? &lt;Nrf+&gt;

:ALL? &lt;Nrf+&gt;

## SENSE

:CURRent

:PREference &lt;type&gt;

sets the phase reference for current harmonic phase measurements (VOLTage | CURRent)

:WINDow

[:TYPE] &lt;type&gt;

selects the window function used in the harmonic measurements (HANNing | KBESSel | RECTangular)

## SYSTEM

:CONFigure &lt;mode&gt;

selects the operating mode of the HFT system (NORMal | IEC)

**CALCulate:INTEgral:TIME**

This command selects the Pst integration time for IEC Flicker measurements. The parameter may only assume values of 1, 5, 10, and 15 minutes in accordance with IEC868. The command will be accepted and may be queried, but will have no meaningful function unless the AC source is placed in IEC555 mode using the SYSTEM:CONFigure command.

Command Syntax:	CALCulate:INTegral:TIME <NRf+>
Parameters:	1, 5, 10, & 15 minutes
*RST Value:	10 minutes
Examples:	CALC:INT:TIME 10
Query Syntax:	CALCulate:INTegral:TIME?
Returned Parameters:	<NR3>
Related Commands:	SYSTEM:CONF MEAS:ARR:VOLT:FLUC:FLIC? MEAS:ARR:VOLT:FLUC:PST? MEAS:ARR:VOLT:FLUC:ALL?

### CALCulate:LIMit:UPPer

This command sets various limits associated with rms voltage fluctuations testing for IEC 1000-3-3, as described in the following table. All five parameters are type NRf. The order in which the five parameters are entered must correspond to the order in the table.

vss delta	Sets the maximum peak-to-peak variation of relative voltage <sup>1</sup> that defines “steady-state”. At *RST this value is set to 0.003 <sup>2</sup> . Note that this number is not specified by IEC 1000-3-3.
dmax limit	Sets the maximum relative voltage <sup>1</sup> change allowed before a dmax error is flagged. At *RST this value is set to 0.04 (see note <sup>2</sup> ).
dc limit	Sets the maximum relative steady-state voltage change allowed before a dc error is flagged. At *RST this value is set to 0.03 (see note <sup>2</sup> ).
dt tlimit	Sets the maximum time in seconds that the relative voltage <sup>1</sup> can exceed the dt limit before a dt error is flagged. At *RST this value is set to 0.2 seconds.
dt limit	Sets the maximum relative voltage <sup>1</sup> that must be exceeded for dt tlimit seconds before a dt error is flagged. At *RST this value is set to 0.03 (see note <sup>2</sup> ).

<sup>1</sup>The expression “relative voltage” as used above is the measured rms voltage divided by the programmed voltage.

<sup>2</sup>This value is the ratio with respect to  $U_n$  (the European nominal line voltage). For example, a value of .03 represents 6.9 volts if  $U_n = 230$  volts. (Ratio \* 100 = % of  $U_n$ )

Command Syntax	CALCulate:LIMit:UPPer[ :DATA] <NRf>,<NRf>,<NRf>,<NRf>,<NRf>
Parameters	see table
*RST Value	see table
Examples	CALC:LIM:UPP .003, .04, .03, .2, .03
Returned Parameters	<NR3>

### CALCulate:SMOothing

This command turns on or off a smoothing filter for current harmonic measurements. The filter transfer function is equivalent to a single pole lowpass function with a 1.5 second time constant and is applied only to current harmonic measurements made when IEC555 mode is selected with SYSTem:CONF.

Command Syntax:	CALCulate:SMOothing <Bool>
Parameters:	0   1   ON   OFF
*RST Value:	OFF
Examples:	CALC:SMO ON
Query Syntax:	CALCulate:SMOothing?
Returned Parameters:	<CRD>
Related Commands:	MEAS:ARR:CURR:HARM? SYST:CONF MEAS:CURR:HARM?

## FORMat

This command specifies the response data format for the following queries:

```
MEASure:ARRay:CURRent:DC?
MEASure:ARRay:CURRent:HARMonic[:AMPLitude]?
MEASure:ARRay:VOLTage:DC?
MEASure:ARRay:VOLTage:FLUCtuations:ALL?
MEASure:ARRay:VOLTage:FLUCtuations:FLICKer?
MEASure:ARRay:VOLTage:FLUCtuations:PST?
```

When ASCII is selected, the response format for these queries is NR3 Numeric Response Data. This format is selected at \*RST. The only valid argument for <length> is 0, which means that the AC source selects the number of significant digits to be returned.

When REAL is selected, the response format is Definite Length Arbitrary Block Response Data. The data within the Arbitrary Block is coded as IEEE single precision floating point, with 4 bytes per value. The second argument to the FORMat:DATA command specifies the number of bits in the returned data. Only the value 32 is permitted in AC source instruments. The byte order within a single value is determined by the FORMat:BORDER command. Definite Length Arbitrary Block Response Data format begins with a header that describes the number of data bytes in the response. The header begins with a pound sign, followed by a single non-zero digit that defines the number of digits in the block length, followed by the digits contained in the block.

For example, the response to the query "MEAS:ARR:CURR:HARM? 1" which returns 45 numeric values when SYSTem:CONFigure is set to IEC would be as follows:

```
# '3' '1' '8' '0' <byte1> <byte2> . . . <byte180> <newline>
```

When a query requests a number of response blocks, each block is separated by the Response Data Separator (comma). For example: The response to the query "MEAS:ARR:CURR:HARM? 2" given under the same conditions described in the example above would be as follows:

```
# '3' '1' '8' '0' <byte1> <byte2> . . . <byte180> ',' # '3' '1' '8' '0' <byte1> <byte2> . . . <byte180> <newline>
```



Command Syntax:	FORMat[:DATA] <CRD>
Parameters:	ASCIi   REAL
*RST Value:	ASCIi
Examples:	FORM REAL
Query Syntax:	FORMat?
Returned Parameters:	<CRD>
Related Commands:	FORM:BORD                    MEAS:ARR:CURR:DC? MEAS:ARR:VOLT:DC? MEAS:ARR:CURR:HARM[:AMPL]? MEAS:ARR:VOLT:FLUC:ALL? MEAS:ARR:VOLT:FLUC:FLIC? MEAS:ARR:VOLT:FLUC:PST?

**FORMat:BORDER**

This command sets the byte order of IEEE floating point values returned within Arbitrary Block Response Data. When NORMal is selected, the first byte sent is the sign bit and seven most significant bits of the exponent, and the last byte sent is the least significant byte of the mantissa. This ordering is most useful for big-endian controllers such as those that use Motorola processors.

When SWAPped is selected, the least significant byte of the mantissa is sent first and the sign bit and seven most significant bits of the exponent are sent last. This ordering is most useful for little-endian controllers such as those that use Intel processors.

Command Syntax:	FORMat:BORDER <CRD>
Parameters:	NORMal   SWAPped
*RST Value:	NORMal
Examples:	FORM:BORD SWAP
Query Syntax:	FORMat:BORDER?
Returned Parameters:	<CRD>
Related Commands:	FORM[:DATA] MEAS:ARR:CURR:DC? MEAS:ARR:VOLT:DC? MEAS:ARR:CURR:HARM[:AMPL]? MEAS:ARR:VOLT:FLUC:ALL? MEAS:ARR:VOLT:FLUC:FLIC? MEAS:ARR:VOLT:FLUC:PST?

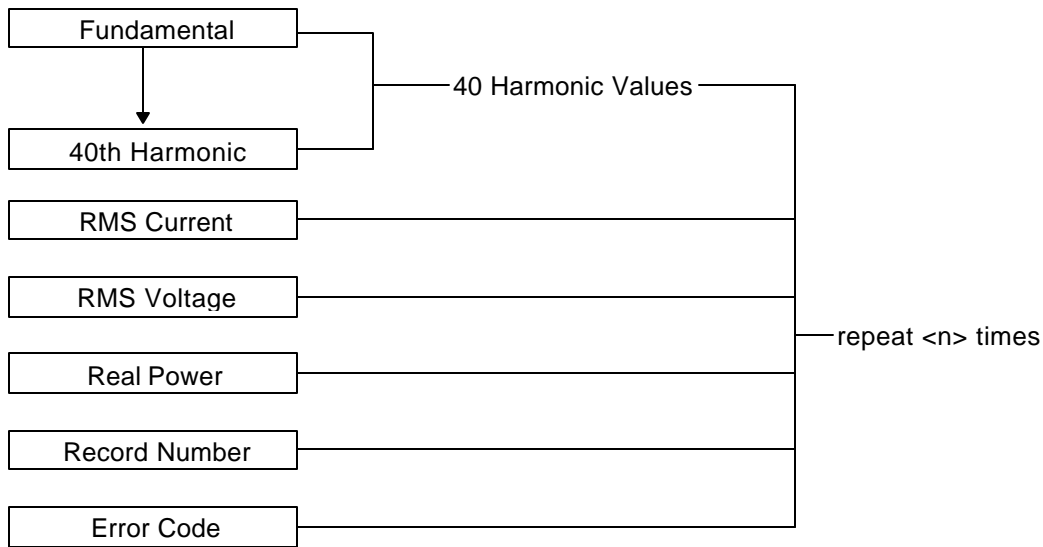
**MEASure:ARRay:CURRent:HARMonic?**

This query returns an array of current harmonic magnitudes. Operation of the query is modified by the SYSTem:CONF command (see summary table under SYSTem:CONFigure). The parameter specifies the number of harmonic arrays to be returned in response to the query. If SYSTem:CONFigure specifies NORMal operation, the parameter is ignored (i.e. it is forced to 1). If SYSTem:CONFigure specifies IEC operation, the SOURce:FREQuency and SENSE:WINDow commands are coupled to modify operation of the measurement underlying the query as shown in the following table:

SOURce:FREQ	SENSe:WINDow	Sample Rate	Acquisition Window	Acquisition Overlap
50 Hz	RECTangular	12.8 kHz	320 ms	None
50 Hz	HANNing	8.533 kHz	480 ms	50%
60 Hz	RECTangular	15.360 kHz	266.7 ms	None
60 Hz	HANNing	7.680 kHz	533.3 ms	50%

SYSTem:CONFIgure also affects availability of the RMS Current, RMS Voltage, and Real Power values. If SYSTem:CONFIgure is set to NORMal, these values are not available. If SYSTem:CONFIgure is set to IEC, the values are returned with the harmonic data. The integration time for these values equals the acquisition window period.

IEC mode operation conforms to IEC and EN requirements for compliance testing of harmonic currents (EN 61555-2 and related standards). The AC source will accept parameters in the range shown below, however, values greater or equal to  $2^{31} - 1$  will be interpreted as infinity. Record numbering begins with zero. The figure below defines the structure of the data returned by this query:



Command Syntax: MEASure:ARRay:CURRent:HARMonic? <NRf+>  
 Parameters: 1 to 9.9E37  
 Examples: MEAS:ARR:CURR:HARM? 1024  
 Related Commands: ABORt SYST:CONF INST:NSEL SENS:WIND  
 SOUR:FREQ

**MEASure:ARRay:VOLTage:FLUCtuations:ALL?**

This query measures voltage fluctuations in accordance with the IEC868 standard. It is only available when IEC mode is selected with SYSTem:CONFIgure. The parameter specifies the number of Pst integration periods during which data will be returned in response to the query.

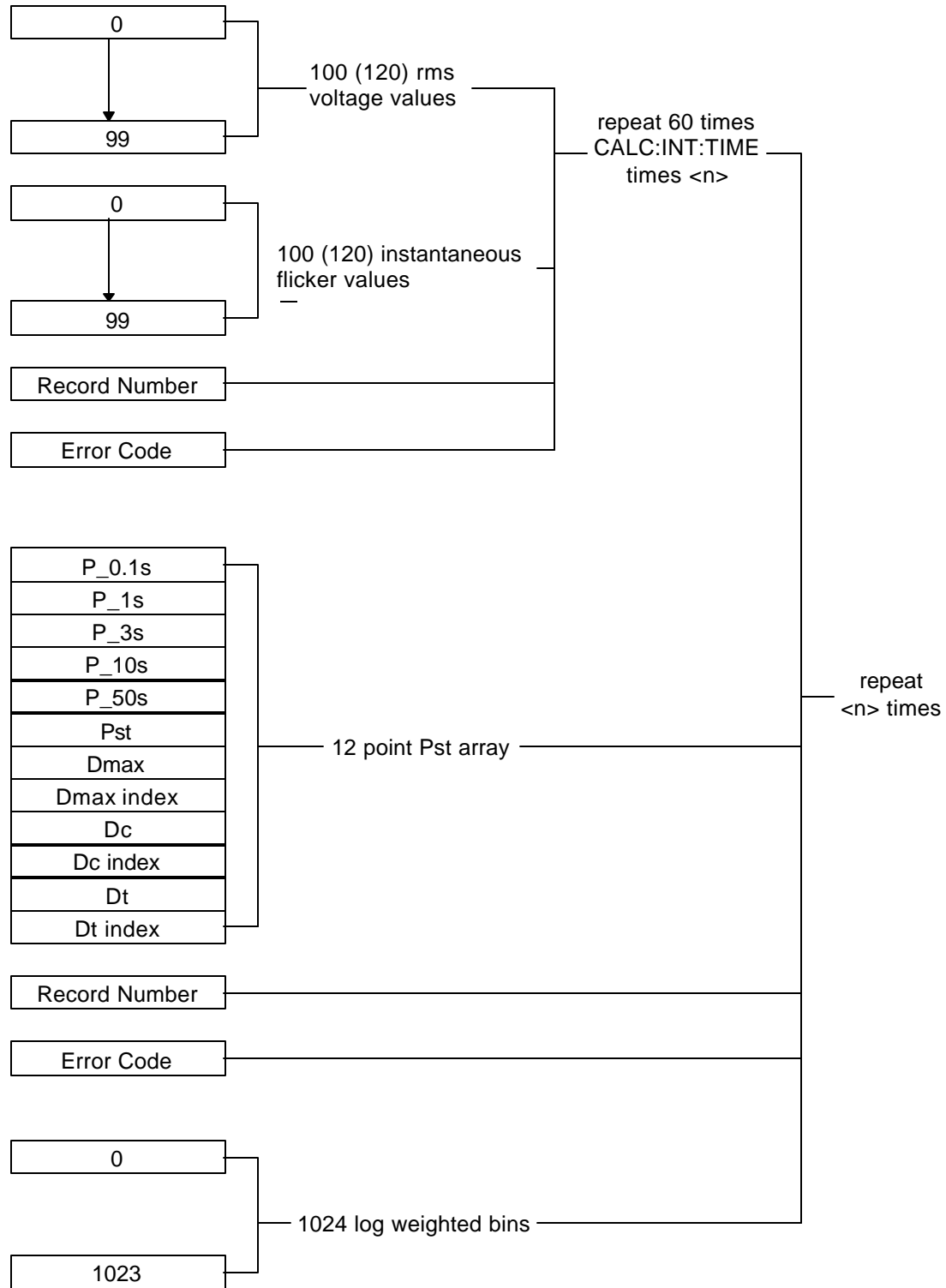
This query returns the data structures associated with both the MEAS:ARR:VOLT:FLUC:FLIC query and the MEAS:ARR:VOLT:FLUC:PST query. The Pst structure includes flicker perceptibility values for the component percentiles making up Pst, the Pst value, various RMS voltage values (Dmax, Dc, and Dt), together with indices for these RMS values that give their exact location in the RMS time series for the corresponding integration period.

An additional structure consisting of a 1024 point array of bins whose indices correspond to a logarithmically weighted range of instantaneous flicker values is returned for each Pst integration period. The array covers a flicker perceptibility range of 0.01 to 10,000 and the individual bins contain counts equal to the accumulated number of occurrences of flicker within the bin range during the Pst integration period. RMS voltage and instantaneous flicker values are returned once a second, while Pst data and the 1024 point arrays are returned once per Pst integration period. The data is always returned in order (i.e. the Pst summary data immediately follows the last array of RMS voltage and flicker values for a given integration period).

The total quantity of data returned by this query is demonstrated by the following example (assuming 50 Hz operation):

If CALCulate:INTEgral:TIME specifies 10 minutes and <n> is set to 12, a 2 hour measurement is initiated (10 minutes times 12) and a total of 1,466,856 data points are returned (202 times 60 times 10 plus the 14 item Pst summary record plus 1024 log points all times 12 Pst integration periods).

This command is closely related to two similar commands that return different data (see MEAS:ARR:VOLT:FLUC:FLIC and MEAS:ARR:VOLT:FLUC:ALL). The figure below defines the structure of the data returned by this query:



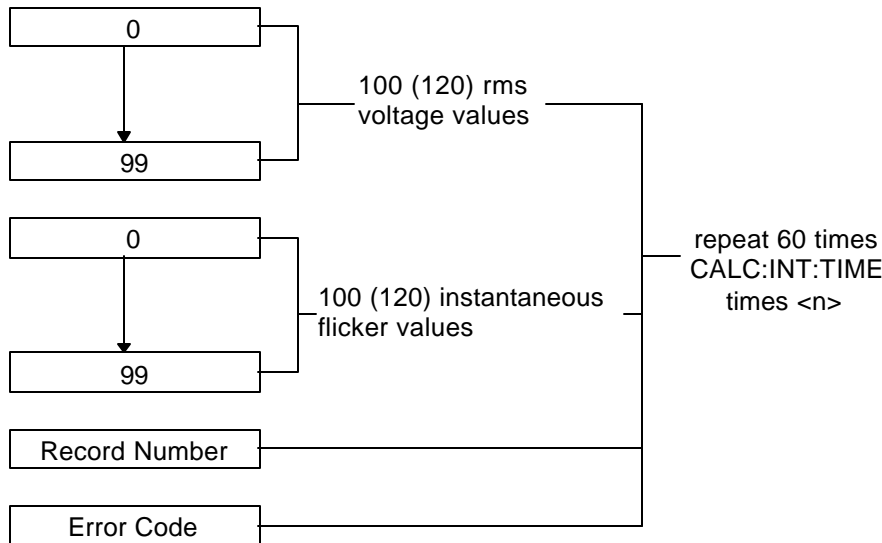
Command Syntax:	MEASure:ARRay:VOLTage:FLUCtuations:ALL? <NRf+>
Parameters:	1 to 1008
Examples:	MEAS:ARR:VOLT:FLUC:ALL? 12
Returned Parameters:	13,158 to 220,588,704 values
Related Commands:	ABOR SYST:CONF INST:NSEL MEAS:ARR:VOLT:FLUC:PST? MEAS:ARR:VOLT:FLUC:FLIC?

**MEASure:ARRar:VOLTage:FLUCtuatuions:FLICker?**

This query measures voltage fluctuations in accordance with the IEC868 standard. It is only available when IEC mode is selected with SYSTem:CONFigure. The parameter specifies the number of Pst integration periods during which voltage fluctuation arrays will be returned in response to the query. The data contained within the arrays represents RMS voltage values integrated over successive half line cycles and the corresponding instantaneous flicker values. This query returns structured data at a rate of one packet per second, with each packet containing 202 (50 Hz) or 242 (60 Hz) data points, for a period of time determined by the specified CALCulate:INTegral:TIME and the parameter specifying the number of Pst integration periods.

For example (assuming 50 Hz operation), if CALCulate:INTegral:TIME specifies 10 minutes and <n> is set to 12, a 2 hour measurement is initiated (10 minutes times 12) and 1,454,400 (202 points/sec times 60 times 10 minutes times 12) data points are returned.

This command is closely related to two similar commands that return different types of data (see MEAS:ARR:VOLT:FLUC:PST and MEAS:ARR:VOLT:FLUC:ALL). Record numbering begins with zero. The figure below defines the structure of the data returned by this query:



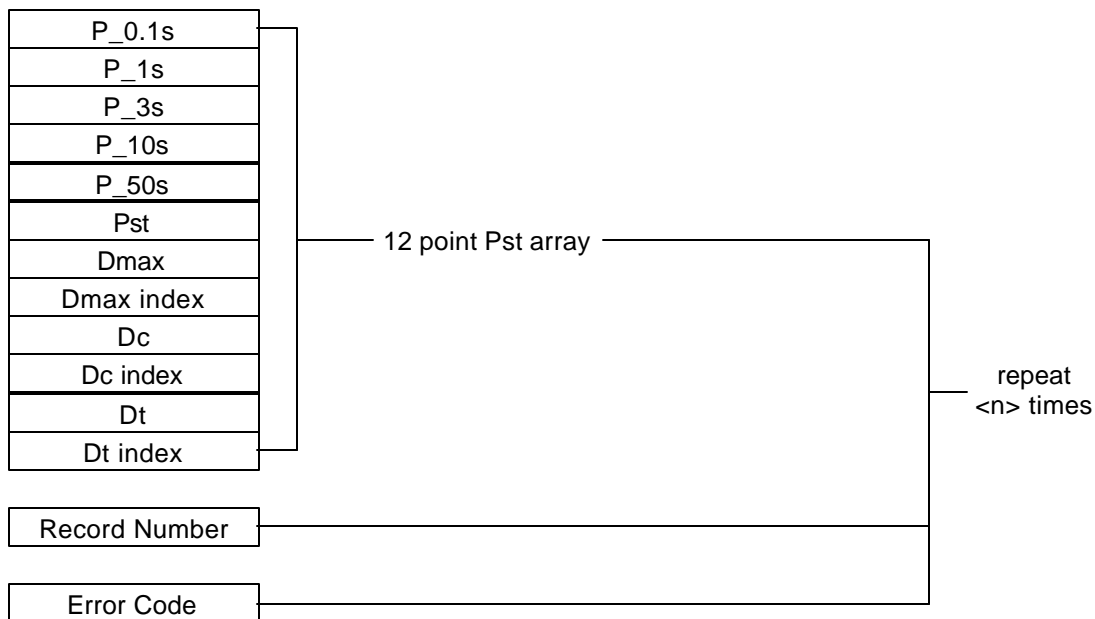
Command Syntax:	MEASure:ARRay:VOLTage:FLUCtuations:FLICker? <NRf+>
Parameters:	1 to 1008
Examples:	MEAS:ARR:VOLT:FLUC:FLIC? 12
Returned Parameters:	12,120 to 219,542,400 values
Related Commands:	ABOR SYST:CONF INST:NSEL MEAS:ARR:VOLT:FLUC:PST? MEAS:ARR:VOLT:FLUC:ALL?

**MEASure:ARRay:VOLTage:FLUCtuations:PST?**

This query measures voltage fluctuations in accordance with the IEC868 standard. It is only available when IEC mode is selected with SYSTem:CONFigure. The parameter specifies the number of Pst integration periods during which data will be returned in response to the query. This query returns 1 data structure per specified integration period for a total of <n> structures.

For example, if CALCulate:INTEgral:TIME specifies 10 minutes and <n> is set to 12, a 2 hour measurement is initiated (10 minutes times 12) and 12 structures are returned. Since there are 14 data points per structure, a total of 168 points are returned. The structure includes flicker perceptibility values for the component percentiles making up Pst, the Pst value, various RMS voltage values (Dmax, Dc, and Dt), together with indices for these RMS values that give their exact location in the RMS time series for the corresponding integration period.

This command is closely related to two similar commands that return different types of data (see MEAS:ARR:VOLT:FLUC:FLIC and MEAS:ARR:VOLT:FLUC:ALL). Record numbering begins with zero. The figure below defines the structure of the data returned by this query:



Command Syntax:	MEASure:ARRay:VOLTage:FLUCtuations:PST? <NRf+>
Parameters:	1 to 1008
Examples:	MEAS:ARR:VOLT:FLUC:PST? 12
Returned Parameters:	14 to 14,112 values
Related Commands:	ABOR SYST:CONF INST:NSEL MEAS:ARR:VOLT:FLUC:FLIC? MEAS:ARR:VOLT:FLUC:ALL?

**SENSE:CURRENT:PREFERENCE**

This command sets the phase reference for current harmonic phase measurements. If VOLTage is selected, the reference is the fundamental component of the measured output voltage. If CURRENT is selected, the reference is the fundamental component of the measured output current.

Command Syntax:	SENSE:CURRENT:PREFERENCE <CRD>
Parameters:	VOLTage   CURRENT
*RST Value:	CURRENT
Examples:	SENS:CURR:PREF CURR
Query Syntax:	SENSE:CURRENT:PREF?
Returned Parameters:	<CRD>
Related Commands:	ABORT MEAS:ARR:CURR:PHAS

**SYSTEM:CONFIGURE**

This command sets the overall operating mode for the HFT system. The choices are normal mode, which causes the product to closely mimic the operating characteristics of an AC source, or IEC mode, which modifies the basic behavior of the transient and measurement systems to facilitate IEC measurements. SYSTEM:CONFIGURE has a variety of global consequences that are summarized below:

FEATURE	NORMAL MODE	IEC MODE
<b>Base Sampling Rate</b>	39.920792 kHz	38.400000 kHz
<b>Output Frequency</b>	45 - 1000 Hz	50 Hz and 60 Hz only
<b>Freq/Window/Fs Mode</b>	Independent	Coupled
<b>Transient System</b>	FIXEd/STEP/PULSE/LIST modes	FIXEd Mode only
<b>Slew Operation</b>	Voltage; Frequency	Voltage only
<b>MEASure:ARRAY: CURRENT:HARM?</b>	DC, Fundamental, and Harmonics to 50th	Fundamental and Harmonics to 40th plus RMS Current, RMS Voltage, and Real Power
<b>CALCulate:SMOothing</b>	Not Available	1.5 Second Smoothing
<b>MEASure:ARRAY: VOLTage:FLUCtuations</b>	Not Available	FLICKer   PST   ALL?
<b>CALCulate:INTegral:TIME</b>	Not Available	1   5   10   15 MINutes

Transmission of a SYSTem:CONFigure command implies ABORt and terminates any transient or measurement actions previously initiated. Selection of normal operating mode does not precisely match operation of standard AC source products because the base sampling rate of the standard products is 39.940828 kHz. Otherwise operation is identical.

Command Syntax:	SYSTem:CONFigure <CRD>		
Parameters:	NORMal   IEC		
*RST Value:	NORMal		
Examples:	SYST:CONF NORM		
Query Syntax:	SYSTem:CONFigure?		
Returned Parameters:	<CRD>		
Related Commands:	ABOR	MEAS:ARR:CURR	MEAS:ARR:VOLT
	CALC:SMO	CALC:INT	SENS:WIND



## 5. Programming Examples

---

### 5.1 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

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

## 5.2 Programming the Output

---

### Power-on Initialization

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

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

\*RST is a convenient way to program all parameters to a known state. Refer to the \*RST command in Section 4.4 to see how each programmable parameter is set by \*RST. Refer to the \*PSC command in Section 4.4 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
```

In the three-phase mode, all phases are programmed to 125 volts rms because the INSTRument:COUPle at \*RST is set to ALL.

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

### 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 volts.

---

**Note:** You cannot program a voltage that produces a higher volt-second on the output than a 300V rms sine wave.

---

Because voltage commands are coupled with the waveform shape command, 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" for more information.

### Voltage Ranges

The iL Series has two voltage ranges that are controlled by a relay that switches taps on an output transformer. The command that controls the range is:

```
VOLTage:RANGe MIN | MAX | 150 | 300
```

When the range is set to MIN (or 150), the maximum rms voltage that can be programmed for a sine waveshape is 150 volts, but it is only on this range that the maximum output current rating is available. For other waveshapes, the maximum programmable voltage may be different, depending on the waveform's voltage crest factor (peak-to-rms ratio).

On the iL Series, the *VOLTage:RANGe* command is coupled with the *CURRent* command. This means that the maximum current limit that can be programmed at a given time depends on the voltage range setting in which the unit is presently operating. Refer to "Coupled Commands" 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 and frequency changes. This can be used to generate ramps or to protect sensitive loads. To set the voltage slew rate to 20 volts 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. See "Programming Transient Outputs".

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

#### Frequency Slew

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

```
FREQuency:SLEW 30
```

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

## 5.2.1 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  $n_1 \dots n_{1024}$  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  $\pm 1$ . Waveform data is stored in nonvolatile memory and is retained when input power is removed. Up to 12 user defined waveforms may be created and stored.

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" for more information.

## 5.2.2 Individual Phases

In the 3000iL and the 4500iL, the following functions can be controlled separately in each phase:

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

### Selecting a Phase

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

*INSTRument:COUPlE ALL | NONE*

*INSTRument:NSElect <n>*

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

To send a programming command to only one of the output phases, set *INSTRument:COUPlE* to NONE, then select the desired output to receive the command with *INSTRument:NSElect*. For example, when the commands

*INSTRument:COUPlE NONE*

*INSTRument:NSElect 2*

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

The *INSTRument:COUPlE* command has no effect on queries. In the three-phase mode, queries are always directed to the output selected by *INSTRument:NSElect*.

### Programming the Output Phase

You can control the phase of the ac 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 3000iL and the 4500iL, the *PHASe* command sets the relative phase of each of the outputs. The *INSTRument:COUPlE* setting is ignored by the *PHASe* command - it always controls the output selected by *INSTRument:NSElect*.

### 5.2.3 Current Limit

---

All 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.

---

**Note:** The *CURRent* command is coupled with the *VOLTage:RANGe*. This means that the maximum current limit that can be programmed at a given time depends on the voltage range setting in which the unit is presently operating. Refer to "Coupled Commands" for more information.

---

---

## 5.3 Coupled Commands

---

This section describes how to avoid programming errors that may be caused because of the error checking done for coupled commands such as `VOLTage:LEVel` and `FUNCTion:SHAPE`.

Commands that are coupled to the `VOLTage` 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.

*CURRENT:LEVel and VOLTage:RANGe*

Programming the current limit by itself to a value that is greater than the maximum allowed on the presently programmed voltage range causes an error. If the commands

*VOLTage:RANGe 300*

*CURRENT 10*

are sent, an error will be generated because the `CURRENT` command is requesting a current limit that is outside the maximum value allowed on that voltage range.

Programming the `VOLTage:RANGe` by itself causes the programmed current limit to be set to the maximum for the given range if it had previously been higher than the maximum setting for the new range. If the commands

*VOLTage:RANGe 150*

*CURRENT 10*

*VOLTage:RANGe 300*

are sent, no error will be generated because the second `VOLTage:RANGe` command automatically sets the programmed current limit to 5, which is the maximum value for the programmed voltage range.

Programming both the current and the voltage range in one program message unit can be done in any order and will not cause an error if the final combination specifies a valid current limit for the indicated range. If the commands

*VOLTage:RANGe 300*

*CURRENT 10;:VOLTage:RANGe 150*

are sent, no error will be generated because the combined current limit and voltage range specified on the second line are within the output ratings of above models.

## 5.4 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

The following transient modes can be generated:

Step	generates a single triggered output change.
Pulse	generates an output change which returns to its original state after some time period.
List	generates a sequence of output changes, each with an associated dwell time or paced by triggers.
Fixed	turns off the transient functions, which means that only the IMMEDIATE values are used as the data source for a particular function.

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



### 5.4.1 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.

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

If there are no further pulses, the immediate value remains in effect. In List mode, the functions return to their immediate value at the completion of the list. You can mix FIXED, STEP, PULSE, and LIST modes among most functions. When a trigger is received, each function will react in a manner defined by its mode. However, this is subject to the following limitation to ensure the proper output voltage in all cases:

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

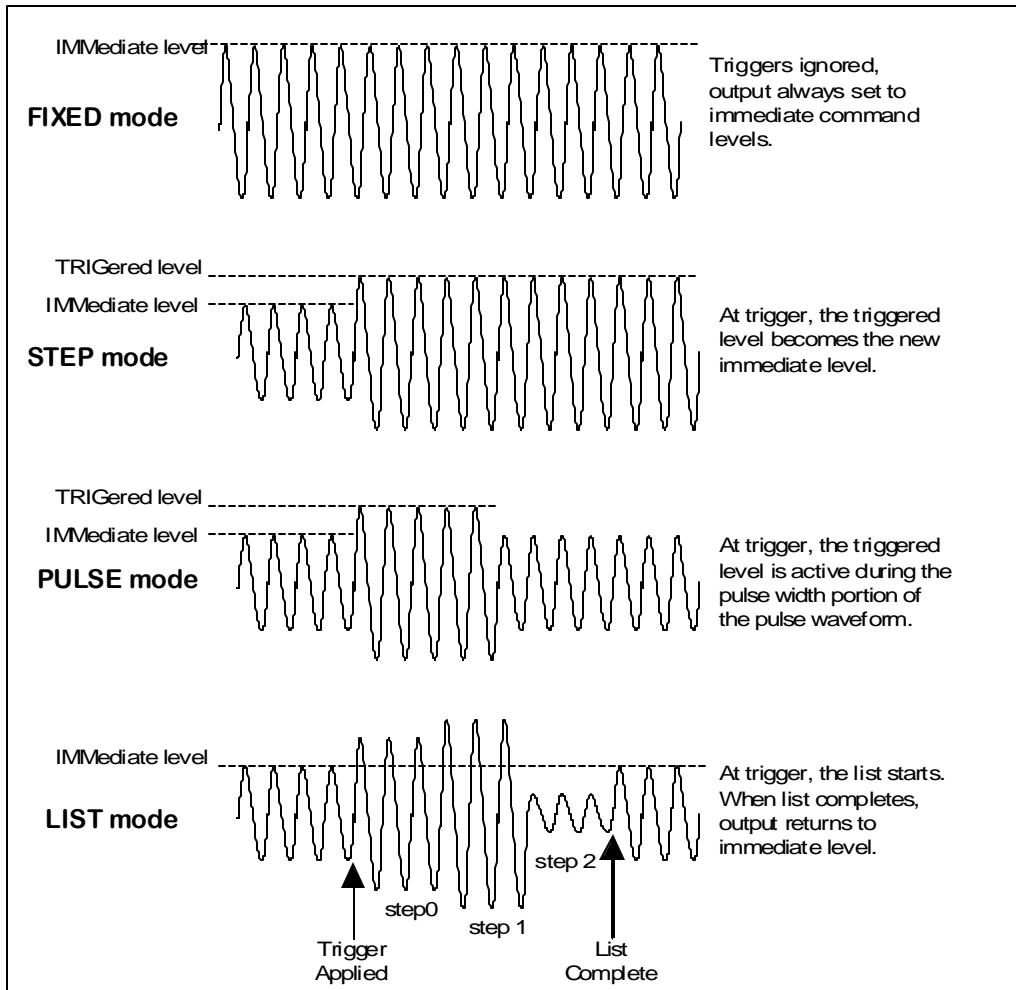


Figure 5-1 : Model of Transient System

## 5.5 Step and Pulse Transients

Proceed as follows to setup step and pulse transients:

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

```
*RST
```

```
VOLTage:MODE 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 135 volts upon receipt of a trigger, use:

```
VOLTage:TRIGger 135
```

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".

4. Only perform this step if you have selected PULSE as the transient mode in Step 1. Specify the pulse count, the pulse period, and then either the duty cycle or the pulse width using the following commands:
  - PULSe:COUNT 1 specifies 1 output pulse
  - PULSe:PERiod 1 specifies a pulse period of 1 second
  - PULSe:DCYCLE 50 specifies a duty cycle of 50%
  - PULSe:WIDTH .5 specifies a pulse width of .5 seconds (not necessary in this case since a duty cycle has already been specified)
5. Initiate the transient trigger system to enable it to receive a trigger. To enable the trigger system for one transient event use:

```
INITiate:IMMEDIATE:SEQUENCE1
```

6. To enable the transient system indefinitely use:

```
INITiate:CONTinuous:SEQUENCE1 ON
```

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

### Example

The following example programs a voltage dropout for 2 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 IEEE-488 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 IEEE-488 bus triggers
TRIG:SYNC:SOUR PHAS	Synchronize triggers to internal phase reference
TRIG:SYNC:PHAS 90	Sets internal phase reference point to 90 degrees
INIT:SEQ1	Set to Wait-for-trigger state
<device trigger>	Send the IEEE-488 bus trigger

## 5.6 List Transients

List mode lets you generate complex sequences of output changes with rapid, precise timing, which may be synchronized with internal or external signals. Each function that can participate in output transients can also have an associated list of values that specify its output at each list point.

You can program up to 100 settings (or points) in the list, the time interval (dwell) that each setting is maintained, the number of times that the list will be executed, and how the settings change in response to triggers.

All list point data is stored in nonvolatile memory. This means that the programmed data for any list function will be retained when the AC source is turned off.

Lists are paced by a separate list of dwell times which define the duration of each output setting. Therefore, each of the up to 100 list points has an associated dwell time, which specifies the time (in seconds) that the output remain at that setting before moving on to the next setting.

The following procedure shows how to generate a simple list of voltage and frequency changes:

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

*VOLTage:MODE LIST*

*FREQuency:MODE LIST*

2. Program the list of output values for each function. The list commands take a comma-separated list of arguments. The order in which the arguments are given determines the sequence in which the values will be output. For example, to cycle the voltage through a sequence 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 110, 120, 130;FREQuency 60*

is the same as:

*LIST:VOLTage 110, 120, 130*

*LIST:FREQuency 60, 60, 60*

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:DWELI 1, 1.5, 2, 2.5, 3*

The number of dwell points must equal the number of output points. If a dwell list has only one value, that value will be applied to all points in the output list.

4. Determine the number of times the list is repeated before it completes. For example, to repeat a list 10 times use:

*LIST:COUNT 10*

Entering INFinity makes the list repeat indefinitely. At \*RST, the count is set to 1.

5. Determines how the list sequencing responds to triggers. For a closely controlled sequence of output levels, you can use a dwell-paced list. To cause the list to be paced by dwell time use:

*LIST:STEP AUTO*

As each dwell time elapses, the next point is immediately output. This is the \*RST setting.

If you need the output to closely follow asynchronous events, then a trigger-paced list is more appropriate. In a trigger-paced list, the list advances one point for each trigger received. To enable trigger-paced lists use:

*LIST:STEP ONCE*

The dwell time associated with each point determines the minimum time that the output remains at that point. If a trigger is received before the previous dwell time completes, the trigger is ignored. Therefore, to ensure that no triggers are lost, program the dwell time to zero.

6. Use the transient trigger system to trigger the list. This is described in detail under "Triggering Output Changes" (Section 5.7).

## 5.7 Triggering Output Changes

---

The AC source has two independent trigger systems. One is used for generating output changes, and the other is used for triggering measurements. This section describes the output trigger system. The measurement trigger system is described under "Triggering Measurements".

The basic components of both systems are the same, but the transient trigger system has additional delay and phase synchronization features that the measurement trigger system does not have.

The following transient trigger sources can be selected:

IMMEDIATE	generates a trigger when the trigger system is initiated.
BUS	selects IEEE-488 bus triggers.
EXTERNAL	selects the external Trigger In BNC connector.

### 5.7.1 SCPI Triggering Nomenclature

---

In SCPI terms, trigger systems are called sequences. When more than one trigger system exists, they are differentiated by naming them SEQUENCE1, SEQUENCE2, ... etc. In the AC source, SEQUENCE1 is the transient trigger system, SEQUENCE2 is the phase synchronization trigger system, and SEQUENCE3 is the measurement trigger system. The AC source uses aliases with more descriptive names for these sequences. These aliases can be used instead of the sequence forms.

<u>Sequence Form</u>	<u>Alias</u>
SEQUENCE1	TRANSient
SEQUENCE2	SYNChronize
SEQUENCE3	ACQUIRE

### 5.7.2 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.

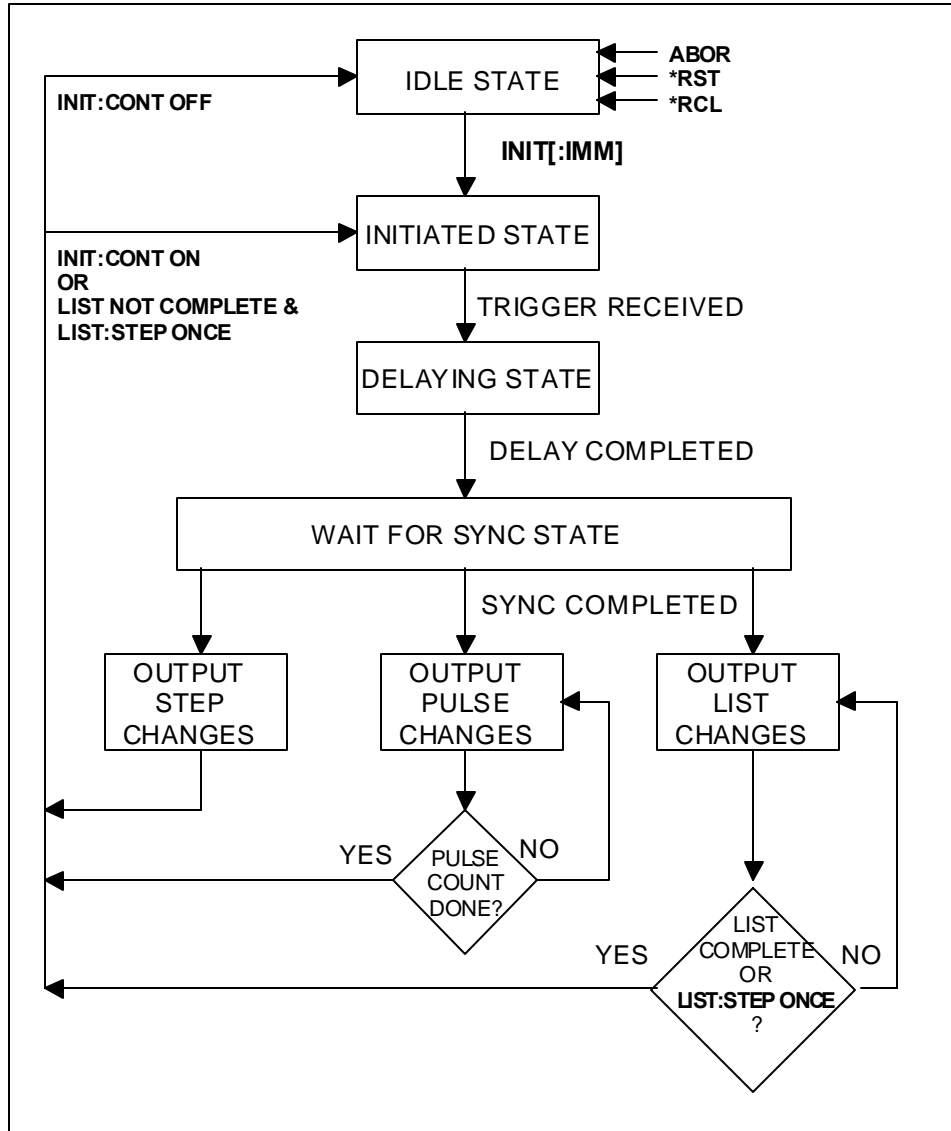


Figure 5-2 : Model of Output Triggers

---

### 5.7.3 Initiating the Output Trigger System

---

When the AC source is turned on, the trigger subsystem is in the idle state. In this state, the trigger subsystem ignores all triggers. Sending the following commands at any time returns the trigger system to the Idle state:

*ABORt \*RST \*RCL*

The INITiate commands move the trigger system from the Idle state to the Initiated state. This enables the AC source to receive triggers. To initiate for a single triggered action, use:

*INITiate:IMMEDIATE:SEQUENCE1* or

*INITiate:IMMEDIATE:NAME TRANSient*

After a trigger is received and the action completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered action is desired. To keep a trigger system initiated for multiple actions without having to send an initiate command for each trigger, use:

*INITiate:CONTinuous:SEQUENCE1 ON* or

*INITiate:CONTinuous:NAME TRANSient, ON*

The SEQUENCE2 (or SYNChronize) trigger sequence does not have an INITiate command. It is always initiated.

---

### 5.7.4 Selecting the Output Trigger Source

---

The trigger system is waiting for a trigger signal in the Initiated state. Before you generate a trigger, you must select a trigger source. To select the external Trigger In BNC as the source, use:

*TRIGger:SEQUENCE1:SOURce EXTernal* or

*TRIGger:TRANSient:SOURce EXTernal*

To select IEEE-488 bus triggers (group execute trigger, device trigger, or \*TRG command), use:

*TRIGger:SEQUENCE1:SOURce BUS* or

*TRIGger:TRANSient:SOURce BUS*

To select a trigger source that is always true, use:

*TRIGger:SEQUENCE1:SOURce IMMEDIATE* or

*TRIGger:TRANSient:SOURce IMMEDIATE*

The immediate source can be combined with INITiate:CONTinuous:SEQUENCE1 ON to generate repetitive output transients. A transition from the Initiated state to the Delay state is made when the trigger signal is received.

### 5.7.5 Specifying a Trigger Delay

---

A time delay can be programmed between the receipt of the trigger signal and the start of the output transient. At \*RST the trigger delay is set to 0, which means that there is no delay. To program a delay, use:

*TRIGger:SEQuence1:DELAy .01* or

*TRIGger:TRANsient:DELAy .01*

which sets a delay time of 10 milliseconds. A trigger delay can only be programmed for SEQuence1 (or TRANsient) triggers. When the programmed trigger delay has elapsed, the trigger system transitions from the Delay state to the Wait-for-sync state.

### 5.7.6 Synchronizing Output Changes to a Reference Phase Angle

---

An output transient normally occurs immediately when the trigger signal is received, or after the delay has expired if a trigger delay has been set. For some applications it is desirable that the transient is synchronized with a particular phase of the output waveform such as the zero crossing point or the positive peak.

To synchronize the start of a transient with a particular phase of the internal phase reference, you must select PHASE as the trigger source. Use:

*TRIGger:SEQuence2:SOURce PHASe* or

*TRIGger:SYNChronize:SOURce PHASe*

To select the desired phase, use:

*TRIGger:SEQuence2:PHASe 90* or

*TRIGger:SYNChronize:PHASe 90*

which specifies the 90 degree phase angle of the internal phase reference as the point where the transient begins.

To turn off transient phase synchronization, use:

*TRIGger:SYNChronous:SOURce IMMEDIATE*

When IMMEDIATE is selected, the trigger system transitions through the Delaying and Wait-for-sync states and goes directly to the Output state. This is the parameter selected at \*RST.

### 5.7.7 Generating Triggers

---

Providing that you have specified the appropriate trigger source, you can generate triggers as follows:

#### Single Triggers

- By sending one of the following over the IEEE-488:
  - *TRIGger:IMMEDIATE*
  - \*TRG
  - a group execute trigger
- By applying a signal with a high-to-low transition to the Trig 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 IEEE-488:

*TRIGger:SEQuence1:SOURce IMMEDIATE*

*INITiate:CONTinuous:SEQuence1 ON*

When the trigger system enters the Output Change state upon receipt of a trigger (see Figure 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.

---

**5.7.8 Specifying a Dwell Time for Each List Point**

---

Each voltage and current list point has an associated dwell time specified by:

*LIST:DWELI <n> {,<n>}*

where <n> specifies the dwell time in seconds. The number of dwell points must equal the number of output points. If a dwell list has only one value, that value will be applied to all points in the output list. After each new output level or point is programmed, the output remains at that point in the list for the programmed dwell interval before the list advances to the next point. Only an **ABORt** command can transfer the system out of the Dwelling state.

At the end of the dwell interval, the transition to the next state depends on whether or not the list has completed its sequencing and the state of the **LIST:STEP** command (see Figure 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.

## 5.8 Making Measurements

---

The AC source has the capability to return a number of current, voltage, and power measurements. When the AC source is turned on, it is continuously sampling the instantaneous output voltage and current for several output cycles and writing the results into a buffer. The buffer holds 4096 voltage and current data points.

The AC source uses the data from the voltage and current buffer to calculate the requested measurement information. Data in the voltage and current buffers is always re-acquired for subsequent measurement requests. There are two ways to make measurements:

- Use the MEASure commands to immediately start acquiring new voltage and current data, and return measurement calculations from this data as soon as the buffer is full. This is the easiest way to make measurements, since it requires no explicit trigger programming.
- Use an acquisition trigger to acquire the voltage and current data from the buffer. Then use the FETCh commands to return calculations from the data that was retrieved by the acquisition trigger. This method gives you the flexibility to synchronize the data acquisition with an external signal. FETCh commands do not trigger the acquisition of new measurement data, but they can be used to return many different calculations from the data that was retrieved by the acquisition trigger.

Making triggered measurements with the acquisition trigger system is discussed under "Triggering Measurements".

---

**Note** For each MEASure form of the query, there is a corresponding query that begins with the header FETCh. FETCh queries perform the same calculation as their MEASure counterparts, but do not cause new data to be acquired. Data acquired by an explicit trigger or a previously programmed MEASure command are used.

---

### 5.8.1 Voltage and Current Measurements

---

To read the sum of ac and dc components of the rms voltage or current, use:

*MEASure:VOLTagE:ACDC?* or  
*MEASure:CURRent:ACDC?*

To measure the dc voltage or current components, use:

*MEASure:VOLTagE:DC?* or  
*MEASure:CURRent:DC?*

To measure the maximum current amplitude and the current crest factor, use:

*MEASure:CURRent:AMPLitude:MAXimum?*  
*MEASure:CURRent:CREStfactor?*

## 5.8.2 Power Measurements

The MEASure and FETCh queries can return real, apparent, and reactive power measurements as well as dc power and power factor using the following commands:

<i>MEASure:POWer:AC:APParent?</i>	measures the ac component of apparent power in VA
<i>MEASure:POWer:AC:REACtive?</i>	measures the reactive power
<i>MEASure:POWer:AC:REAL?</i>	measures the in-phase component of power in watts
<i>MEASure:POWer:AC:PFACtor?</i>	returns the output power factor
<i>MEASure:POWer:AC:TOTal?</i>	measures the total real power being sourced
<i>MEASure:POWer:DC?</i>	measures the dc component of power

## 5.8.3 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:VOLTag:e:HARMonic:AMPLitude? <harmonic number>
MEASure:VOLTag:e: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:VOLTag:e:HARMonic:AMPLitude?
MEASure:ARRay:VOLTag:e: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:VOLTag:e:HARMonic:THD?

```

### 5.8.4 Simultaneous Output Phase Measurements

---

You can return simultaneous measurements from all output phases of the source in the three phase mode using the FETCh query. Unlike MEASure queries, FETCh queries do not trigger the acquisition of new data when they are executed. First, you must initiate the measurement trigger system and generate a measurement trigger as explained in the following section "Triggering Measurements". When the measurement data has been acquired by the voltage and current data buffers for each output phase, use INSTRument:NSElect to select each phase, and FETCh to return the specified measurement data. The following commands return rms voltage:

```
INSTRument:NSElect 1
```

```
FETCh:VOLTagE:AC?
```

```
INSTRument:NSElect 2
```

```
FETCh:VOLTagE:AC?
```

```
INSTRument:NSElect 3
```

```
FETCh:VOLTagE:AC?
```

### 5.8.5 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:

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

```
MEASure:ARRay:VOLTagE?
```

### 5.8.6 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 IEEE-488 bus triggers.
EXTernal	selects the external Trigger In BNC connector.
TTLTrg	selects the signal driving the Trigger Out BNC connector.

### 5.8.7 SCPI Measurement Triggering Nomenclature

As previously explained under "Triggering Output Changes", the AC source uses the following sequence name and alias for the measurement trigger system. This alias can be used instead of the sequence form.

Sequence Form	Alias
SEquence3	ACQuire

### 5.8.8 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.

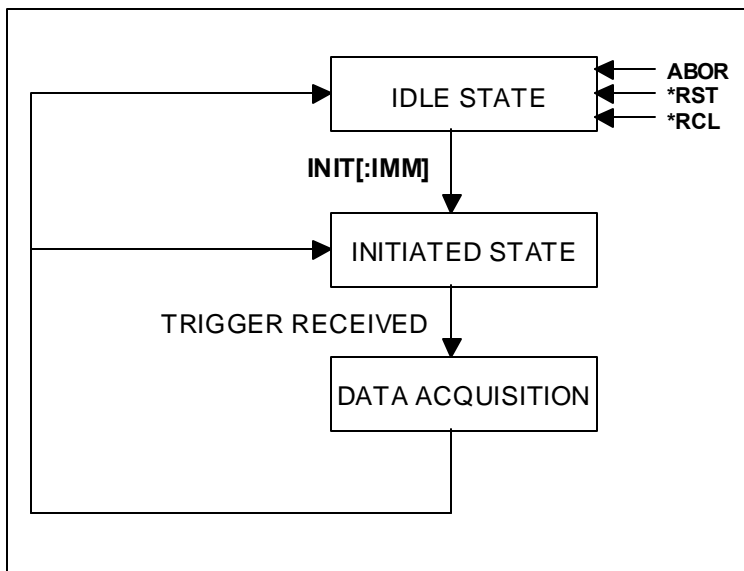


Figure 5-3 : Model of Measurement Triggers

### 5.8.9 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.

---

### 5.8.10 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 IEEE-488 bus triggers (group execute trigger, device trigger, or \*TRG command), use:

```
TRIGger:SEquence3:SOURce BUS  or  
TRIGger:ACQuire:SOURce BUS
```

To select the signal driving the Trigger Out BNC connector, use:

```
TRIGger:SEquence3:SOURce TTLTrg  or  
TRIGger:ACQuire:SOURce TTLTrg
```

### 5.8.11 Generating Measurement Triggers

---

Providing that you have specified the appropriate trigger source, you can generate triggers as follows:

- By sending one of the following over the IEEE-488:
  - TRIGger:SEquence3:IMMediate
  - TRIGger:ACQuire:IMMediate
  - \*TRG
  - a group execute trigger
- By applying a signal with a high-to-low transition to the Trig 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.

### 5.8.12 Using the DFI Output to Indicate Error Conditions

---

The Discrete Fault Indicator output on the rear of the iL Series unit can be used to provide a control or status signal to an external device. The following example illustrates the use of the DFI output to signal a current limit fault condition.

The DFI output is the floating collector and emitter of an opto isolator. As such, these have to be terminated somehow. One possible way to terminate this signal is to tie FLT high to INH high and FLT low to INH low; this works as long as you are not using the INH inputs. Next, the OCP detector has to be activated by sending the following command:

```
CURR:PROT:STATE ON
```

To enable the DFI circuit, send:

```
OUTP:DFI:SOURCE QUES
```

```
OUTP:DFI:STATE ON
```

To enable the OCP signal to get to the STAT:QUES register, send:

```
STAT:QUES:PTR 2
```

```
STAT:QUES:ENAB 2
```

After sending these commands, the DFI output will indicate when the RMS OCP goes active. Note that when an OCP fault happens, the PROT lamp comes on indicating that one has to send OUTP:PROT:CLEAR to re-enable the source once the over current condition is removed. Note also that the DFI output will stay low until the questionable status register (STAT:QUES?) is read.

## **5.9 Controlling the Instantaneous Voltage and Current Data Buffers**

---

Measurements taken by the iL series measurement systems are based on a digital representation of the voltage and current waveforms. Both waveforms are digitized with 16 bits of amplitude resolution and down to 25  $\mu$ sec of time resolution. (40 Ks/sec real-time sampling rate). The available memory depth to hold this information is 4K (4096 samples) for each waveform. This section covers the SCPI commands that can be used to control the sampling rate and the position of the trigger point in the 4K deep data buffer.

### **5.9.1 Varying the Voltage and Current Sampling Rate**

---

At \*RST, the output voltage and current sampling rate is 40 kHz (period = 25  $\mu$ sec). This means that it takes about 100 milliseconds to fill up 4096 data points in the voltage and current data buffers with the information required to make a measurement calculation. You can vary this data sampling rate with:

```
SENSe:SWEep:TINterval <sample period>
```

The sample period can be programmed from a minimum period of 25 microseconds (the default), to 250 microseconds in 25 microsecond steps.

## 5.9.2 Pre-event and Post-event Triggering

The range for this offset is 4096 to  $2E9$  points. As shown in the following figure, when the offset is negative, the values at the beginning of the data record represent samples taken prior to the trigger. When the value is 0, all of the values are taken after the trigger. Values greater than zero can be used to program a delay time from the receipt of the trigger until the data points that are entered into the buffer are valid. (Delay time = Offset  $\times$  Sample period)

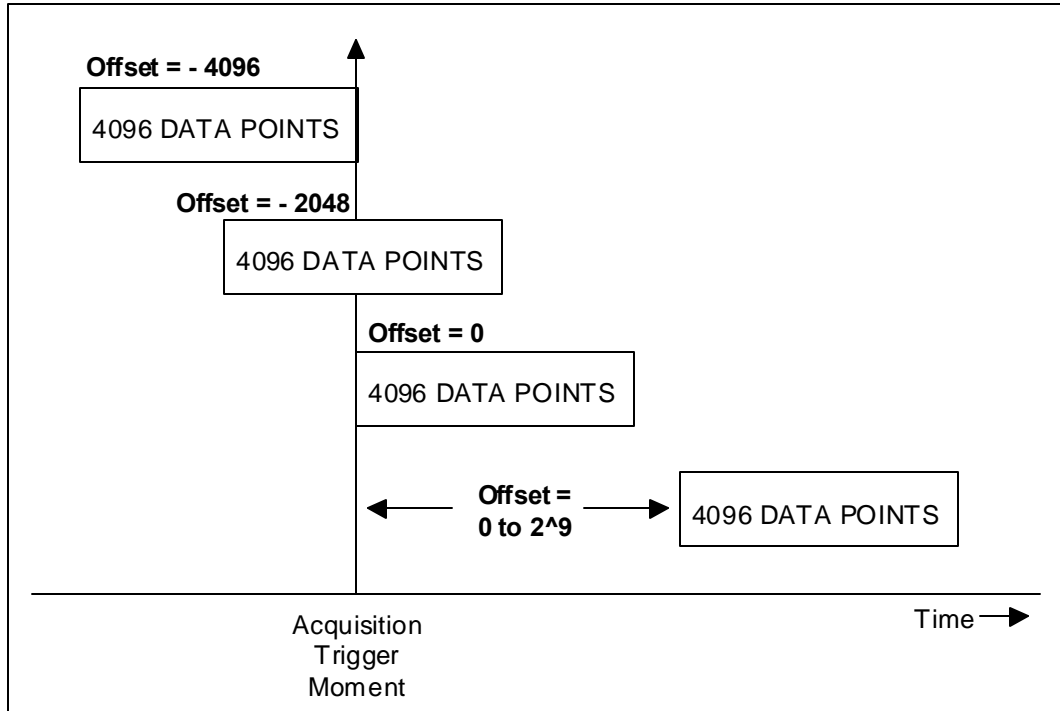


Figure 5-4 : Pre-event and Post-event Triggering

## 5.9.3 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 power source has a command which alters both the output frequency control and the harmonic measurement algorithms to meet these requirements. The command is

SENSe:WINDow KBESsel | RECTangular



## 6. Programming the Status and Event Registers

You can use status register programming to determine the operating condition of the AC source at any time. For example, you may program the AC source to generate an interrupt (assert SRQ) when an event such as a current limit occurs. When the interrupt occurs, your program can then act on the event in the appropriate fashion.

Figure 6-1 shows the status register structure of the AC Source. Table 6-2 defines the status bits. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard IEEE-488 functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The Operation Status, Questionable Status, and Questionable Instrument Summary Status registers implement functions that are specific to the AC source.

### 6.1 Power-On Conditions

Refer to the \*RST command description in chapter 4, Table 4-7 for the power-on conditions of the status registers.

### 6.2 Operation Status Group

The Operation Status registers record signals that occur during normal operation. The group consists of the following registers:

Register	Command	Description
Condition	STAT:OPER:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:OPER:PTR <n>	A positive transition filter that functions as described under STAT:OPER:NTR PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:OPER:NTR <n>	A negative transition filter that functions as described under STAT:OPER:NTR PTR commands in chapter 4. It is a read/write register.
Event	STAT:OPER:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:OPER:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

**Table 6-1 : Operation Status registers**

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

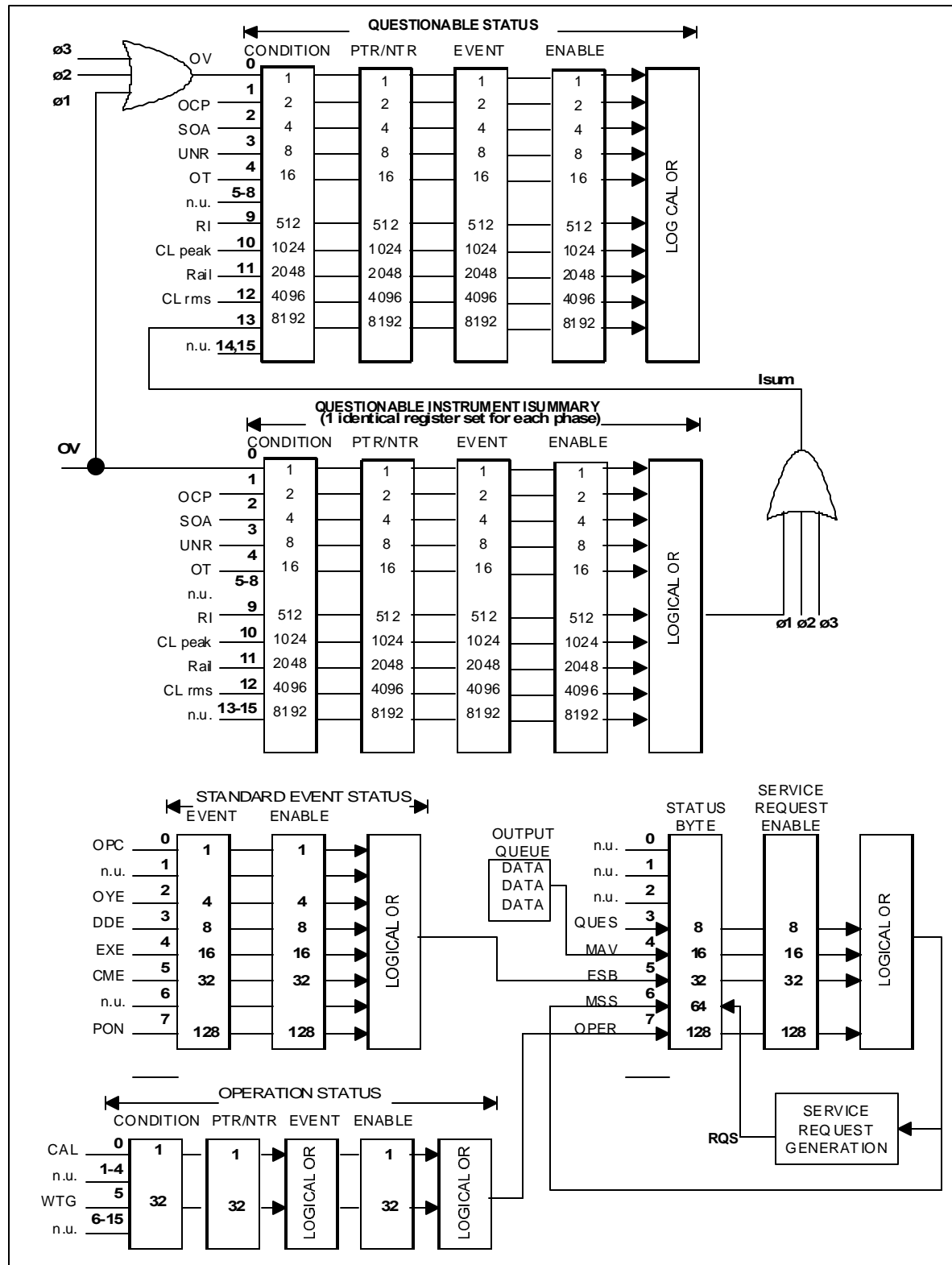


Figure 6-1 : AC Source Status Model

Bit	Signal	Meaning
<b>Operation Status Group</b>		
0	CAL	Interface is computing new cal constants
5	WTG	Interface is waiting for a trigger
8	CV	The output voltage is regulated
<b>Questionable and Questionable Instrument Summary Status Groups</b>		
0	OV	The overvoltage protection circuit has tripped
1	OCP	The overcurrent protection circuit has tripped
3	UNR	The output is unregulated
4	TO	An overtemperature condition has occurred
9	RI	The remote inhibit state is active
11	Rail	Loss of input phase voltage detected
12	CLrms	The rms current limit circuit is active
13	Isum	Summary of QUES:INST:ISUM registers
<b>Standard Event Status Group</b>		
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
<b>Status Byte and Service Request Enable Registers</b>		
3	QUES	Questionable status summary bit
4	MAV	Message Available summary bit
5	ESB	Event Status Summary bit
6	MSS	Master Status Summary bit
	RQS	Request Service bit
7	OPER	Operation status summary bit

Table 6-2 : Bit Configurations of Status Registers

### 6.3 Questionable Status Group

The Questionable Status registers record signals that indicate abnormal operation of the AC source. As shown in Figure 6-1, the group consists of the same type of registers as the Status Operation group.

Register	Command	Description
Condition	STAT:QUES:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:QUES:PTR <n>	A positive transition filter that functions as described under STAT:QUES:NTR PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:QUES:NTR <n>	A negative transition filter that functions as described under STAT:QUES:NTR PTR commands in chapter 4. It is a read/write register.
Event	STAT:QUES:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:QUES:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

**Table 6-3 : Questionable Status registers**

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

---

**Note:** If the AC source is in the three-phase mode, each signal that is fed into the Questionable Status Condition register is logically-ORed from three corresponding status signals that originate from each phase. Figure 6-1 illustrates this for the OV bit; the same illustration also applies to the other bits in the Condition register.

---

## 6.4 Questionable Instrument Isummary Status Group

Although only one group of Questionable Instrument Isummary Status registers is shown in Figure 6-1, there are actually three identical register groups. With the exception of bit 13, the register structure is the same as the Questionable Status group. These three register groups monitor the status signals of each individual phase of the three-phase AC source. To determine which phase of the AC source is currently selected, use:

*INSTrument:NSElect?*

To set or read the status registers of another phase, first use:

*INSTrument:NSElect <n>*

where <n> is the phase number. Then send the appropriate register commands.

Register	Command	Description
Condition	STAT:QUES:INST:I SUM:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:QUES:INST:I SUM:PTR <n>	A positive transition filter that functions as described under STAT:QUES:INST:ISUM:NTR PTR commands in chapter 4. It is a read/write register.
NTR Filter	STAT:QUES:INST:I SUM:NTR <n>	A negative transition filter that functions as described under STAT:QUES:INST:ISUM:NTR PTR commands in chapter 4. It is a read/write register.
Event	STAT:QUES:INST:I SUM:EVENT?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:QUES:INST:I SUM:ENAB <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

**Table 6-4 : Questionable Instrument Isummary Status registers**

The outputs of the Questionable Instrument Isummary Status group are logically-ORed into the Isum bit (13) of the Questionable Condition register.

## 6.5 Standard Event Status Group

---

This group consists of an Event register and an Enable register that are programmed by Common commands. The Standard Event register latches events relating to interface communication status (see Figure 6-1). It is a read-only register that is cleared when read.

The Standard Event Enable register functions similarly to the enable registers of the Operation and Questionable status groups.

<b>Command</b>	<b>Action</b>
*ESE	programs specific bits in the Standard Event Enable register.
*PSC ON	clears the Standard Event Enable register at power-on.
*ESR?	reads and clears the Standard Event register.

### The PON (Power On) Bit

The PON bit in the Standard Event register is set whenever the AC source is turned on. The most common use for PON is to generate an SRQ at power-on following an unexpected loss of power. To do this, bit 7 of the Standard Event Enable register must be set so that a power-on event registers in the ESB (Standard Event Summary Bit). Bit 5 of the Service Request Enable register must be set to permit an SRQ to be generated, and \*PSC OFF must be sent. The commands to accomplish these conditions are:

*\*PSC OFF*

*\*ESE 128*

*\*SRE 32*

## 6.6 Status Byte Register

This register summarizes the information from all other status groups as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The bit configuration is shown in Table 4-8.

<b>Command</b>	<b>Action</b>
*STB?	reads the data in the register but does not clear it (returns MSS in bit 6)
serial poll	reads and clears the data in the register (returns RQS in bit 6)

### The MSS Bit

This is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the AC source has one or more reasons for requesting service. \*STB? reads the MSS in bit position 6 of the response but does not clear any of the bits in the Status Byte register.

### The RQS Bit

The RQS bit is a latched version of the MSS bit. Whenever the AC source requests service, it sets the SRQ interrupt line true and latches RQS into bit 6 of the Status Byte register. When the controller does a serial poll, RQS is cleared inside the register and returned in bit position 6 of the response. The remaining bits of the Status Byte register are not disturbed.

### The MAV bit and Output Queue

The Output Queue is a first-in, first-out (FIFO) data register that stores AC source-to-controller messages until the controller reads them. Whenever the queue holds one or more bytes, it sets the MAV bit (bit 4) of the Status byte register.

## 6.7 Examples

---

### 6.7.1 Determining the Cause of a Service Interrupt

---

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

1. Determine which summary bits are active. Use  
*\*STB?* or  
serial poll
2. Read the corresponding Event register for each summary bit to determine which events caused the summary bit to be set. Use:  
*STATus:QUESTionable:EVENT?*  
*STATus:OPERation:EVENT?*  
*ESR?*

---

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

---

3. Remove the specific condition that caused the event. If this is not possible, the event may be disabled by programming the corresponding bit of the status group Enable register or NTR|PTR filter. A faster way to prevent the interrupt is to disable the service request by programming the appropriate bit of the Service Request Enable register.

### 6.7.2 Servicing Questionable Status Events

---

This example assumes you want a service request generated whenever the AC source's overvoltage, overcurrent, or overtemperature circuits have tripped. From Figure 6-1, note the required path for Questionable Status conditions at bits 0, 1, and 4 to generate a service request (RQS) at the Status Byte register. The required register programming is as follows:

1. Program the Questionable Status PTR register to allow a positive transition at bits 0, 1, or 4 to be latched into the Status Event register. Use:  
*STATus:QUESTionable:PTR 19* (1 + 2 + 16 = 19)
2. Program the Questionable Status Enable register to allow the latched events to be summed into the QUES summary bit. Use:  
*STATus:QUESTionable:ENABle 19*
3. Program the Service Request Enable register to allow the QUES summary bit from the Status Byte register to generate RQS. Use:  
*\*SRE 8*
4. When you service the request, read the event register to determine which Questionable Status Event register bits are set and clear the register for the next event. Use:  
*STATus:QUESTionable:EVENT?*



### 6.7.3 Monitoring Both Phases of a Status Transition

You can monitor a status signal for both its positive and negative transitions. For example, to generate RQS when the AC source either enters the CLrms (rms current limit) condition or leaves that condition, program the Questionable Status PTR/NTR filter as follows:

```
STATus:QUEStionable:PTR 4096;NTR 4096
```

```
STATus:QUEStionable:ENABle 4096;*SRE 8
```

The PTR filter will cause the QUES summary bit to set RQS when CLrms occurs. When the controller subsequently reads the event register with STATus:QUEStionable:EVEN?, the register is cleared. When CLrms subsequently goes false, the NTR filter causes the QUES summary bit to again set RQS.

### 6.7.4 Programming the Trigger In and Trigger Out BNC Connectors

The AC source has two BNC connectors labeled Trigger In and Trigger Out (see Figure 6-2 : BNC Connector Trigger Model). Refer to "Typical Specifications" in Chapter 5 of the User's Manual 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 TTLT* or

*TRIGger:ACQuire:SOURce TTLT*

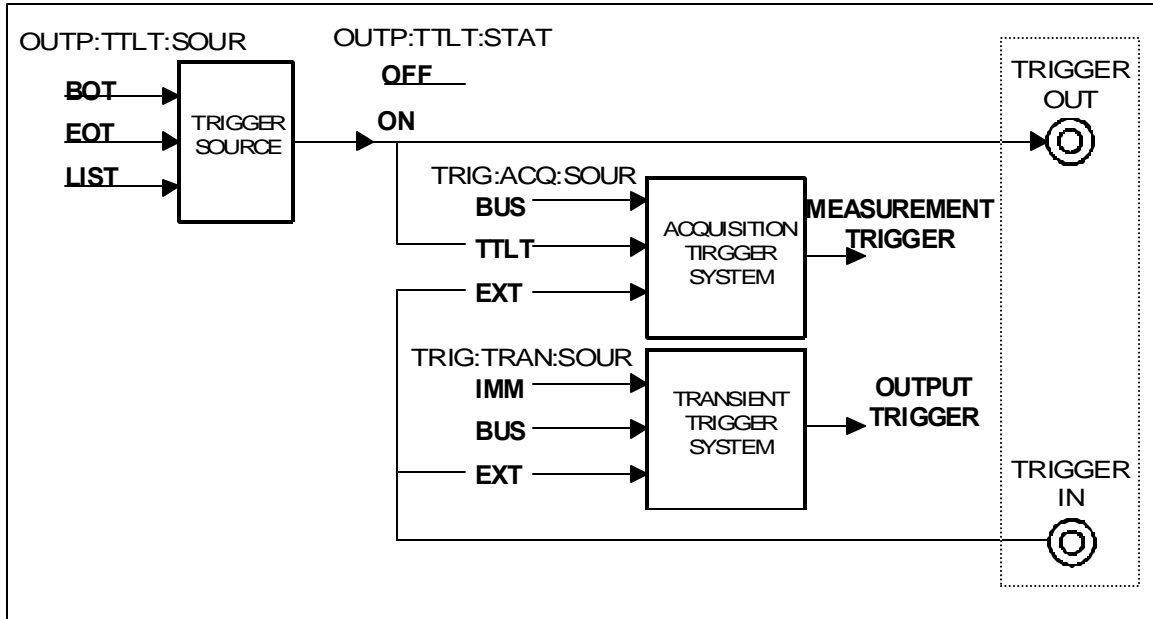


Figure 6-2 : BNC Connector Trigger Model

## 6.8 Remote Inhibit and Discrete Fault Indicator

---

The remote inhibit and discrete fault indicators are implemented through the respective INH and FLT connections on the rear panel. Refer to "Operating Characteristics" in appendix A of the User's Manual for the electrical parameters.

### 6.8.1 Remote Inhibit (RI)

---

Remote inhibit is an external, chassis-referenced logic signal routed through the rear panel INH connection, which allows an external device to signal a fault.

To select an operating modes for the remote inhibit signal, use:

*OUTPut:RI:MODE LATCHing | LIVE | OFF*

LATCHing	causes a low-true signal on the INH input to disable the output. The only way to clear the latch is by sending an <i>OUTPut:PROTEction:CLEAR</i> 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.

### 6.8.2 Discrete Fault Indicator (DFI)

---

The discrete fault indicator is a chassis-referenced, open-collector logic signal connected to the rear panel FLT connection, that can be used to signal external devices when a fault condition is detected. To select the internal fault source that drives this signal, use:

*OUTPut:DFI:SOURce QUESTIONable | OPERATION | ESB | RQS | OFF*

QUESTIONable	selects the Questionable event summary bit (bit 3 of the Status Byte Register)
OPERation	selects the Operation Event summary bit (bit 7 of the Status Byte Register)
ESB	selects the Standard Event summary bit (bit 5 of the Status Byte Register)
RQS	selects the Request Service bit (bit 6 of the Status Byte Register)
OFF	selects no DFI source

To enable or disable the DFI output, use:

*OUTPut:DFI:STATe ON | OFF*

## 6.9 SCPI Command Completion

---

SCPI commands sent to the AC source are processed either sequentially or in parallel. Sequential commands finish execution before a subsequent command begins. Parallel commands allow other commands to begin executing while the parallel command is still executing. Commands that affect list and trigger actions are among the parallel commands.

The \*WAI, \*OPC, and \*OPC? common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. The syntax and parameters for these commands are described in chapter 4. Some practical considerations for using these commands are as follows:

- |       |   |
|-------|---|
| *WAI  | This prevents the AC source from processing subsequent commands until all pending operations are completed.   |
| *OPC? | This places a 1 in the Output Queue when all pending operations have completed. Because it requires your program to read the returned value before executing the next program statement, *OPC? can be used to cause the controller to wait for commands to complete before proceeding with its program. |
| *OPC  | This sets the OPC status bit when all pending operations have completed. Since your program can read this status bit on an interrupt basis, *OPC allows subsequent commands to be executed.   |

## Appendix A : SCPI Command tree

### SCPI Common Commands

\*CLS  
 \*ESE <value>  
 \*ESE?  
 \*ESR?  
 \*IDN?  
 \*OPC  
 \*OPC?  
 \*OPT?  
 \*PSC <bool>  
 \*PSC?  
 \*RCL <value>  
 \*RST  
 \*SAV <value>  
 \*SRE <value>  
 \*SRE?  
 \*STB?  
 \*TRG  
 \*TST?  
 \*WAI

### ABORt

### CALibrate

:CURRent  
 :AC  
 :MEAS  
 :DATA <n>  
 :IMPedance  
 :LEVel P1 | P2 | P3 | P4  
 :PASSword <n>  
 :SAVE  
 :STATe <bool>  
 :VOLTage  
 :AC  
 :PROT  
 :RTIME

### DATA | TRACe

:CATalog?  
 [:DATA] <trace\_name>,<n>{,<n>}  
 :DEFine <trace\_name>[,<trace\_name> | 1024]  
 :DELete  
 [:NAME] <trace\_name>

### DIAGnostic

:CURRent?  
 :EEPROM <n>,<n>,<n>[,INTeger | FLOat]  
 :FAN:MODE AUTO | MANual  
 :FAN:SPEEd:MANual <n>  
 :FAN:SPEEd:AUTO?  
 :NOP  
 :RESet  
 :RWLongform <n>  
 :TEMPerature:AMBient?

### DISPlay

[:WINDow]  
 [:STATe] <bool>  
 :MODE NORMal | TEXT  
 :TEXT  
 [:DATA] <display\_string>

### INITiate

[:IMMEDIATE]  
 :SEQUence[1 | 3]  
 :NAME TRANsient | ACQUIRE  
 :CONTinuous  
 :SEQUence[1] <bool>  
 :NAME TRANsient, <bool>

### INSTrument

:COUPle ALL | NONE  
 :NSElect 1 | 2 | 3  
 :SElect OUTPut1 | OUTPut2 | OUTPut 3

### FETCH | MEASure

[:SCALar]  
 :CURRent  
 [:DC]?  
 :AC?  
 :ACDC?  
 :AMPLitude  
 :MAX?  
 :CREStfactor?  
 :HARMonic  
 [:AMPLitude]? <n>  
 :PHASe? <n>

```

        :THD?
    :NEUTral
        [:DC]?
        :AC?
        :ACDC?
        :HARMonic
FETCh | MEASure[:SCALar] (continued)
        [:AMPLitude]? <n>
        :PHASe? <n>
    :FREQuency?
    :POWer
        [:DC]?
        :AC
            [:REAL]?
            :APParent?
            :REACtive?
            :PFACtor?
            :TOTal?
    :VOLTagE
        [:DC]?
        :AC?
        :ACDC?
        :HARMonic
            [:AMPLitude]? <n>
            :PHASe? <n>
            :THD?
    :ARRAy
        :CURRent
            [:DC]?
            :HARMonic
                [:AMPLitude]? <n>
                :PHASe? <n>
            :NEUTral
                [:DC]?
                :HARMonic
                    [:AMPLitude]? <n>
                    :PHASe? <n>
        :VOLTagE
            [:DC]?
            :HARMonic
                [:AMPLitude]? <n>
                :PHASe? <n>
OUTPut
        [:STATe] <bool>
        :DFI
            [:STATe] <bool>
            :SOURce QUES | OPER | ESB | RQS | OFF
        :IMPedance
            [:STATe] <bool>
            :REAL <n>
            :REACtive <n>
        :PON
            :STATe RST | RCL0
        :PROTection
            :CLEar
            :DElay <n>
        :RI
            :MODE LATCHing | LIVE | OFF
        TTLTrig
            [:STATe] <bool>
            :SOURce BOT | EOT | LIST
SENSe
        :CURRent
            :ACDC
                :RANGe
                    [:UPPer] <n>
        :SWEep
            :OFFSet
                :POINts <n>
            :TINTerval <n>
        :WINDow
            [:TYPE] KBESsel | RECTangular
[SOURce:]
    CURRent
        [:LEVel]
            [:IMMEdiate]
                [:AMPLitude] <n>
        :PROTection
            :STATe <bool>
    FREQuency
        [:CW | :IMMEdiate] <n>
        :MODE FIXed | STEP | PULSe | LIST
        :SLEW
            [:IMMEdiate] <n> | INFinity
            :MODE FISed | STEP | PULSe | LIST
            :TRIGgered <n> | INFinity
            :TRIGgered <n>
    FUNCTion
        [:SHAPE]
            [:IMMEdiate] SINusoid | SQUARE | CSINusoid |
            <user>
            :MODE FIXed | STEP | PULSe | LIST

```

```

:TRIGgered SINusoid | SQUare | CSINusoid |
<user>
:CSINusoid <n> [THD]
LIST
:COUNt <n> | INFIinity
:CURRent <n>{,<n>}
:POINTs?
:DWELl <n>{,<n>}
:POINTs?
:FREQuency
[:LEVel] <n>{,<n>}
:POINTs?
:SLEW <n>{,<n>}
:POINTs?

[SOURCE:]LIST (continued)
:PHASe <n>{,<n>}
:POINTs?
:SHAPE <shape>{,<shape>}
:POINTs?
:STEP ONCE | AUTO
:TTLTrg <bool>{,<bool>}
:VOLTage
[:LEVel] <n>{,<n>}
:POINTs?
:SLEW <n>{,<n>}
:POINTs?

PHASe
[:IMMediate | ADJust] <n>
:MODE FIXed | STEP | PULSe | LIST
:TRIGgered <n>

PULSe
:COUNt <n> | INFIinity
:DCYCLE <n>
:HOLD WIDTH | DCYCLE
:PERiod <n>
:WIDTH <n>

VOLTage
[:LEVel]
[:IMMediate]
[:AMPLitude] <n>
:TRIGgered
[:AMPLitude] <n>
:SENSe | :ALC
:DETEctor RTIME | RMS
:SOURce INT | EXT

:MODE FIXed | STEP | PULSe | LIST
:PROTEction
[:LEVel] <n>
:RANGe 150 | 300
:SLEW
[:IMMediate] <n> | INFIinity
:MODE FIXed | STEP | PULSe | LIST
:TRIGgered <n> | INFIinity

STATus
:OPERation
[:EVENT]?
:CONDition?
:ENABle <n>
:NTRansition <n>
:PTRansition <n>
:PRESet
:QUEStionable
[:EVENT]?
:CONDition?
:ENABle <n>
:INSTrument
:ISUMmary
[:EVENT]?
:CONDition?
:ENABle <n>
:NTRansition <n>
:PTRansition <n>
:NTRansition <n>
:PTRansition <n>

SYSTEM
:CONFIguration
:NOUtputs <n>
:ERRor?
:VERSion?
:LANGuage SCPI | E9012
:LOCal
:REMote
:RWLOCK

TRIGger
[:TRANsient | SEQuence1]
[:IMMediate]
:SOURce BUS | EXTErnal | IMMEDIATE
DELAy <n>
:SYNChronize | :SEQuence2
:SOURce PHASe | IMMEDIATE

```

:PHASe <n>	:SEquence2
:ACQuire   :SEquence3	:DEFine SYNChronize
[:IMMediate]	:SEquence3
:SOURce BUS   EXTernal   IMMediate	:DEFine ACQuire
:SEquence1	
:DEFine TRANsient	



## Appendix B : SCPI Conformance Information

### SCPI Version

The power source conforms to SCPI version 1992.0.

### SCPI Confirmed commands

ABOR	[SOUR:]LIST:DWEL:POIN?
CAL:DATA	[SOUR:]LIST:FREQ[:LEV]
CAL:STAT	[SOUR:]LIST:FREQ[:LEV]:POIN?
DISP[:WIND][:STAT]	[SOUR:]LIST:VOLT[:LEV]
DISP[:WIND]:TEXT[:DATA]	[SOUR:]LIST:VOLT[:LEV]:POIN?
INIT[:IMM]	[SOUR:]PHAS[:IMM]
INIT[:IMM]:SEQ or NAME	[SOUR:]PULS:COUN
INIT:CONT:SEQ or NAME	[SOUR:]PULS:DCYC
INST:COUP	[SOUR:]PULS:HOLD
INST:NSEL	[SOUR:]PULS:PER
MEAS or FETC[:SCAL]:CURR[:DC]?	[SOUR:]PULS:WIDT
MEAS or FETC[:SCAL]:CURR:AC?	[SOUR:]VOLT:ALC or SENS:SOUR
MEAS or FETC[:SCAL]:FREQ?	[SOUR:]VOLT[:LEV][:IMM][:AMPL]
MEAS or FETC[:SCAL]:POW[:DC]?	[SOUR:]VOLT[:LEV][:TRIG][:AMPL]
MEAS or FETC[:SCAL]:POW:AC[:REAL]?	[SOUR:]VOLT:MODE
MEAS or FETC[:SCAL]:VOLT:AC?	[SOUR:]VOLT:PROT[:AMPL]
MEAS or FETC:ARR:CURR[:DC]?	[SOUR:]VOLT:RANG
MEAS or FETC:ARR:VOLT[:DC]?	[SOUR:]VOLT:SLEW[:IMM]
OUTP[:STAT]	[SOUR:]VOLT:PROT:STAT
OUTP:COUP	[SOUR:]VOLT:ALC or SENS:SOUR
OUTP:PROT:CLE	STAT:OPER[:EVEN]?
OUTP:PROT:DEL	STAT:OPER:COND?
OUTP:TTLT[:STAT]	STAT:OPER:ENAB?
SENSe:WIND	STAT:OPER:NTR
SENSe:CURR:ACDC:RANG[:UPP]	STAT:OPER:PTR
SENS:SWE:OFFS:POIN	STAT:PRES
SENS:SWE:TINT	STAT:QUES:
[SOUR:]CURR[:LEV][:IMM][:AMPL]	STAT:QUES[:EVEN]?
[SOUR:]CURR:PROT:STAT	STAT:QUES:COND?
[SOUR:]FREQ[:CW or :IMM]	STAT:QUES:ENAB?
[SOUR:]FREQ:MODE	STAT:QUES:NTR
[SOUR:]FUNC[:SHAP][:IMM]	STAT:QUES:PTR
[SOUR:]LIST:COUN	STAT:QUES:INST:ISUM[:EVEN]?
[SOUR:]LIST:CURR	STAT:QUES:INST:ISUM:COND?
[SOUR:]LIST:CURR:POIN?	STAT:QUES:INST:ISUM:ENAB
[SOUR:]LIST:DWEL	STAT:QUES:INST:ISUM:NTR

STAT:QUES:INST:ISUM:PTR	*CLS
SYST:ERR?	*ESE *ESE? *ESR?
SYST:VERS?	*IND?
TRAC or DATA:CAT?	*OPC *OPC? *OPT?
TRAC or DATA[:DATA]	*PSC *PSC?
TRAC or DATA:DEF	*RCL *RST
TRAC or DATA:DEL[:NAME]	*SAV *SRE *STB?
TRIG:SEQ<n>:DEL	*TRG *TST?
TRIG:SEQ<n>[:IMM]	*WAI
TRIG:SEQ<n>:SOUR	
TRIG:SOUR	

**Non-SCPI commands.**

CAL:CURR:AC	MEAS or FETC[:SCAL]:VOLT:HARM:PHAS?
CAL:LEV	MEAS or FETC[:SCAL]:VOLT:HARM:THD?
CAL:PASS	MEAS or FETC:ARR:CURR:HARM[:AMPL]?
CAL:SAVE	MEAS or FETC:ARR:CURR:HARM:PHAS?
CAL:VOLT:AC	MEAS or FETC:ARR:CURR:NEUT[:DC]?
DISP[:WIND]:MODE	MEAS or
MEAS or FETC[:SCAL]:CURR:ACDC?	FETC:ARR:CURR:NEUT:HARM[:AMPL]?
MEAS or FETC[:SCAL]:CURR:AMPL:MAX?	MEAS or
MEAS or FETC[:SCAL]:CURR:CRES?	FETC:ARR:CURR:NEUT:HARM:PHAS?
MEAS or	MEAS or FETC:ARR:VOLT:HARM[:AMPL]?
FETC[:SCAL]:CURR:HARM[:AMPL]?	MEAS or FETC:ARR:VOLT:HARM:PHAS?
MEAS or FETC[:SCAL]:CURR:HARM:PHAS?	OUTP:DFI[:STAT]
MEAS or FETC[:SCAL]:CURR:HARM:THD?	OUTP:DFI:SOUR
MEAS or FETC[:SCAL]:CURR:NEUT[:DC]?	OUTP:IMP:REAL
MEAS or FETC[:SCAL]:CURR:NEUT:AC?	OUTP:IMP:REAC
MEAS or FETC[:SCAL]:CURR:NEUT:DC?	OUTP:RI:MODE
MEAS or FETC[:SCAL]:CURR:NEUT:ACDC?	OUTP:TTLT:SOUR
MEAS or	[SOUR:]CURR:PEAK[:IMM]
FETC[:SCAL]:CURR:NEUT:HARM[:AMPL]?	[SOUR:]CURR:PEAK:MODE
MEAS or	[SOUR:]CURR:PEAK:TRIG
FETC[:SCAL]:CURR:NEUR:HARM:PHAS?	[SOUR:]FREQ:SLEW[:IMM]
MEAS or FETC[:SCAL]:POW:AC:APP?	[SOUR:]FREQ:SLEW:MODE
MEAS or FETC[:SCAL]:POW:AC:REAC?	[SOUR:]FREQ:SLEW:TRIG
MEAS or FETC[:SCAL]:POW:AC:PFAC?	[SOUR:]FREQ:TRIG
MEAS or FETC[:SCAL]:POW:AC:TOT	[SOUR:]FUNC[:SHAP]:CSIN
MEAS or FETC[:SCAL]:VOLT:ACDC?	[SOUR:]FUNC[:SHAP]:MODE
MEAS or	[SOUR:]FUNC[:SHAP]:TRIG
FETC[:SCAL]:VOLT:HARM[:AMPL]?	[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:OFFS  
[SOUR:]LIST:VOLT:OFFS:POIN?  
[SOUR:]LIST:VOLT:SLEW  
[SOUR:]LIST:VOLT:SLEW:POIN?  
[SOUR:]PHAS:MODE  
[SOUR:]PHAS:TRIG  
[SOUR:]VOLT:ALC or SENS:DET  
[SOUR:]VOLT:OFFS[:IMM]  
[SOUR:]VOLT:OFFS:MODE  
[SOUR:]VOLT:OFFS:TRIG  
[SOUR:]VOLT:SLEW:MODE  
[SOUR:]VOLT:SLEW:TRIG  
SYST:LOC  
SYST:REM  
SYST:RWL  
TRIG:SEQ2 or SYNC:PHAS  
TRIG:SEQ:DEF



## Appendix C : Error Messages

This appendix gives the error numbers and descriptions that are returned by the AC power source. Error numbers are returned in two ways:

- Error numbers are displayed on the front panel
- Error numbers and messages are read back with the “SYSTem:ERRor?” query. SYSTem:ERRor? returns the error number into a variable and returns two parameters: an NR1 and a string.

When errors occur, the Standard Event Status register records them as shown in the following table.

Bit Set	Code Error	Type
5	-100 thru -199	Command
4	-200 thru -299	Execution
3	-300 thru -399 or 1 thru 32767	Device-dependent
2	-400 thru -499	Query

### Standard Event Status register error bits

The following table lists the errors that are associated with SCPI syntax errors and interface problems. It also lists the device dependent errors. Information inside the brackets is not part of the standard error message, but is included for clarification.

<b>Error</b>	<b>Error String</b>
-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
-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
-200	Execution error [generic]
-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

<b>Error</b>	<b>Error String</b>
-310	System error
-350	Too many errors [errors beyond 9 lost due to queue overflow]
-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]
0	No error
1	Non-volatile RAM RDO 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 STATE section checksum failed
10	RAM selftest
11	DAC selftest 1
12	DAC selftest 2
13	DAC selftest 3
14	DAC selftest 4
15	DAC selftest 5
16	DAC selftest 6
17	DAC selftest 7
18	DAC selftest 8
40	Voltage selftest error, output 1
41	Voltage selftest error, output 2
42	Voltage selftest error, output 3
43	Current selftest error, output 1
44	Current selftest error, output 2
45	Current selftest error, output 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

<b>Error</b>	<b>Error String</b>
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 [refer to AC source Service Manual]
402	CAL passcode is incorrect
403	CAL not enabled
404	Computed readback cal constants are incorrect
405	Computed programming cal constants are incorrect
406	Incorrect sequence of calibration commands
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:SHAPE modes incompatible

**SCPI error codes and messages**



## Index

*		<b>D</b>	
*CLS .....	4-91	DATA .....	4-81
*ESE .....	4-91	DATA:CATalog.....	4-82
*ESR? .....	4-92	DATA:DEFine.....	4-82
*IDN? .....	4-92	DATA:DELete.....	4-82
*OPC.....	4-93	DIAGnostic:CURRent? .....	4-7
*OPT?.....	4-94	DIAGnostic:EEPRom.....	4-7
*PSC .....	4-94	DISPlay .....	4-10
*RCL.....	4-94	DISPlay:MODE.....	4-10
*RST .....	4-95	DISPlay:TEXT .....	4-11
*SAV .....	4-96	DOS Drivers .....	3-1
*SRE .....	4-96		
*STB? .....	4-97	<b>E</b>	
*TRG.....	4-97	Error Handling .....	3-2
*TST? .....	4-97		
*WAI.....	4-98	<b>F</b>	
<b>A</b>		FETCh:ARRay:CURRent:HARMonic:PHASe?4-16	
ABORt .....	4-84	FETCh:ARRay:CURRent:HARMonic?.....	4-15
address		FETCh:ARRay:CURRent:NEUTral:HARMonic:PH	
IEEE.....	1-2	ASe? .....	4-17
<b>C</b>		FETCh:ARRay:CURRent:NEUTral:HARMonic? .4-	
CALCulate:INTegral:TIME .....	4-100	17	
CALCulate:SMOothing .....	4-101	FETCh:ARRay:CURRent:NEUTral? .....	4-16
CALibrate:CURRent:AC.....	4-3	FETCh:ARRay:CURRent? .....	4-15
CALibrate:CURRent:MEASure .....	4-3	FETCh:ARRay:VOLTage:HARMonic? .....	4-18
CALibrate:DATA .....	4-4	FETCh:ARRay:VOLTage?.....	4-18
CALibrate:IMPedance .....	4-4	FETCh:CURRent:AC? .....	4-20
CALibrate:LEVel.....	4-4	FETCh:CURRent:ACDC?.....	4-20
CALibrate:PASSword .....	4-5	FETCh:CURRent:AMPLitude:MAXimum? .....	4-20
CALibrate:SAVE.....	4-5	FETCh:CURRent:CREStfactor? .....	4-21
CALibrate:STATe.....	4-5	FETCh:CURRent:HARMonic:PHASe?.....	4-22
CALibrate:VOLTage:AC.....	4-6	FETCh:CURRent:HARMonic:THD? .....	4-22
CALibrate:VOLTage:PROTection.....	4-6	FETCh:CURRent:HARMonic?.....	4-21
CALibrate:VOLTage:RTIME.....	4-6	FETCh:CURRent:NEUTral:AC? .....	4-23
command		FETCh:CURRent:NEUTral:ACDC? .....	4-23
subsystem .....	4-2	FETCh:CURRent:NEUTral:HARMonic:PHASe? .4-	
commands		24	
common.....	2-3	FETCh:CURRent:NEUTral:HARMonic? .....	4-24
coupled .....	2-4	FETCh:CURRent:NEUTral? .....	4-23
Conventions .....	2-1	FETCh:CURRent? .....	4-19
Copyright .....	ii	FETCh:FREQuency?.....	4-25
coupling .....	2-4	FETCh:POWer:AC:APParent? .....	4-27
CURRent .....	4-40	FETCh:POWer:AC:PFACtor? .....	4-28
CURRent:PROTection:STATe.....	4-41	FETCh:POWer:AC:REACTive? .....	4-27
		FETCh:POWer:AC:TOTAL?.....	4-28
		FETCh:POWer:AC? .....	4-27
		FETCh:POWer? .....	4-26
		FETCh:VOLTage:AC? .....	4-30

FETCh:VOLTagE:ACDC? ..... 4-30  
 FETCh:VOLTagE:HARMonic:PHASe? ..... 4-31  
 FETCh:VOLTagE:HARMonic:THD? ..... 4-31  
 FETCh:VOLTagE:HARMonic? ..... 4-30  
 FETCh:VOLTagE? ..... 4-29  
 FORMat ..... 4-102  
 FORMat:BORDER ..... 4-103  
 formats  
     data ..... 2-7  
 FREQuency:SLEW:TRIGgered ..... 4-45  
 FREQuency ..... 4-42  
 FREQuency:MODE ..... 4-43  
 FREQuency:SLEW ..... 4-44  
 FREQuency:SLEW:MODE ..... 4-44  
 FREQuency:TRIGgered ..... 4-45  
 FUNction ..... 4-46  
 FUNction:CSINusoid ..... 4-48  
 FUNction:MODE ..... 4-47  
 FUNction:TRIGgered ..... 4-47

**G**

GW-BASIC ..... 3-1

**H**

header  
     optional ..... 2-6  
     separator ..... 2-6  
**Header** ..... 2-2  
**headers**  
     **optional** ..... 2-3  
 Headers ..... 2-5  
 HP 82335A  
     driver ..... 3-1  
 HP82335A ..... 3-1

**I**

IEC555 Commands ..... 4-99  
 IEEE ..... 1-1  
     setting address ..... 1-2  
 IEEE-488  
     controller cards ..... 3-1  
**Initialization** ..... 5-2  
 INITiate:CONTInuous:NAME ..... 4-85  
 INITiate:CONTInuous:SEQuence ..... 4-85  
 INITiate:NAME ..... 4-84  
 INITiate:SEQuence ..... 4-84  
 INSTRument:COUple ..... 4-12  
 INSTRument:NSElect ..... 4-13  
 INSTRument:SElect ..... 4-13

**L**

LIST:COUNT ..... 4-50  
 LIST:DWELl ..... 4-50

LIST:DWELl:POINts? ..... 4-51  
 LIST:FREQuency ..... 4-51  
 LIST:FREQuency:POINts? ..... 4-51  
 LIST:FREQuency:SLEW ..... 4-52  
 LIST:FREQuency:SLEW:POINts? ..... 4-52  
 LIST:PHASe ..... 4-52  
 LIST:PHASe:POINts? ..... 4-53  
 LIST:SHAPe ..... 4-53  
 LIST:SHAPe:POINts? ..... 4-54  
 LIST:STEP ..... 4-54  
 LIST:TTLTrg ..... 4-54  
 LIST:TTLTrg:POINts? ..... 4-55  
 LIST:VOLTagE ..... 4-55  
 LIST:VOLTagE:POINts? ..... 4-55  
 LIST:VOLTagE:SLEW ..... 4-56  
 LIST:VOLTagE:SLEW:POINts? ..... 4-56

**M**

MEASure:ARRay:VOLTagE:FLUCtuatiions:FLICKer? ..... 4-107  
 MEASure:ARRay:CURREnt:HARMonic:PHASe? ..... 4-16  
 MEASure:ARRay:CURREnt:HARMonic?... 4-15, 4-103  
 MEASure:ARRay:CURREnt:NEUTral:HARMonic:PHASe? ..... 4-17  
 MEASure:ARRay:CURREnt:NEUTral:HARMonic? ..... 4-17  
 MEASure:ARRay:CURREnt:NEUTral?" ..... 4-16  
 MEASure:ARRay:CURREnt? ..... 4-15  
 MEASure:ARRay:VOLTagE:FLUCtuations:ALL? ..... 4-104  
 MEASure:ARRay:VOLTagE:FLUCtuations:PST?4-108  
 MEASure:ARRay:VOLTagE:HARMonic? ..... 4-18  
 MEASure:ARRay:VOLTagE? ..... 4-18  
 MEASure:CURREnt:AC? ..... 4-20  
 MEASure:CURREnt:ACDC? ..... 4-20  
 MEASure:CURREnt:AMPLitude:MAXimum? 4-20  
 MEASure:CURREnt:CREStfactor? ..... 4-21  
 MEASure:CURREnt:HARMonic:PHASe? ..... 4-22  
 MEASure:CURREnt:HARMonic:THD? ..... 4-22  
 MEASure:CURREnt:HARMonic? ..... 4-21  
 MEASure:CURREnt:NEUTral:AC? ..... 4-23  
 MEASure:CURREnt:NEUTral:ACDC? ..... 4-23  
 MEASure:CURREnt:NEUTral:HARMonic:PHASe? ..... 4-24  
 MEASure:CURREnt:NEUTral:HARMonic? ..... 4-24  
 MEASure:CURREnt:NEUTral? ..... 4-23  
 MEASure:CURREnt? ..... 4-19  
 MEASure:FREQuency? ..... 4-25  
 MEASure:POWer:AC:APParent? ..... 4-27  
 MEASure:POWer:AC:PFACtor? ..... 4-28

MEASure:POWer:AC:REACtive?.....	4-27
MEASure:POWer:AC:TOTal? .....	4-28
MEASure:POWer:AC? .....	4-27
MEASure:POWer? .....	4-26
MEASure:VOLTag:e:AC? .....	4-30
MEASure:VOLTag:e:ACDC?.....	4-30
MEASure:VOLTag:e:HARMonic:PHASe?.....	4-31
MEASure:VOLTag:e:HARMonic:THD? .....	4-31
MEASure:VOLTag:e:HARMonic?.....	4-30
MEASure:VOLTag:e? .....	4-29

**N**

## National Instruments

PC-IIA .....	3-1
NI PC-IIA	
driver .....	3-2
error-handling .....	3-2

**O**

OUTPut .....	4-32
OUTPut:DFL.....	4-33
OUTPut:DFI:SOURce.....	4-33
OUTPut:IMP:REACtive.....	4-34
OUTPut:IMP:REAL.....	4-34
OUTPut:IMPedance .....	4-33
OUTPut:PON:STATe.....	4-34
OUTPut:PROTection:CLear.....	4-35
OUTPut:PROTection:DELay.....	4-35
OUTPut:RI:MODE.....	4-35
OUTPut:TTLTrg .....	4-36
OUTPut:TTLTrg:SOURce .....	4-36

**P**

PHASe.....	4-57
PHASe:MODE.....	4-58
PHASe:TRIGgered.....	4-58
programming.....	3-1
programming commands	
calculate limit upper .....	4-101
PULSe:COUNT.....	4-59
PULSe:DCYCLe .....	4-59
PULSe:HOLD.....	4-60
PULSe:PERiod.....	4-61
PULSe:WIDTh .....	4-61

**Q**

queries .....	2-4
---------------	-----

**R**

root.....	2-6
<b>Root</b> .....	<b>2-2</b>
root specifier.....	2-6

**S**

<b>Safety</b> .....	<b>iii</b>
SCPI.....	2-1
Command Tree .....	2-2
message.....	2-2
message unit .....	2-4
program message.....	2-2
response message .....	2-2
structure .....	2-4
SENSe:CURRent:ACDC:RANGe .....	4-37
SENSe:CURRent:PREFeRence .....	4-109
SENSe:SWEEp:OFFSet:POINts .....	4-38
SENSe:SWEEp:TINTErval.....	4-38
SENSe:WINDow .....	4-39
Separator .....	2-6
STATus:OPERation:CONDition? .....	4-70
STATus:OPERation:ENABle .....	4-71
STATus:OPERation:NTR	
STATus:OPERation:PTR.....	4-71
STATus:OPERation?.....	4-70
STATus:PRESet.....	4-70
STATus:QUEStionable:CONDition? .....	4-72
STATus:QUEStionable:ENABle.....	4-73
STATus:QUEStionable:INSTrument:ISUMmary:C	
ONDition? .....	4-75
STATus:QUEStionable:INSTrument:ISUMmary:E	
NABle .....	4-76
STATus:QUEStionable:INSTrument:ISUMmary:N	
TR.....	4-77
STATus:QUEStionable:INSTrument:ISUMmary:P	
TR.....	4-77
STATus:QUEStionable:INSTrument:ISUMmary?4-	
74	
STATus:QUEStionable:NTR	
STATus:QUEStionable:PTR.....	4-73
STATus:QUEStionable?.....	4-72
<b>SYSTEM:CONFIgure:NOUtputs</b> .....	<b>4-78</b>
SYSTEM:CONFIgure .....	4-109
SYSTEM:ERRor? .....	4-79
SYSTEM:LANGUage .....	4-79
SYSTEM:LOCAl.....	4-80
SYSTEM:REMote.....	4-80
SYSTEM:RWLock.....	4-80
SYSTEM:VERSion?.....	4-79

**T**

<b>Table of Contents</b> .....	<b>iv</b>
Table of Figures .....	vi
Table of Tables.....	vi
terminator .....	2-6
TRACe .....	4-81
TRACe:CATalog.....	4-82
TRACe:DEFine.....	4-82

TRACe:DELeTe .....4-82  
 TRIGger.....4-85  
 TRIGger:ACQuire .....4-88  
 TRIGger:ACQuire:SOURce .....4-88  
 TRIGger:DELAy.....4-86  
 TRIGger:SEQuence1:DEFine .....4-89  
 TRIGger:SEQuence2:DEFine .....4-89  
 TRIGger:SEQuence2:PHASe .....4-87  
 TRIGger:SEQuence2:SOURce .....4-87  
 TRIGger:SEQuence3 .....4-88  
 TRIGger:SEQuence3:DEFine .....4-89  
 TRIGger:SEQuence3:SOURce .....4-88  
 TRIGger:SOURce.....4-86  
 TRIGger:SYNChronize:PHASe .....4-87  
 TRIGger:SYNChronize:SOURce .....4-87

*U*

User's Manual .....1-1

*V*

VOLTage.....4-62  
 VOLTage:ALC:DETEctor.....4-66  
 VOLTage:MODE.....4-63  
 VOLTage:PROTEction .....4-65  
 VOLTage:RANGe .....4-66  
 VOLTage:SENSe:DETEctor.....4-66  
 VOLTage:SENSe:SOURce VOLTage:ALC:SOURce  
 .....4-67  
 VOLTage:SLEW .....4-67  
 VOLTage:SLEW:MODE.....4-68  
 VOLTage:SLEW:TRIGgered .....4-68  
 VOLTage:TRIGgered.....4-63