Keysight Models 6811B, 6812B, 6813B, 6814B, 6834B, and 6843A AC Power Solutions

Programming Guide

KEYSIGHT TECHNOLOGIES

NOTICE: This document contains references to Agilent Technologies. Agilent's former Test and Measurement business has become Keysight Technologies. For more information, go to **www.keysight.com.**



Safety Summary

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

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

Printing History

The edition and current revision of this manual are indicated below. Reprints of this manual containing minor corrections and updates may have the same printing date. Revised editions are identified by a new printing date. A revised edition incorporates all new or corrected material since the previous printing date. Changes to the manual occurring between revisions are covered by change sheets shipped with the manual.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior consent of Keysight Technologies. The information contained in this document is subject to change without notice.

© Copyright 1996 - 2014 Keysight Technologies

Edition 1 _____August, 1996
Edition 2 _____March, 1997
Edition 3 _____December, 1998
Update _____April, 2000
Edition 4 _____December, 2014

Table of Contents

Safety Summary **2**
Printing History **2**
Table of Contents **3**

1 - GENERAL INFORMATION **11**
About this Guide **11**
Earlier AC Source Models 11
Documentation Summary 11
External References **12**
SCPI References 12
GPIB References 12
Keysight VXI*plug&play* Power Products Instrument Drivers **12**
Supported Applications 12
System Requirements 13
Downloading and Installing the Driver 13
Accessing Online Help 13

2 - INTRODUCTION TO PROGRAMMING **15**
GPIB Capabilities of the AC Source **15**
GPIB Address 15
RS-232 Capabilities of the AC Source **15**
RS-232 Data Format 15
Baud Rate 15
RS-232 Programming Example 16
RS-232 Troubleshooting 16
Introduction to SCPI **17**
Conventions Used in This Guide 17
Types of SCPI Commands 17
Types of SCPI Messages 17
The SCPI Command Tree **18**
The Root Level 18
Active Header Path 18
The Effect of Optional Headers 19
Moving Among Subsystems 19
Including Common Commands 20
Using Queries 20
Coupled Commands 20
Structure of a SCPI Message **20**
The Message Unit 20
Combining Message Units 21
Headers 21
Query Indicator 22
Message Unit Separator 22
Root Specifier 22
Message Terminator 22
SCPI Data Formats **23**
Numerical Data Formats 23
Suffixes and Multipliers 23
Character Data 23

System Considerations **24**
Assigning the GPIB Address in Programs 24
Types of DOS Drivers 24
Error Handling 25
Keysight BASIC Controllers 25

3 - LANGUAGE DICTIONARY 27

Introduction **27**
Subsystem Commands **28**
Calibration Subsystem Commands **29**
Subsystem Syntax 29
CALibrate:CURRent:AC 29
CALibrate:CURRent:MEASure 30
CALibrate:DATA 30
CALibrate:IMPedance 30
CALibrate:LEVel 30
CALibrate:PASSword 31
CALibrate:PWM:FREQuency 31
CALibrate:PWM:RAMP 31
CALibrate:SAVE 31
CALibrate:STATe 32
CALibrate:VOLTage:AC 32
CALibrate:VOLTage:DC 32
CALibrate:VOLTage:OFFSet 33
CALibrate:VOLTage:PROTection 33
CALibrate:VOLTage:RTIMe 33
Display Subsystem Commands **34**
Subsystem Syntax 34
DISPlay 34
DISPlay:MODE 34
DISPlay:TEXT 34
Instrument Subsystem **35**
Subsystem Syntax 35
INSTrument:COUPle 35
INSTrument:NSELect INSTrument:SELect 36
Measurement Subsystem (Arrays) **37**
Subsystem Syntax 37
MEASure:ARRay:CURRent? FETCh:ARRay:CURRent? 37
MEASure:ARRay:CURRent:HARMonic? FETCh:ARRay:CURRent:HARMonic? 38
MEASure:ARRay:CURRent:HARMonic:PHASe? FETCh:ARRay:CURRent:HARMonic:PHASe? 38
MEASure:ARRay:CURRent:NEUTral? FETCh:ARRay:CURRent:NEUTral? 39
MEASure:ARRay:CURRent:NEUTral:HARMonic? FETCh:ARRay:CURRent:NEUTral:HARMonic? 39
MEASure:ARRay:CURRent:NEUTral:HARMonic:PHASe?
FETCh:ARRay:CURRent:NEUTral:HARMonic:PHASe? 40
MEASure:ARRay:VOLTage? FETCh:ARRay:VOLTage? 40
MEASure:ARRay:VOLTage:HARMonic? FETCh:ARRay:VOLTage:HARMonic? 41
MEASure:ARRay:VOLTage:HARMonic:PHASe? FETCh:ARRay:VOLTage:HARMonic:PHASe? 41
Measurement Subsystem (Current) **42**
Subsystem Syntax 42
MEASure:CURRent? FETCh:CURRent? 42
MEASure:CURRent:AC? FETCh:CURRent:AC? 43
MEASure:CURRent:ACDC? FETCh:CURRent:ACDC? 43
MEASure:CURRent:AMPLitude:MAXimum? FETCh:CURRent:AMPLitude:MAXimum? 43
MEASure:CURRent:CREStfactor? FETCh:CURRent:CREStfactor? 44

MEASure:CURRent:HARMonic? FETCh:CURRent:HARMonic? 44
MEASure:CURRent:HARMonic:PHASe? FETCh:CURRent:HARMonic:PHASe? 45
MEASure:CURRent:HARMonic:THD? FETCh:CURRent:HARMonic:THD? 45
MEASure:CURRent:NEUTral? FETCh:CURRent:NEUTral? 45
MEASure:CURRent:NEUTral:AC? FETCh:CURRent:NEUTral:AC? 46
MEASure:CURRent:NEUTral:ACDC? FETCh:CURRent:NEUTral:ACDC? 46
MEASure:CURRent:NEUTral:HARMonic? FETCh:CURRent:NEUTral:HARMonic? 46
MEASure:CURRent:NEUTral:HARMonic:PHASe? FETCh:CURRent:NEUTral:HARMonic:PHASe? 47
Measurement Subsystem (Frequency) 48
Subsystem Syntax 48
MEASure:FREQuency? FETCh:FREQuency? 48
Measurement Subsystem (Power) 49
Subsystem Syntax 49
MEASure:POWer? FETCh:POWer? 49
MEASure:POWer:AC? FETCh:POWer:AC? 49
MEASure:POWer:AC:APParent? FETCh:POWer:AC:APParent? 50
MEASure:POWer:AC:REACtive? FETCh:POWer:AC:REACtive? 50
MEASure:POWer:AC:PFACtor? FETCh:POWer:AC:PFACtor? 50
MEASure:POWer:AC:TOTal? FETCh:POWer:AC:TOTal? 51
Measurement Subsystem (Voltage) 52
Subsystem Syntax 52
MEASure:VOLTage? FETCh:VOLTage? 52
MEASure:VOLTage:AC? FETCh:VOLTage:AC? 52
MEASure:VOLTage:ACDC? FETCh:VOLTage:ACDC? 53
MEASure:VOLTage:HARMonic? FETCh:VOLTage:HARMonic? 53
MEASure:VOLTage:HARMonic:PHASe? FETCh:VOLTage:HARMonic:PHASe? 54
MEASure:VOLTage:HARMonic:THD? FETCh:VOLTage:HARMonic:THD? 54
Output Subsystem 55
Subsystem Syntax 55
OUTPut 55
OUTPut:COUPling 56
OUTPut:DFI 56
OUTPut:DFI:SOURce 56
OUTPut:IMPedance 57
OUTPut:IMPedance:REAL 57
OUTPut:IMPedance:REACtive 57
OUTPut:PON:STATe 58
OUTPut:PROTection:CLEar 58
OUTPut:PROTection:DELay 58
OUTPut:RI:MODE 59
OUTPut:TTLTrg 59
OUTPut:TTLTrg:SOURce 59
Sense Subsystem 60
Subsystem Syntax 60
SENSe:CURRent:ACDC:RANGe 60
SENSe:SWEep:OFFSet:POINts 61
SENSe:SWEep:TINTerval 61
SENSe:WINDow 61
Source Subsystem (Current) 62
Subsystem Syntax 62
CURRent 62
CURRent:PEAK 63
CURRent:PEAK:MODE 63
CURRent:PEAK:TRIGgered 64
CURRent:PROTection:STATe 64

Source Subsystem (Frequency) **65**
Subsystem Syntax 65
FREQuency 65
FREQuency:MODE 65
FREQuency:SLEW 66
FREQuency:SLEW:MODE 66
FREQency:SLEW:TRIGgered 66
FREQuency:TRIGgered 67
Source Subsystem (Function) **68**
Subsystem Syntax 68
FUNCtion 68
FUNCtion:MODE 69
FUNCtion:TRIGgered 69
FUNCtion:CSINusoid 70
Source Subsystem (List) **71**
Subsystem Syntax 71
LIST:COUNt 72
LIST:CURRent 72
LIST:CURRent:POINts? 72
LIST:DWELl 73
LIST:DWELl:POINts? 73
LIST:FREQuency 73
LIST:FREQuency:POINts? 74
LIST:FREQuency:SLEW 74
LIST:FREQuency:SLEW:POINts? 74
LIST:PHASe 74
LIST:PHASe:POINts? 75
LIST:SHAPe 75
LIST:SHAPe:POINts? 75
LIST:STEP 76
LIST:TTLTrg 76
LIST:TTLTrg:POINts? 76
LIST:VOLTage 77
LIST:VOLTage:POINts? 77
LIST:VOLTage:SLEW 77
LIST:VOLTage:SLEW:POINts? 78
LIST:VOLTageOFFSet 78
LIST:VOLTage:OFFSet:POINts? 78
LIST:VOLTage:OFFSet:SLEW 79
LIST:VOLTage:OFFSet:SLEW:POINts? 79
Source Subsystem (Phase) **80**
PHASe 80
PHASe:MODE 81
PHASe:TRIGgered 81
Source Subsystem (Pulse) **82**
Subsystem Syntax 82
PULSe:COUNt 82
PULSe:DCYCle 82
PULSe:HOLD 83
PULSe:PERiod 84
PULSe:WIDTh 84

Source Subsystem (Voltage) **85**
Subsystem Syntax 85
VOLTage 86
VOLTage:TRIGgered 86
VOLTage:MODE 87
VOLTage:OFFSet 87
VOLTage:OFFSet:MODE 88
VOLTage:OFFSet:TRIGgered 88
VOLTage:OFFSet:SLEW 89
VOLTage:OFFSet:SLEW:MODE 89
VOLTage:OFFSet:SLEW:TRIGgered 90
VOLTage:PROTection 90
VOLTage:PROTection:STATe 90
VOLTage:RANGe 91
VOLTage:SENSe:DETector VOLTage:ALC:DETector 91
VOLTage:SENSe:SOURce VOLTage:ALC:SOURce 92
VOLTage:SLEW 92
VOLTage:SLEW:MODE 93
VOLTage:SLEW:TRIGgered 93
Status Subsystem **94**
Subsystem Syntax 94
STATus:PRESet 94
Bit Configuration of Operation Status Registers 95
STATus:OPERation? 95
STATus:OPERation:CONDition? 95
STATus:OPERation:ENABle 95
STATus:OPERation:NTRansition STATus:OPERation:PTRansition 96
Bit Configuration of Questionable Status Registers 97
STATus:QUEStionable? 97
STATus:QUEStionable:CONDition? 97
STATus:QUEStionable:ENABle 98
STATus:QUEStionable:NTRansition STATus:QUEStionable:PTRansition 98
Bit Configuration of Questionable Instrument Summary Registers 99
STATus:QUEStionable:INSTrument:ISUMmary? 99
STATus:QUEStionable:INSTrument:ISUMmary:CONDition? 100
STATus:QUEStionable:INSTrument:ISUMmary:ENABle 100
STATus:QUEStionable:INSTrument:ISUMmary:NTR STATus:QUEStionable:INSTrument:ISUMmary:PTR101
System Commands **102**
Subsystem Syntax 102
SYSTem:CONFigure 102
SYSTem:CONFigure:NOUTputs 103
SYSTem:ERRor? 103
SYSTem:VERSion? 103
SYSTem:LANGuage 104
SYSTem:LOCal 104
SYSTem:REMote 104
SYSTem:RWLock 104
Trace Subsystem **105**
Subsystem Syntax 105
TRACe DATA 105
TRACe:CATalog? DATA:CATalog? 106
TRACe:DEFine DATA:DEFine 106
TRACe:DELete DATA:DELete 106

Trigger Subsystem **107**
Subsystem Syntax 107
ABORt 108
INITiate:SEQuence INITiate:NAME 108
INITiate:CONTinuous:SEQuence INITiate:CONTinuous:NAME 109
TRIGger 109
TRIGger:DELay 109
TRIGger:SOURce 110
TRIGger:SEQuence2:SOURce TRIGger:SYNChronize:SOURce 110
TRIGger:SEQuence2:PHASe TRIGger:SYNCHronize:PHASe 111
TRIGger:SEQuence3 TRIGger:ACQuire 111
TRIGger:SEQuence3:SOURce TRIGger:ACQuire:SOURce 112
TRIGger:SEQuence1:DEFine TRIGger:SEQuence2:DEFine TRIGger:SEQuence3:DEFine 112
Common Commands **113**
Common Commands Syntax 113
*CLS 114
*ESE 114
Bit Configuration of Standard Event Status Enable Register 114
*ESR? 115
*IDN? 115
*OPC 115
*OPT? 116
*PSC 116
*RCL 116
*RST 117
*SAV 118
*SRE 118
*STB? 119
Bit Configuration of Status Byte Register 119
*TRG 119
*TST? 119
*WAI 120

4 - PROGRAMMING EXAMPLES **121**
Introduction **121**
Programming the Output **121**
Power-on Initialization 121
Enabling the Output 121
AC Voltage and Frequency 122
Voltage and Frequency Slew Rates 123
Waveform Shapes 123
Individual Phases (Keysight 6834B only) 124
Current Limit 125
DC Output (Keysight 6811B/6812B/6813B only) 126
Coupled Commands 127
Programming Output Transients **128**
Transient System Model 129
Step and Pulse Transients 130
List Transients 130
Triggering Output Changes **132**
SCPI Triggering Nomenclature 132
Output Trigger System Model 132
Initiating the Output Trigger System 134
Selecting the Output Trigger Source 134

Specifying a Trigger Delay 135
Synchronizing Output Changes to a Reference Phase Angle 135
Generating Output Triggers 136
Specifying a Dwell Time for Each List Point 136
Making Measurements 137
Voltage and Current Measurements 137
Power Measurements 138
Harmonic Measurements 138
Simultaneous Output Phase Measurements (Keysight 6834B only) 138
Returning Voltage and Current Data From the Data Buffer 139
Regulatory-Compliant Measurement of Quasi-Stationary Harmonics 139
Triggering Measurements 139
SCPI Triggering Nomenclature 139
Measurement Trigger System Model 139
Initiating the Measurement Trigger System 140
Selecting the Measurement Trigger Source 140
Generating Measurement Triggers 141
Controlling the Instantaneous Voltage and Current Data Buffers 141
Programming the Status Registers 142
Power-On Conditions 142
Operation Status Group 142
Questionable Status Group 144
Questionable Instrument Isummary Status Group 145
Standard Event Status Group 146
Status Byte Register 147
Examples 147
Programming the Trigger In and Trigger Out BNC Connectors 148
Trigger In BNC 148
Trigger Out BNC 149
Remote Inhibit and Discrete Fault Indicator 149
Remote Inhibit (RI) 150
Discrete Fault Indicator (DFI) 150
SCPI Command Completion 150

A - SCPI COMMAND TREE 151
Command Syntax 151

B - SCPI CONFORMANCE INFORMATION 155
SCPI Confirmed Commands 155
Non SCPI Commands 156

C - ERROR MESSAGES 157
Error Number List 157

D - ELGAR MODEL 9012 COMPATIBILITY 161
Elgar Model 9012 Plug-in Programmer Compatibility 161
Main Board W1 Jumper Option Emulation 161
Syntax Compatibility 161
Status Model 162
Power-on State 162
Protection 163
Front Panel Operation 163
System Keys 163

Function Keys 163
Entry Keys 164
E9012 Language Command Summary 164

E - IEC MODE COMMAND SUMMARY 167
Introduction 167
Using the SENSe:CURRent:ACDC:RANGe command 167
Command Syntax 168
CALCulate:INTegral:TIME 169
CALCulate:SMOothing 169
CALCulate:LIMit:UPPer 170
FORMat 171
FORMat:BORDer 172
MEASure:ARRay:CURRent:HARMonic? 173
MEASure:ARRay:VOLTage:FLUCtuations:ALL? 174
MEASure:ARRay:VOLTage:FLUCtuations:FLICker? 176
MEASure:ARRay:VOLTage:FLUCtuations:PST? 177
SENSe:CURRent:PREFerence 178
SENSe:WINDow 178
SYSTem:CONFigure 179

INDEX 181

1

General Information

About this Guide

This manual contains programming information for the Keysight 6811B, 6812B, 6813B, 6814B, 6834B, 6843A AC Power Solutions. These units will be referred to as "ac sources" throughout this manual. You will find the following information in the rest of this guide:

Chapter 1 Introduction to this guide.
Chapter 2 Introduction to SCPI messages structure, syntax, and data formats.
Chapter 3 Dictionary of SCPI commands.
Chapter 4 Introduction to programming the ac source with SCPI commands.
Appendix A SCPI command tree.
Appendix B SCPI conformance information.
Appendix C Error messages
Appendix D Elgar Model 9012 plug-in programmer compatibility
Appendix E IEC mode SCPI commands

Earlier AC Source Models

With the exception of some minor readback specification differences, information in this manual also applies to the following earlier ac source models:

Information about this current model	also applies to the following earlier models:
Keysight 6811B	Keysight 6811A AC Power Source/Analyzer
Keysight 6812B	Keysight 6812A AC Power Source/Analyzer Keysight 6841A Harmonic/Flicker Test System in normal mode
Keysight 6813B	Keysight 6813A AC Power Source/Analyzer Keysight 6842A Harmonic/Flicker Test System in normal mode

Documentation Summary

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

- *Quick Start Guide*. Information on how to quickly get started using the ac source.
- *User's Guide*. Includes specifications and supplemental characteristics, how to use the front panel, how to connect to the instrument, and calibration procedures.
- *Quick Reference Card*. Designed as a memory jogger for front panel and GPIB operation.
- *Keysight 14761A, Keysight 14762A, Keysight 14763A User's Guides* are shipped along with the specific software application and with Keysight 6843A units only.

External References

SCPI References

The following documents will assist you with programming in SCPI:

- *Beginner's Guide to SCPI*. Keysight Part No. H2325-90001. Highly recommended for anyone who has not had previous experience programming with SCPI.
- *Tutorial Description of the General Purpose Interface Bus*. Keysight Part No. 5952-0156. Highly recommended for those not familiar with the IEEE 488.1 and 488.2 standards.

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

GPIB References

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

The following are two formal documents concerning the GPIB interface:

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

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

Keysight VXI*plug&play* Power Products Instrument Drivers

Keysight VXI*plug&play* Power Products instrument drivers for Microsoft Windows 95 and Windows NT are now available on the Web at http://www.ag.com/go/drivers. These instrument drivers provide a high-level programming interface to your Keysight Power Products instrument. Keysight VXI*plug&play* instrument drivers are an alternative to programming your instrument with SCPI command strings. Because the instrument driver's function calls work together on top of the VISA I/O library, a single instrument driver can be used with multiple application environments.

Supported Applications

- Keysight VEE
- Microsoft Visual BASIC
- Microsoft Visual C/C++
- Borland C/C++
- National Instruments LabVIEW
- National Instruments LabWindows/CVI

System Requirements

The Keysight VXI*plug&play* Power Products instrument driver complies with the following:

- Microsoft Windows 95
- Microsoft Windows NT 4.0
- HP VISA revision F.01.02
- National Instruments VISA 1.1

Downloading and Installing the Driver

NOTE: Before installing the Keysight VXIplug&play instrument driver, make sure that you have one of the supported applications installed and running on your computer.

1. Access Keysight Technologies' Web site at http://www.ag.com/go/drivers.
2. Select the instrument for which you need the driver.
3. Click on the driver, either Windows 95 or Windows NT, and download the executable file to your PC.
4. Locate the file that you downloaded from the Web. From the **Start** menu select **Run** <path>:\agxxxx.exe - where <path> is the directory path where the file is located, and agxxxx is the instrument driver that you downloaded .
5. Follow the directions on the screen to install the software. The default installation selections will work in most cases. The readme.txt file contains product updates or corrections that are not documented in the on-line help. If you decide to install this file, use any text editor to open and read it.
6. To use the VXI*plug&play* instrument driver, follow the directions in the Keysight VXI*plug&play* online help under "Introduction to Programming".

Accessing Online Help

A comprehensive online programming reference is provided with the driver. It describes how to get started using the instrument driver with Keysight VEE, LabVIEW, and LabWindows. It includes complete descriptions of all function calls as well as example programs in C/C++ and Visual BASIC.

- To access the online help when you have chosen the default **Vxipnp** start folder, click on the **Start** button and select **Programs** | **Vxipnp** | **agxxxx Help (32-bit)**.

- where agxxxx is the instrument driver.

2

Introduction to Programming

GPIB Capabilities of the AC Source

All ac source functions except for setting the GPIB address are programmable over the GPIB. The IEEE 488.2 capabilities of the ac source are listed in the appendix A of the User's Guide.

GPIB Address

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

RS-232 Capabilities of the AC Source

The ac source provides an RS-232 programming interface, which is activated by commands located under the front panel **Address** key. All SCPI and E9012 commands are available through RS-232 programming. When the RS-232 interface is selected, the GPIB interface is disabled.

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

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

RS-232 Data Format

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

EVEN	Seven data bits with even parity
ODD	Seven data bits with odd parity
MARK	Seven data bits with mark parity (parity is always true)
SPACE	Seven data bits with space parity (parity is always false)
NONE	Eight data bits without parity

Parity options are stored in non-volatile memory.

Baud Rate

The front panel Address key lets you select one of the following baud rates, which is stored in non-volatile memory: 300 600 1200 2400 4800 9600

RS-232 Programming Example

The following program illustrates how to program the ac source using RS-232 to set the output voltage and frequency and to read back the model number and output voltage. The program was written to run on any controller using Microsoft QBasic.

NOTE: The ac source must be configured for RS232 and the same baud rate and parity as the controller.

```
' Program to write and read via RS232
' Configure serial port for:
'   9600 baud
'   7 bit data
'   2 stop bits
'   Ignore request to send
'   Ignore carrier detect
'   Even parity                 ' Needed with Vectra basic, ignored with QBasic
'   Send line feed
'   Reserve 1000 character buffer for serial I/O
'
DECLARE FUNCTION gets$ ()       ' Function to read string from ac source
CLS                             ' Clears screen
LOCATE 1, 1                     ' Position cursor at top left
' Configure Com1 Port
OPEN "com1:9600,e,7,2,rs,cd,pe,lf" FOR RANDOM AS #1 LEN = 1000
PRINT #1, "*RST"                ' Resets the ac source
PRINT #1, "VOLT 60"             ' Set voltage to 60 volts
PRINT #1, "FREQ 50"             ' Set frequency to 50 hertz
PRINT #1, "OUTPUT ON"           ' Turn on the output
PRINT #1, "*IDN?"               ' Query the ac source identification string
PRINT gets$                     ' Go to gets$ Function and print data returned
PRINT #1, MEAS"VOLT?"; volt     ' Query the ac source voltage
Volt = VAL (gets$)              ' Convert gets$ string to a value
PRINT gets$                     ' Print the value of the voltage
END                             ' End of main program

FUNCTION gets$                  ' Get a new line feed terminated string from device #1
C$ = ""                         ' Set C$ to null
WHILE c$ <> CHR$ (10)           ' Set loop to stop at Line Feed
    C$ = INPUT$ (1, #1)         ' Read 1 bit into file #1
    Resp$ = resp$ + c$          ' Concatenate bit with previous bits
WEND                            ' End of WHILE loop
gets$ = resp$                   ' Assign response to gets$
END FUNCTION
```

RS-232 Troubleshooting

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

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

Introduction to SCPI

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

Conventions Used in This Guide

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.
`Computer` font		Computer font is used to show program lines in text. `TRIGger:DELay .5` shows a program line.

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.

Types of SCPI Messages

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

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

The SCPI Command Tree

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

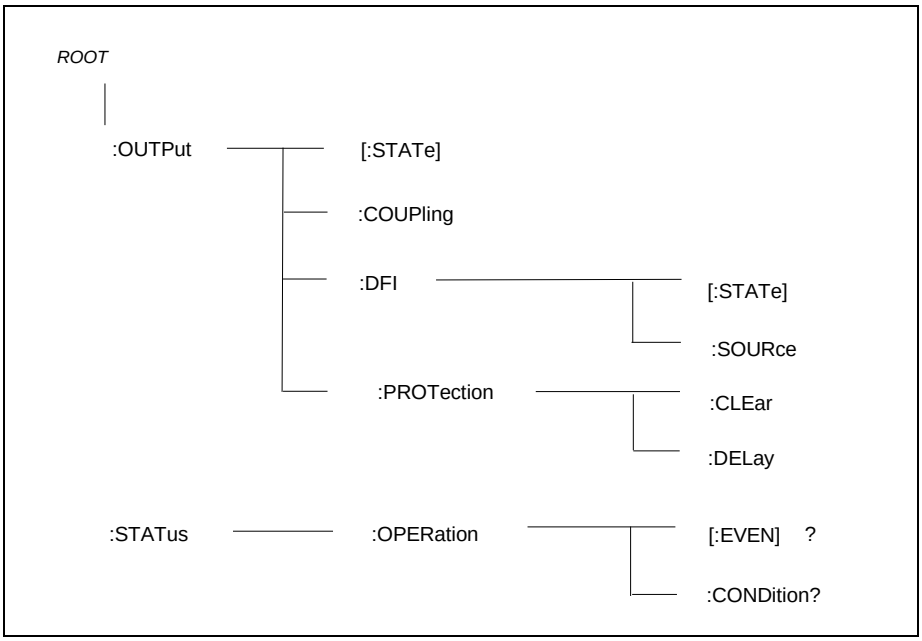


Figure 2-1. Partial Command Tree

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
- the SCPI interface encounters a root specifier

Active Header Path

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

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

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

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

```
OUTPut:PROTection:CLEAr    and
OUTPut:PROTection:DELay 20.
```

The entire message would be:

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

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

The Effect of Optional Headers

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

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

in one message. If you tried to send

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

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

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

Moving Among Subsystems

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

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

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

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

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

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

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

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

Including Common Commands

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

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

Using Queries

Observe the following precautions with queries:

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

Coupled Commands

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

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

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

Structure of a SCPI Message

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

The Message Unit

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

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

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

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

Combining Message Units

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

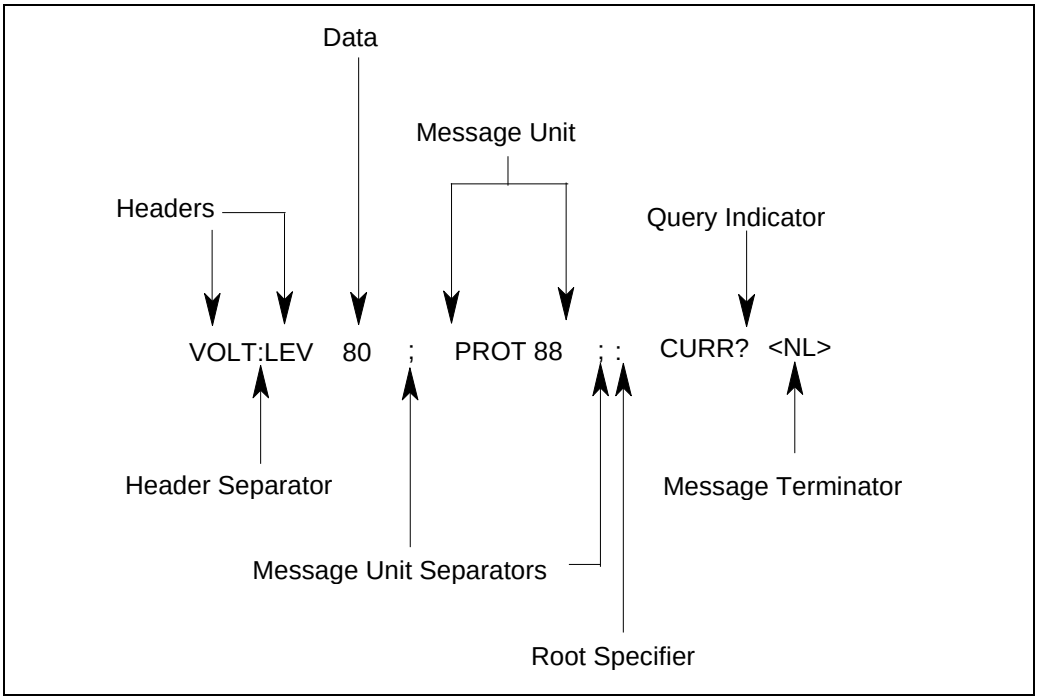


Figure 2-2. Command Message Structure

The basic parts of the above message are:

Message Component	**Example**
Headers	VOLT LEV PROT CURR
Header Separator	The colon in VOLT:LEV
Data	8088
Data Separator	The space in VOLT 80 and PROT 88
Message Units	VOLT:LEV 80 PROT 88 CURR?
Message Unit Separator	The semicolons in VOLT:LEV 80; and PROT 88;
Root Specifier	The colon in PROT 88;:CURR?
Query Indicator	The question mark in CURR?
Message Terminator	The <NL> (newline) indicator. Terminators are not part of the SCPI syntax

Headers

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

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

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

The SCPI interface is not sensitive to case. It will recognize any case mixture, such as TRIGGER, Trigger, TRIGger.

NOTE: Short form headers result in faster program execution.

Header Convention In the command descriptions in Chapter 3 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.

Query Indicator

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

Message Unit Separator

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

Root Specifier

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

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

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

Message Terminator

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

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

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

SCPI Data Formats

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

Numerical Data Formats

Symbol	Data Form
	Talking Formats
<NR1>	Digits with an implied decimal point assumed at the right of the least-significant digit. Examples: 273
<NR2>	Digits with an explicit decimal point. Example: .0273
<NR3>	Digits with an explicit decimal point and an exponent. Example: 2.73E+2
<Bool>	Boolean Data. Example: 0 | 1 or OFF | ON (0 = OFF; 1 = ON)
	Listening Formats
<Nrf>	Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273273. 2.73E2
<Nrf+>	Expanded decimal format that includes <NRf> and MINMAX. Examples: 273 73.2 .73E2 MAX. MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter.
<Bool>	Boolean Data. Example: 0 | 1

Suffixes and Multipliers

Class	Suffix	Unit	Unit with Multiplier
Current	A	ampere	MA (milliampere)
Amplitude	V	volt	MV (millivolt)
Time	S	second	MS (millisecond)
Frequency	HZ	Hertz	KHZ (kilohertz)
		Common Multipliers	
	1E3	K	kilo
	1E-3	M	milli
	1E-6	U	micro

Character Data

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

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

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

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

System Considerations

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

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

Assigning the GPIB Address in Programs

The ac source address cannot be set remotely. It must be set from the front panel. Once the address is set, you can assign it inside programs. The following examples assume that the GPIB select code is 7, and the ac source will be assigned to the variable ACS.

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

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

Types of DOS Drivers

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

Keysight **82335A Driver**

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

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

National Instruments GPIB **Driver**

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

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

Important Use error detection after every call to a subroutine.

Keysight BASIC Controllers

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

3

Language Dictionary

Introduction

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

Syntax Forms Syntax definitions use the long form, but only short form headers (or "keywords") appear in the examples. Use the long form to help make your program self-documenting.

Parameters Most commands require a parameter and all queries will return a parameter.The range for a parameter may vary according to the model of ac source. Parameters for all models are listed in the Specifications table in the User's Guide.

Models If a command only applies to specific models, those models are listed in the <Model> Only entry. If there is no <Model> Only entry, the command applies to all models.

Phases If a command can apply to individual phases of an , 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.

Order of Presentation The dictionary is organized as follows:

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

Subsystem Commands

Subsystem commands are specific to 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:

Subsystem	**Page**
Calibration Subsystem	29
Display Subsystem	34
Instrument Subsystem	35
Measurement Subsystem (Arrays)	37
Measurement Subsystem (Current)	42
Measurement Subsystem (Frequency)	48
Measurement Subsystem (Power)	49
Measurement Subsystem (Voltage)	52
Output Subsystem	55
Sense Subsystem	60
Source Subsystem (Current)	62
Source Subsystem (Frequency)	65
Source Subsystem (Function)	68
Source Subsystem (List)	71
Source Subsystem (Phase)	80
Source Subsystem (Pulse)	82
Source Subsystem (Voltage)	85
Status Subsystem	94
System Commands	102
Trace Subsystem	105
Trigger Subsystem	107
Common Commands	113

Calibration Subsystem Commands

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

- Enable and disable the calibration mode
- Change the calibration password
- Calibrate the current and voltage output levels, and store new calibration constants in 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
    :PWM
        :FREQuency <n>      Trim pulse width modulator frequency
        :RAMP <n>           Trim pulse width modulator ramp
    :SAVE                   Save new cal constants in non-volatile memory
    :STATE <bool> [,<n>]    Enable or disable calibration mode
    :VOLTage
        :AC                 Begin ac voltage calibration sequence
        :DC                 Begin dc voltage calibration sequence
        :OFFSet             Begin offset 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:SAV CAL:LEV

CALibrate:CURRent:MEASure

Keysight **6811B,** Keysight **6812B,** Keysight **6813B,** Keysight **6843A Only**

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

Command Syntax CALibrate:CURRent:MEASure
Parameters None
Examples `CAL:CURR:MEAS`
Related Commands CAL:STAT CAL:SAV 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:STAT CAL:SAV

CALibrate:IMPedance

Keysight **6811B,** Keysight **6812B,** Keysight **6813B,** Keysight **6843A Only**

This command can only be used in calibration mode. It calibrates the output impedance circuits. The 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:STAT CAL:SAV

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:SAV

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:PASSword <NRf>
Parameters 0 (default)
Examples `CAL:PASS 6812` `CAL:PASS 02.1997`
Related Commands CAL:STAT

CALibrate:PWM:FREQuency

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax CALCulate:PWM:FREQuency <NRf>
Parameters 0 through 7
Examples `CAL:PWM:FREQ 1`
Query Syntax CALibrate:PWM:FREQuency?
Returned Parameters <NR1>
Related Commands CAL:PWM:RAMP

CALibrate:PWM:RAMP

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax CALCulate:PWM:RAMP <NRf>
Parameters 0 through 255
Examples `CAL:PWM:RAMP 100`
Query Syntax CALibrate:PWM:RAMP?
Returned Parameters <NR1>
Related Commands CAL:PWM:FREQ

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 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 1,6812  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:DC

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax	CALibrate:VOLTage:DC
Parameters	None
Examples	`CAL:VOLT:DC`
Related Commands	CAL:SAVE CAL:STAT

CALibrate:VOLTage:OFFSet

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax CALibrate:VOLTage:OFFSet
Parameters None
Examples `CAL:VOLT:OFFS`
Related Commands CAL:SAVE CAL:STAT CAL:LEV

CALibrate:VOLTage:PROTection

This command can only be used in calibration mode. It calibrates the overvoltage protection (OV) circuit. The 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
Examples `CAL:VOLT:PROT`
Related Commands CAL:SAVE CAL:STAT

CALibrate:VOLTage:RTIMe

Keysight **6843A Only**

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

Command Syntax CALibrate:VOLTage:RTIMe
Parameters None
Examples `CAL:VOLT:RTIM`
Related Commands CAL:SAVE CAL:STAT

Display Subsystem Commands

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

Subsystem Syntax

```
DISPlay
    [:WINDow]
        [:STATe] <bool>              Enable/disable front panel display
        :MODE <mode>                 Set display mode (NORMal | TEXT)
        :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]:STATe <bool>
Parameters	0 | 1 | OFF | ON
***RST Value**	ON
Examples	`DISP:STAT 1,  DISP:STAT OFF`
Query Syntax	DISPlay[:WINDow]:STATe?
Returned Parameters	0 | 1
Related Commands	DISP:MODE DISP:TEXT

DISPlay:MODE

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

Command Syntax	DISPlay[:WINDow]:MODE <mode>
Parameters	NORMal | TEXT
***RST Value**	NORMal
Examples	`DISP:MODE TEXT`
Query Syntax	DISPlay[:WINDow]:MODE?
Returned Parameters	<CRD>
Related Commands	DISP DISP:TEXT

DISPlay:TEXT

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

Command Syntax	DISPlay[:WINDow]:TEXT[:DATA] <display_string>
Parameters	<display_string>
***RST Value**	null string
Examples	`DISP:TEXT "DO TEST1"`
Query Syntax	DISPlay[:WINDow]:TEXT?
Returned Parameters	<SRD> (the last programmed string)
Related Commands	DISP DISP:MODE

Instrument Subsystem

This subsystem programs the three-phase output capability of the Keysight 6834B .

Subsystem Syntax

INSTrument

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

INSTrument:COUPle

Keysight **6834B Only**

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.

NOTE: INSTrument:COUPle only affects the operation of subsequent commands. It does not by itself immediately affect the '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 <phase>
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

Keysight **6834B Only**

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` `INST:SEL OUTP1`
Query Syntax	INSTrument:NSELect?
Returned Parameters	<NR1>
Related Commands	INST:COUP

Measurement Subsystem (Arrays)

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 45Hz, the time interval is 25 microseconds. If digitization is caused by an acquire trigger, the time interval is set by SENSe:SWEep:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.

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

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

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	MEAS:ARR:VOLT:HARM? MEAS:ARR:CURR:HARM:PHAS?

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	MEAS:ARR:VOLT:HARM:PHAS? MEAS:ARR:CURR:HARM?

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

Keysight **6834B Only**

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 45Hz, 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

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

Keysight **6834B Only**

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 MEAS:ARR:CURR:NEUT:HARM:PHAS?

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

Keysight **6834B Only**

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
Examples	`MEAS:ARR:CURR:NEUT:HARM:PHAS?` `FETC:ARR:CURR:NEUT:HARM:PHAS?`
Returned Parameters	51 NR3 values
Related Commands	MEAS:ARR:CURR:NEUT:HARM?

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 45Hz, the time interval is 25 microseconds. If digitization is caused by an acquire trigger, the time interval is set by SENSe:SWEep:TINTerval, and the position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.

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

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

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	MEAS:ARR:CURR:HARM? MEAS:ARR:VOLT:HARM:PHAS?

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

Phase Selectable

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

The first value returned is the dc component (always returned as 0 degrees phase) , the second value is the fundamental frequency, and so on up to the 50th harmonic. Harmonic orders can be measured up to the fundamental measurement bandwidth of the measurement system, which is 12.6kHz. 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:PHASe? <NRf> FETCh:ARRay:VOLTage:HARMonic:PHASe? <NRf>
Parameters	None
Examples	`MEAS:ARR:VOLT:HARM:PHAS?` `FETC:ARR:VOLT:HARM:PHAS?`
Returned Parameters	51 NR3 values
Related Commands	MEAS:ARR:CURR:HARM:PHAS? MEAS:ARR:VOLT:HARM?

Measurement Subsystem (Current)

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

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 (3-phase only)
                    :PHASe? <n>         Returns neutral current harmonic phase (3-phase only)
```

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	MEAS:VOLT? MEAS:CURR:AC?

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	MEAS:VOLT:AC? MEAS:CURR?

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

Phase Selectable

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

Query Syntax	MEASure:[SCALar]:CURRent:ACDC? FETCh:[SCALar]:CURRent:ACDC?
Parameters	None
Examples	`MEAS:CURR:ACDC?` `FETC:CURR:ACDC?`
Returned Parameters	<NR3>
Related Commands	MEAS:VOLT:ACDC? MEAS:CURR:AMPL:MAX?

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

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	MEAS:CURR:ACDC? MEAS:CURR:CRES?

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:CRESfactor?
Parameters	None
Examples	`MEAS:CURR:CRES?  FETC:CURR:CRES?`
Returned Parameters	<NR3>
Related Commands	MEAS:CURR:ACDC? MEAS:CURR:AMPL:MAX?

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

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

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	MEAS:CURR:HARM? MEAS:CURR:HARM:THD?

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

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

Keysight **6834B Only**

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	MEAS:CURR:NEUT:AC? MEAS:CURR:NEUT:ACDC?

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

Keysight **6834B Only**

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	MEAS:CURR:NEUT? MEAS:CURR:NEUT:ACDC?

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

Keysight **6834B Only**

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

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	MEAS:CURR:NEUT? MEAS:CURR:NEUT:AC?

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

Keysight **6834B Only**

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	MEAS:CURR:NEUT:HARM:PHAS?

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

Keysight **6834B Only**

These queries return the phase angle of the Nth harmonic of current in the neutral output terminal of a three-phase ac source, 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	MEAS:CURR:NEUT:HARM?

Measurement Subsystem (Frequency)

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

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>

Measurement Subsystem (Power)

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

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	MEAS:POW?

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

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

Phase Selectable

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

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

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

Phase Selectable

These queries return the output power factor. The power factor is computed as:
pfactor = real power/apparent power

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

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

Keysight **6834B Only**

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>

Measurement Subsystem (Voltage)

This subsystem programs the voltage measurement capability of the ac source. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new measurement data before returning a reading. FETCh returns a reading computed from previously acquired data. Individual outputs of a three-phase source are specified by the setting of INSTrument:NSELect.

Subsystem Syntax

```
MEASure | FETCh
    [:SCALar]
        :VOLTage
            [:DC]?                  Returns dc component of the voltage
            :AC?                    Returns ac rms voltage
            :ACDC?                  Returns ac+dc rms voltage
            :HARMonic
                [:AMPLitude]? <n>   Returns amplitude of the Nth harmonic of voltage
                :PHASe? <n>         Returns phase of the Nth harmonic of voltage
                :THD?               Returns % of total harmonic distortion of voltage
```

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 MEAS:CURR? MEAS:VOLT:AC?

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 MEAS:CURR:AC? MEAS:VOLT?

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

Phase Selectable

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

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

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

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

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	MEAS:VOLT:HARM? MEAS:VOLT:HARM:THD?

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

Output Subsystem

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

Subsystem Syntax

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

OUTPut

This command enables or disables the 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 | 1 | OFF | ON
***RST Value**	OFF
Examples	`OUTP 1` `OUTP:STAT ON`
Query Syntax	OUTPut[:STATe]?
Returned Parameters	0 | 1
Related Commands	*RCL *SAV

OUTPut:COUPling

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

Command Syntax	OUTPut:COUPling <coupling>
Parameters	AC | DC
***RST Value**	AC
Examples	`OUTP:COUP DC`
Query Syntax	OUTPut:COUPling?
Returned Parameters	<CRD>
Related Commands	*RCL *SAV

OUTPut:DFI

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

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

OUTPut:DFI:SOURce

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

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

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

OUTPut:IMPedance

Keysight **6811B,** Keysight **6812B,** Keysight **6813B 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 ON`
Query Syntax	OUTPut:IMPedance[:STATe]?
Returned Parameters	0 | 1
Related Commands	OUTP:IMP:REAL OUTP:IMP:REAC

OUTPut:IMPedance:REAL

Keysight **6811B,** Keysight **6812B,** Keysight **6813B 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	OUTPut:IMPedance:REAL <NRf>
Parameters	0 to 1 (ohms)
***RST Value**	0
Examples	`OUTP:IMP:REAL 0.25`
Query Syntax	OUTPut:IMPedance:REAL?
Returned Parameters	<NR3>
Related Commands	OUTP:IMP OUTP:IMP:REAC

OUTPut:IMPedance:REACtive

Keysight **6811B,** Keysight **6812B,** Keysight **6813B 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	OUTPut:IMPedance:REACtive <NRf>
Parameters	0.00002 to 0.001 (henrys)
***RST Value**	0.0005
Examples	`OUTP:IMP:REAC 100E-6`
Query Syntax	OUTPut:IMPedance:REAC?
Returned Parameters	<NR3>
Related Commands	OUTP:IMP OUTP:IMP:REAL

OUTPut:PON:STATe

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

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

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

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

OUTPut:PROTection:CLEar

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

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

OUTPut:PROTection:DELay

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

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

OUTPut:RI:MODE

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

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

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

OFF The instrument ignores the RI input.

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

OUTPut:TTLTrg

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

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

OUTPut:TTLTrg:SOURce

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

BOT Beginning of transient output

EOT End of transient output

LIST Specified by the TTLTrg list

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

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

Sense Subsystem

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] <type>          Sets measurement window type (KBESsel | RECTangular)
```

SENSe:CURRent:ACDC:RANGe

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Keysight **6811B**

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

Keysight **6812B,** Keysight **6813B**

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

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

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

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

SENSe:SWEep:OFFSet:POINts

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

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

SENSe:SWEep:TINTerval

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

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

Command Syntax	SENSe:SWEep:TINTeral <NRf+>
Parameters	25.037 through 250.37 (microseconds) | MAXimum | MINimum
***RST Value**	25.037 μs (Keysight 6814B/6834B) 25.049 μs (Keysight 6811B/6812B/6813B/6843A)
Examples	`SENS:SWE:TINT 100E-6`
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 10.000207Hz

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

Source Subsystem (Current)

This subsystem programs the output current of the ac source.

Subsystem Syntax

```
[SOURce:]
    CURRent
        [:LEVel]
            [:IMMediate]
                [:AMPLitude] <n>        Sets the rms current limit
        :PEAK
            [:IMMediate] <n>            Sets the peak current limit
            :MODE <mode>                Sets peak current limit mode (FIX | STEP | PULS | LIST)
            :TRIGgered <n>              Sets the transient level for peak 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 with the limit. The CL bit of the Questionable Status register indicates that the current limit control loop is active. If the current protection state is programmed on, the output latches into a disabled state when current limiting occurs.

NOTE: On Keysight models 6814B, 6834B and 6843A, 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 Chapter 4 under "Coupled Commands" for more information.

Command Syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>
Parameters	refer to Specifications Table in User's Guide
Unit	*A* (rms amperes)
***RST Value**	MAXimum (Keysight 6811B/6812B/6813B) 1 (Keysight 6814B/6834B/6843A)
Examples	`CURR 5` `CURR:LEV .5`
Query Syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	CURR:PROT:STAT VOLT:RANG

CURRent:PEAK

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax	[SOURce:]CURRent:PEAK[:IMMediate] <NRf+>
Parameters	refer to Specifications Table in User's Guide
Unit	*A* (peak amperes)
***RST Value**	13 (Keysight 6811B/6812B) 26 (Keysight 6813B)
Examples	`CURR:PEAK:IMM 15`
Query Syntax	[SOURce:]CURRent:PEAK[:IMMediate]?
Returned Parameters	<NR3>
Related Commands	CURR:PEAK:MODE CURR:PEAK:TRIG

CURRent:PEAK:MODE

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

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

CURRent:PEAK:TRIGgered

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

This command sets the output limit of the absolute value of peak instantaneous current when a step or pulse transient is triggered.

Command Syntax	[SOURce:]CURRent:PEAK:TRIGgered <NRf+>
Parameters	refer to Specifications Table in User's Guide
Unit	*A* (peak amperes)
***RST Value**	13 (Keysight 6811B/6812B) 26 (Keysight 6813B)
Examples	`CURR:PEAK:TRIG 15`
Query Syntax	[SOURce:]CURRent:PEAK:TRIG?
Returned Parameters	<NR3>
Related Commands	CURR:PEAK CURR:PEAK:MODE

CURRent:PROTection:STATe

This command enables or disables the overcurrent (OC) protection function. If the overcurrent protection function is enabled and the exceeds the programmed level, then the output is disabled and the Questionable Condition status register OC bit is set (see Chapter 4 under "Programming the Status Registers"). An overcurrent condition can be cleared with OUTPut:PROTection:CLEar after the cause of the condition is removed.

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

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

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	refer to Specifications Table in User's Guide
Unit	*HZ* (hertz)
***RST Value**	60
Examples	`FREQ 50`
Query Syntax	[SOURce:]FREQuency[:CW | :IMMediate]?
Returned Parameters	<NR3>
Related Commands	FREQ:MODE FREQ:SLEW

FREQuency:MODE

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

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

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

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

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

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

FREQuency:SLEW

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

Command Syntax	[SOURce:]FREQuency:SLEW[:IMMediate] <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
***RST Value**	MAXimum
Examples	`FREQ:SLEW:IMM 75` `FREQ:SLEW MAX`
Query Syntax	[SOURce:]FREQuency:SLEW[:IMMediate]?
Returned Parameters	<NR3>
Related Commands	FREQ FREQ:SLEW:MODE

FREQuency:SLEW:MODE

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

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

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

FREQency:SLEW:TRIGgered

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

Command Syntax	[SOURce:]FREQuency:SLEW:TRIGgered <NRf+> | INFinity
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
***RST Value**	MAXimum
Examples	`FREQ:SLEW:TRIG 75` `FREQ:SLEW:TRIG MAX`
Query Syntax	[SOURce:]FREQuency:SLEW:TRIGgered?
Returned Parameters	<NR3>
Related Commands	FREQ FREQ:SLEW:MODE

FREQuency:TRIGgered

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

Command Syntax	[SOURce:]FREQuency:TRIGgered <NRf+>
Parameters	refer to Specifications Table in User's Guide
Unit	*HZ* (hertz)
***RST Value**	60
Examples	`FREQ:TRIG 50`
Query Syntax	[SOURce:]FREQuency:TRIGgered?
Returned Parameters	<NR3>
Related Commands	FREQ FREQ:MODE

Source Subsystem (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>)
            :MODE <mode>           Sets the waveform shape mode (FIX | STEP | PULS | LIST)
            :TRIGgered <shape>     Sets the triggered transient shape (SIN | SQU | CSIN | <table>)
            :CSINusoid <n> [THD] Sets the % of peak at which the sinewave 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 sinewave. Both positive and negative peak amplitudes are clipped at a value determined by the FUNCtion:CSINusoid command.
<table>	The output shape is described by one of the user-defined waveform tables.

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

NOTE: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

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

FUNCtion:MODE

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

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

NOTE: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

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

FUNCtion:CSINusoid

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

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

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

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 sequences through a list before that list is completed. LIST:DWELl specifies the time interval that each value (point) of a list is to remain in effect. LIST:STEP determines if a trigger causes a list to advance only to its next point or to sequence through all of its points.

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

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

Subsystem Syntax

[SOURce:]	
LIST	
:COUNt <n> | INFinity	Sets the list repeat count
:CURRent <n> {,<n>}	Sets the peak current limit list
:POINts?	Returns the number of peak current limit list points
:DWELl <n> {,<n>}	Sets the list of dwell times
:POINts?	Returns the number of dwell list points
:FREQuency	
[:LEVel] <n> {,<n>}	Sets the frequency list
:POINts?	Returns the number of frequency points
:SLEW <n> {,<n>}	Sets the frequency slew list
:POINts?	Returns the number of frequency slew points
:PHASe <n> {,<n>}	Sets the phase list
:POINts?	Returns the number of phase list points
:SHAPe <shape> {,<shape>}	Sets the waveform shape list
:POINts?	Returns the number of shape list points
:STEP <step>	Specifies how the list responds to triggers (ONCE | AUTO)
:TTLTrg <bool> {,<bool>}	Defines the output marker list
:POINts?	Returns the number of output marker list points
:VOLTage	
[:LEVel] <n> {,<n>}	Sets the voltage list
:POINts?	Returns the number of voltage level points
:SLEW <n> {,<n>}	Sets the voltage slew list
:POINts?	Returns the number of voltage slew points
:OFFSet <n> {,<n>}	Sets the voltage offset list
:POINts?	Returns the number of voltage offset points
:SLEW <n> {,<n>}	Sets the offset voltage slew list
:POINts?	Returns the number of offset voltage slew points

LIST:COUNt

This command sets the number of times that the list is executed before it is completed. The command accepts parameters in the range 1 through 9.9E37, but any number greater than 2E9 is interpreted as infinity. Use INFinity to execute a list indefinitely.

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

LIST:CURRent

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax	[SOURce:]LIST:CURRent <NRf+> {,<NRf+>}
Parameters	refer to Specifications Table in User's Guide
Unit	*A* (peak current)
Examples	`LIST:CURR 2.5,3.0,3.5` `LIST:CURR MAX,3.5,2.5,MIN`
Query Syntax	[SOURce:]LIST:CURRent?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:CURR:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:CURRent:POINts?

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Query Syntax	[SOURce:]LIST:CURRent:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:CURR:POIN?`
Related Commands	LIST:CURR

LIST:DWELl

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

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

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

Command Syntax	[SOURce:]LIST:DWELl <NRf+> {,<NRf+>}
Parameters	3-phase models: 0 to 1.07533E6 | MINimum | MAXimum 1-phase models: 0 to 4.30133E5 | MINimum | MAXimum
Unit	*S* (seconds)
Examples	`LIST:DWEL 2.5,1.5,.5`
Query Syntax	[SOURce:]LIST:DWELl?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:CURR LIST:FREQ LIST:TTLT LIST:VOLT

LIST:DWELl:POINts?

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

Query Syntax	[SOURce:]LIST:DWELl:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:DWEL:POIN?`
Related Commands	LIST:DWEL

LIST:FREQuency

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

Command Syntax	[SOURce:]LIST:FREQuency[:LEVel] <NRf+> {,<NRf+>}
Parameters	refer to Specifications Table in User's Guide
Unit	*HZ* (hertz)
Examples	`LIST:FREQ 55,60,65`
Query Syntax	[SOURce:]LIST:FREQuency[:LEVel]?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:FREQ:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:FREQuency:POINts?

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

Query Syntax	[SOURce:]LIST:FREQuency[:LEVel]:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:FREQ:POIN?`
Related Commands	LIST:FREQ

LIST:FREQuency:SLEW

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

Command Syntax	[SOURce:]LIST:FREQuency:SLEW <NRf+> | INF {,<NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*HZ* (hertz per second)
Examples	`LIST:FREQ:SLEW 10,20,1E2`
Query Syntax	[SOURce:]LIST:FREQuency:SLEW?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:FREQ:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:FREQuency:SLEW:POINts?

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

Query Syntax	[SOURce:]LIST:FREQuency:SLEW:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:FREQ:SLEW:POIN?`
Related Commands	LIST:FREQ:SLEW

LIST:PHASe

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 (degrees) | MAXimum | MINimum
Examples	`LIST:PHAS 90,120,150`
Query Syntax	[SOURce:]LIST:PHAS?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:FREQ:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:PHASe:POINts?

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

Query Syntax [SOURce:]LIST:PHASe:POINTs?
Returned Parameters <NR1>
Examples `LIST:PHAS:POIN?`
Related Commands LIST:PHAS

LIST:SHAPe

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

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

NOTE: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

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

LIST:SHAPe:POINts?

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

Query Syntax [SOURce:]LIST:SHAPe:POINTs?
Returned Parameters <NR1>
Examples `LIST:SHAP:POIN?`
Related Commands LIST:SHAP

LIST:STEP

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

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

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

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

LIST:TTLTrg

This command sets the sequence of Trigger Out list points. Each point which is set **ON** will cause a pulse to be output at Trigger Out when that list step is reached. Those entries which are set **OFF** will not generate Trigger Out pulses.

The order in which the list points are given determines the sequence in which Trigger Out pulses will be output when a list is triggered. Changing list data while a subsystem is in list mode generates an implied ABORt.

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

LIST:TTLTrg:POINts?

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

Query Syntax	[SOURce:]LIST:TTLTrg:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:TTLT:POIN?`
Related Commands	LIST:TTLT

LIST:VOLTage

Phase Selectable

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

NOTE: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

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

LIST:VOLTage:POINts?

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

Query Syntax	[SOURce:]LIST:VOLTage[:LEVel]:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:VOLT:POIN?`
Related Commands	LIST:VOLT

LIST:VOLTage:SLEW

Phase Selectable

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

Command Syntax	[SOURce:]LIST:VOLTage:SLEW <NRf+> | INF {,<NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
Examples	`LIST:VOLT:SLEW 10,20,1E2`
Query Syntax	[SOURce:]LIST:VOLTage:SLEW?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:VOLT:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:VOLTage:SLEW:POINts?

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

Query Syntax	[SOURce:]LIST:VOLTage:SLEW:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:VOLT:SLEW:POIN?`
Related Commands	LIST:VOLT:SLEW

LIST:VOLTageOFFSet

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

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

LIST:VOLTage:OFFSet:POINts?

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Query Syntax	[SOURce:]LIST:VOLTage:OFFSet:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:VOLT:OFFS:POIN?`
Related Commands	LIST:VOLT:OFFS

LIST:VOLTage:OFFSet:SLEW

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Command Syntax	[SOURce:]LIST:VOLTage:OFFSet:SLEW <NRf+> | INF {,<NRf+> | INF}
Parameters	0 to 9.9E37 | MAXimum | MINimum | INFinity
Unit	*V* (volts per second)
Examples	`LIST:VOLT:OFFS:SLEW 10,20,1E2`
Query Syntax	[SOURce:]LIST:VOLTage:OFFSet:SLEW?
Returned Parameters	<NR3> {,<NR3>}
Related Commands	LIST:VOLT:SLEW:POIN? LIST:COUN LIST:DWEL LIST:STEP

LIST:VOLTage:OFFSet:SLEW:POINts?

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

Query Syntax	[SOURce:]LIST:VOLTage:OFFSet:SLEW:POINTs?
Returned Parameters	<NR1>
Examples	`LIST:VOLT:OFFSet:SLEW:POIN?`
Related Commands	LIST:VOLT:OFFS

Source Subsystem (Phase)

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

Subsystem Syntax

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

PHASe

Phase Selectable

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

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

Command Syntax	[SOURce:]PHASe[:ADJust|:IMMediate] <NRf+>
Parameters	–360 through +360 (degrees) | MAXimum | MINimum
***RST Value**	phase 1 = 0, phase 2 = 240, phase 3 = 120
Examples	`PHAS 90` `PHAS MAX`
Query Syntax	[SOURce:]PHASe[:ADJust|:IMMediate]?
Returned Parameters	<NR3>
Related Commands	PHAS:MODE PHASE: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 output phase is controlled by the phase list when a triggered transient occurs.

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

PHASe:TRIGgered

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 (degrees) | MAXimum | MINimum
***RST Value** triggered phase 1 = 0, triggered phase 2 = 240, triggered phase 3 = 120
Examples
```
PHAS:TRIG 90      PHAS:TRIG MAX
```
Query Syntax [SOURce:]PHASe:TRIGgered?
Returned Parameters <NR3>
Related Commands PHAS:MODE PHASE

Source Subsystem (Pulse)

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

Subsystem Syntax

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

PULSe:COUNt

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

Command Syntax	[SOURce:]PULSe:COUNt <Nrf+> | INFinity
Parameters	1 to 9.9E37 | MINimum | MAXimum | INFinity
***RST Value**	1
Examples	`PULS:COUN 3` `PULS:COUN MIN` `PULS:COUN INF`
Query Syntax	[SOURce:]PULSe:COUNt?
Returned Parameters	<NR3>
Related Commands	PULS:DCYC PULS:HOLD PULS:PER PULS:PER

PULSe:DCYCle

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

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

PULSe:HOLD

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

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

PULSe:HOLD = WIDTh

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

PULSe:HOLD = DCYCle

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

PULSe:PERiod

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

Command Syntax	[SOURce:]PULSe:PERiod <NRf+>
Parameters	3-phase models: 0 to 1.07533E6 | MINimum | MAXimum 1-phase models: 0 to 4.30133E5 | MINimum | MAXimum
Unit	*S* (seconds)
***RST Value**	.03333
Examples	`PULS:PER 0.001` `PULS:PER MIN`
Query Syntax	[SOURce:]PULSe:PERiod?
Returned Parameters	<NR3>
Related Commands	PULS:COUN PULS:DCYC PULS:PER PULS:HOLD

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**	.01667 (equals the period of a single 60 Hz cycle)
Examples	`PULS:WIDT 0.001` `PULS:WIDT MIN`
Query Syntax	[SOURce:]PULSe:WIDTh?
Returned Parameters	<NR3>
Related Commands	PULS:COUN PULS:DCYC PULS:PER PULS:HOLD

Source Subsystem (Voltage)

This subsystem programs the output voltage of the ac source.

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)
        :OFFSet
            [:IMMediate] <n>                Sets the dc offset voltage
            :MODE <mode>                    Sets the offset mode (FIX | STEP | PULS | LIST)
            :TRIGgered <n>                  Sets the transient dc offset voltage
            :SLEW
                [:IMMediate] <n> | INFinity Sets the voltage slew rate
                :MODE <mode>                Sets voltage slew mode (FIX | STEP | PULS | LIST)
                :TRIGgered <n> | INFinity   Sets the transient voltage slew rate
        :PROTection
            [:LEVel] <n>                    Sets the overvoltage protection threshold
            :STATe <bool>                   Sets the overvoltage protection state
        :RANGe <n>                          Sets the voltage range
        :SENSe |ALC
            :DETector RTIMe | RMS           Sets the sense detector for the voltage control loop
            :SOURce INTernal | EXTernal     Sets voltage sense source
        :SLEW
            [:IMMediate] <n> | INFinity     Sets the voltage slew rate
            :MODE <mode>                    Sets voltage slew mode (FIX | STEP | PULS | LIST)
            :TRIGgered <n> | INFinity       Sets the transient voltage slew rate
```

VOLTage

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, 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: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

Command Syntax	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>
Parameters	For sinewaves: 0 to 300 | MAXimum | MINimum
Unit	*V* (rms voltage)
***RST Value**	1
Examples	`VOLT 115` `VOLT:LEV 250`
Query Syntax	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?
Returned Parameters	<NR3>
Related Commands	VOLT:MODE VOLT:TRIG VOLT:OFFS FUNC:SHAP

VOLTage:TRIGgered

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, 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: For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

Command Syntax	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf+>
Parameters	For sinewaves: 0 to 300 | MAXimum | MINimum
Unit	*V* (rms voltage)
***RST Value**	1
Examples	`VOLT:TRIG 120` `VOLT:LEV:TRIG 150`
Query Syntax	[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
Returned Parameters	<NR3> (if the trigger level is not programmed, the immediate level is returned)
Related Commands	VOLT VOLT:MODE VOLT:OFFS FUNC:SHAP

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

VOLTage:OFFSet

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

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

Command Syntax [SOURce:]VOLTage:OFFSet[:IMMediate] <NRf+>
Parameters –425 to +425 | MAXimum | MINimum
Unit *V* (dc voltage)
***RST Value** 0
Examples `VOLT:OFFS 100`
Query Syntax [SOURce:]VOLTage:OFFSet[:IMMediate]?
Returned Parameters <NR3>
Related Commands VOLT:OFFS:MODE OUTP:COUP FUNC:SHAP

VOLTage:OFFSet:MODE

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

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

VOLTage:OFFSet:TRIGgered

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

The maximum peak voltage that the ac source can output is 425 V peak. This includes any combination of voltage, voltage offset, and function shape values. Therefore, the maximum value that can be programmed depends on the peak-to-rms ratio of the selected waveform. For a sinewave, the maximum voltage that can be programmed is 300 V rms.

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

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

VOLTage:OFFSet:SLEW

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

VOLTage:OFFSet:SLEW:MODE

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

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

VOLTage:OFFSet:SLEW:TRIGgered

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

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

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

VOLTage:PROTection

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 output is disabled and the Questionable Condition status register OV bit is set (see Chapter 4 under Programming the Status 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 | MAXimum | MINimum
Unit	*V* (peak voltage)
***RST Value**	MAXimum
Examples	`VOLT:PROT 400` `VOLT:PROT:LEV MAX`
Query Syntax	[SOURce:]VOLTage:PROTection[:LEVel]?
Returned Parameters	<NR3>
Related Commands	OUTP:PROT:CLE OUTP:PROT:DEL

VOLTage:PROTection:STATe

Keysight **6811B,** Keysight **6812B,** Keysight **6813B Only**

This command enables or disables the over-voltage protection feature.

Command Syntax	[SOURce:]VOLTage:PROTection:STATe <Bool>
Parameters	0 | 1 | OFF | ON
***RST Value**	OFF
Examples	`VOLT:PROT:STAT 1` `VOLT:PROT:STAT ON`
Query Syntax	[SOURce:]VOLTage:PROTection:STATe?
Returned Parameters	<NR3>
Related Commands	VOLT:PROT

VOLTage:RANGe

Keysight **6814B,** Keysight **6834B,** Keysight **6843A Only**

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 | MAXimum | MINimum
***RST Value**	MAXimum
Examples	`VOLT:RANG 150` `VOLT:RANG MIN`
Query Syntax	[SOURce:]VOLTage:RANGe?
Returned Parameters	<NR3>
Related Commands	VOLT

VOLTage:SENSe:DETector
VOLTage:ALC:DETector

Keysight **6811B,** Keysight **6812B,** Keysight **6813B 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 rear panel voltage sense terminals, which allows remote voltage sensing at the load.

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

VOLTage:SLEW

Phase Selectable

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

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

VOLTage:SLEW:MODE

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** FIXed
Examples `VOLT:SLEW:MODE STEP`
Query Syntax [SOURce:]VOLTage:SLEW:MODE?
Returned Parameters <CRD>
Related Commands VOLT:SLEW VOLT:SLEW:TRIG

VOLTage:SLEW:TRIGgered

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 sets the slew to its maximum possible rate. The SCPI representation for infinity is 9.9E37.

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

Status Subsystem

This subsystem programs the ac source status registers. The ac source has 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 4 under “Programming the Status Registers” for more information.

Subsystem Syntax

STATus	
:PRESet	Presets all enable and transition registers to power-on
:OPERation	
[:EVENt]?	Returns the value of the event register
:CONDition?	Returns the value of the condition register
:ENABle <n>	Enables specific bits in the Event register
:NTRansition<n>	Sets the Negative transition filter
:PTRansition<n>	Sets the Positive transition filter
:QUEStionable	
[:EVENt]?	Returns the value of the event register
:CONDition?	Returns the value of the condition register
:ENABle <n>	Enables specific bits in the Event register
:NTRansition<n>	Sets the Negative transition filter
:PTRansition<n>	Sets the Positive transition filter
: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 Configuration of Operation Status Registers

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.

STATus:OPERation?

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

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

STATus:OPERation:CONDition?

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

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

STATus:OPERation:ENABle

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

Command Syntax STATus:OPERation:ENABle <NRf+>
Parameters 0 to 32767 | MAXimum | MINimum
Default Value 0
Examples `STAT:OPER:ENAB 32` `STAT:OPER:ENAB 1`
Query Syntax STATus:OPERation:ENABle?
Returned Parameters <NR1> (register value)
Related Commands STAT:OPER?

STATus:OPERation:NTRansition
STATus:OPERation:PTRansition

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

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

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

Command Syntax	STATus:OPERation:NTRansition <NRf+> STATus:OPERation:PTRansition <NRf+>
Parameters	0 to 32767 | MAXimum | MINimum
Default Value	0
Examples	`STAT:OPER:NTR 32` `STAT:OPER:PTR 1`
Query Syntax	STATus:OPERation:NTRansition? STATus:OPERation:PTRansition?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:OPER:ENAB

Bit Configuration of Questionable Status Registers

Bit Position	15	14	13	12	11	10	9	8–5	4	3	2	1	0
Bit Name	not used	Meas Ovld	Isum	CL rms	Rail	CL peak	RI	not used	OT	UNR	SOA	OCP	OV
Bit Weight		16384	8192	4096	2048	1024	512		16	8	4	2	1

OV	over-voltage protection has tripped
OCP	over-current protection has tripped
SOA	safe operating area protection has tripped (Keysight 6811B, 6812B, 6813B)
UNR	output is unregulated
OT	over-temperature protection has tripped
RI	remote inhibit is active
CL peak	peak current limit is active (Keysight 6811B, 6812B, 6813B)
Rail	rail protection tripped (Keysight 6811B, 6812B, 6813B); rail voltage unregulated (Keysight 6814B, 6834B, 6843A)
CL rms	rms current limit is active
Isum	summary of Isum registers (Keysight 6834B)
MeasOvld	current measurement exceeded low current range capability (Keysight 6811B, 6812B, 6813B)

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.

NOTE: On the Keysight 6834B, 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
Examples `STAT:QUES:EVEN?`
Returned Parameters <NR1> (register value)
Related Commands *CLS STAT:QUES:NTR STAT:QUES:PTR

STATus:QUEStionable:CONDition?

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

Query Syntax STATus:QUEStionable:CONDition?
Parameters None
Examples `STAT:QUES:COND?`
Returned Parameters <NR1> (register value)
Related Commands STAT:OPER:COND?

STATus:QUEStionable:ENABle

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

Command Syntax	STATus:QUEStionable:ENABle <NRf+>
Parameters	0 to 32767 | MAXimum | MINimum
Default Value	0
Examples	`STAT:QUES:ENAB 32` `STAT:QUES:ENAB 1`
Query Syntax	STATus:QUEStionable:ENABle?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES?

STATus:QUEStionable:NTRansition
STATus:QUEStionable:PTRansition

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

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

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

Command Syntax	STATus:QUEStionable:NTRansition <NRf+> STATus:QUEStionable:PTRansition <NRf+>
Parameters	0 to 32767 | MAXimum | MINimum
Default Value	0
Examples	`STAT:QUES:NTR 32` `STAT:QUES:PTR 1`
Query Syntax	STATus:QUEStionable:NTRansition? STATus:QUEStionable:PTRansition?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES:ENAB

Bit Configuration of Questionable Instrument Summary Registers

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

OV	over-voltage protection has tripped
OCP	over-current protection has tripped
UNR	output is unregulated
OT	over-temperature protection has tripped
RI	remote inhibit is active
Rail	rail protection tripped (Keysight 6811B, 6812B, 6813B); rail voltage unregulated (Keysight 6814B, 6834B, 6843A)
CL rms	rms current limit is active

STATus:QUEStionable:INSTrument:ISUMmary?

Keysight **6834B Only**

Phase Selectable

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

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

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

STATus:QUEStionable:INSTrument:ISUMmary:CONDition?

Keysight **6834B Only**

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 ac source.

Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:CONDition]?
Parameters	None
Examples	`STAT:QUES:INST:ISUM:COND?`
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES:COND?

STATus:QUEStionable:INSTrument:ISUMmary:ENABle

Keysight **6834B Only**

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 | MAXimum | MINimum
Default Value	0
Examples	`STAT:QUES:INST:ISUM:ENAB 32`
Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:ENABle?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES:INST:ISUM?

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

Keysight **6834B Only**

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

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

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

Command Syntax	STATus:QUEStionable:INSTrument:ISUMmary:NTRansition <NRf+> STATus:QUEStionable:INSTrument:ISUMmary:PTRansition <NRf+>
Parameters	0 to 32767 | MAXimum | MINimum
Default Value	0
Examples	`STAT:QUES:INST:ISUM:NTR 32`
Query Syntax	STATus:QUEStionable:INSTrument:ISUMmary:NTRansition? STATus:QUEStionable:INSTrument:ISUMmary:PTRansition?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES:INST:ISUM:ENAB

System Commands

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

Subsystem Syntax

SYSTem

:CONFigure <mode>	Selects the operating mode of the ac source (NORM | IEC)
:NOUTputs <n>	Select the number of output phases (1 or 3)
:ERRor?	Returns the error number and error string
:VERSion?	Returns the SCPI version number
:LANGuage <language>	Sets the programming language (SCPI | E9012)
: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

Keysight **6812B, 6813B, 6843A Only**

This command sets the overall operating mode of the ac source. The choices are:

NORMal Causes the unit to operate in standard ac source mode.

IEC Modifies the basic behavior of the transient and measurement systems to facilitate harmonic and flicker emissions testing. (IEC mode is automatically selected when running the Keysight 14761A Harmonic and Flicker Emissions Tests software application.)

Refer to SYSTem:CONFigure in appendix E for more information about the differences between Normal mode and IEC mode.

Command Syntax	SYSTem:CONFigure <mode>
Parameters	NORMal | IEC
***RST Value**	NORMal
Examples	`SYST:CONF NORM`
Query Syntax	SYSTem:CONFigure?
Returned Parameters	<CRD>
Related Commands	ABORt MEAS:ARR:CURR MEAS:ARR:VOLT SENS:WIND

SYSTem:CONFigure:NOUTputs

Keysight **6834B Only**

This command selects the number of output phases for ac sources that have single-phase and three-phase switchable capability. This selection is stored in non-volatile memory and is retained after power-off. The execution of this command disables all outputs, reconfigures the current readback and programming calibration constants, returns all lists and *RCL states to their factory default states, and reboots the ac source. Note that this may require you to reprogram the lists and recall states each time the outputs are switched.

Command Syntax	SYSTem:CONFigure:NOUTputs <NR1>
Parameters	1 or 3
Examples	`SYST:CONF:NOUT 1`
Query Syntax	SYSTem:CONFigure:NOUTputs?
Returned Parameters	<NR1>

SYSTem:ERRor?

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

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

SYSTem:VERSion?

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

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

SYSTem:LANGuage

Sets the command language of the ac 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. Refer to Appendix D for more information.

Command Syntax	SYSTem:LANGuage <language>
Parameters	SCPI | 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:REM`
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:REM SYST:LOC

Trace Subsystem

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

Subsystem Syntax

TRACe | DATA	
:CATalog?	Return list of defined waveforms
[:DATA] <waveform_name>, <n> {, <n>}	Assign values to a waveform
:DEFine <waveform_name>[, <waveform_name> | 1024]	Create and name new waveform
:DELete	
[:NAME] <waveform_name>	Delete waveform to free its memory

TRACe
DATA

These commands set the values of a user-defined waveform table.

The first parameter is the name of a waveform that was previously defined with TRACe:DEFine. Following the name are 1024 data points that define the relative amplitudes of exactly one cycle of the waveform. The first data point defines the relative amplitude that will be output at 0 degrees phase reference. An error will occur if exactly 1024 data points are not sent with the command.

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

Waveform data is stored in 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>
Examples	`TRAC flattop,0.1,0.3,0.7,.....-0.7,-0.3,-0.1`
Query Syntax	TRACe[:DATA]? <waveform_name> DATA[:DATA]? <waveform_name>
Returned Parameters	<NR3> {,<NR3>} (a total of 1024 data points)
Related Commands	TRAC:DATA TRAC:DEL FUNC:SHAP

TRACe:CATalog?
DATA:CATalog?

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

Query Syntax	TRACe:CATalog? DATA:CATalog?
Parameters	None
Examples	`TRAC:CAT?` `DATA:CAT?`
Returned Parameters	<SRD>
Related Commands	TRAC: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, tha data from the first waveform name is copied to the second. When the number of points in the trace is sent, only the number 1024 is accepted.

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

Trigger Subsystem

This subsystem controls the triggering of the ac source. See Chapter 4 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.

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

Subsystem Syntax

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

ABORt

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

ABORt also resets the WTG bit in the Operation Condition Status register (see Chapter 4 under "Programming the Status Registers"). 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 Number	Description
1 (the default)	TRANsient	Step, pulse, or list transient trigger sequence
3	ACQuire	Measurement acquire trigger sequence

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

INITiate:CONTinuous:SEQuence
INITiate:CONTinuous:NAME

These commands control the transient generator trigger system as follows:

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

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

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

TRIGger

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

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

TRIGger:DELay

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

Command Syntax TRIGger[:SEQuence1]:DELay <NRf+>
TRIGger[:TRANsient]:DELay <NRf+>
Parameters 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 GPIB device, *TRG, or <GET> (Group Execute Trigger)
EXTernal ac source's backplane Trigger In BNC
IMMediate trigger is generated as soon as the trigger system is initiated.

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

TRIGger:SEQuence2:SOURce
TRIGger:SYNChronize:SOURce

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

IMMediate Starts the transient output immediately, unless a delay time other than 0 has been specified by TRIGger:DELay. In this case the transient output starts after the expiration of the delay time.
PHASe Starts the transient output at the reference phase set by TRIG:SYNC:PHAS.

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

TRIGger:SEQuence2:PHASe
TRIGger:SYNCHronize:PHASe

These commands set the phase angle with respect to an internal phase reference 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 (degrees) | MAXimum | MINimum
***RST Value**	0
Examples	`TRIG:SYNC:PHAS 90` `TRIG:SEQ2:PHAS 180`
Query Syntax	TRIGger:SEQuence2:PHASe? TRIGger:SYNChronize:PHASe?
Returned Parameters	<NR3>
Related Commands	ABOR TRIG TRIG:DEL TRIG:SYNC INIT INIT:CONT *TRG WAI

TRIGger:SEQuence3
TRIGger:ACQuire

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

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

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

TRIGger:SEQuence3:SOURce
TRIGger:ACQuire:SOURce

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

BUS GPIB device, *TRG, or <GET> (Group Execute Trigger)
EXTernal ac source's backplane Trigger In BNC
TTLTrg the signal driving the Trigger Out BNC

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

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

These commands define the names that are aliased to trigger sequences 1, 2 and 3. The command accepts only TRANsient for sequence 1, 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
***RST Value** BUS
Examples
```
TRIG:SEQ1:DEF TRAN      TRIG:SEQ3:DEF ACQ
```
Query Syntax TRIGger:SEQuence1:DEFine?
TRIGger:SEQuence2:DEFine?
TRIGger:SEQuence3:DEFine?
Returned Parameters <CRD>

Common Commands

Common commands begin with an * and consist of three letters (command) IEEE 488.2 standard to perform some common interface functions. The Keysight ac sources respond to the required common commands that control status reporting, synchronization, and internal operations. The ac sources also respond to optional common commands that control triggers, power-on conditions, and stored operating parameters.

Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data or status specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately. The description for each common command or query specifies any status registers affected. Refer to Chapter 4 under “Programming the Status 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 4 under “Programming the Status 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 Chapter 4 under “Programming the Status Registers” for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

Bit Configuration of Standard Event Status Enable Register

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

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

Command Syntax *ESE <NRf>
Parameters 0 to 255
Power-On Value see *PSC
Examples `*ESE 129`
Query Syntax *ESE?
Returned Parameters <NR1>
Related Commands *ESR? *PSC *STB?

*ESR?

This query reads the Standard Event Status Event register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see *ESE). See Chapter 4 under “Programming the Status Registers” for a detailed explanation of this register.

Query Syntax *ESE?
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 *ESE?
Parameters None
Returned Parameters <AARD>

Field	Information
Keysight Technologies	manufacturer
xxxxA	model number
nnnnA-nnnnn	serial number or 0
<R>.xx.xx	firmware revision

Example

```
Keysight Technologies, 6812B, 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 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 ac source. The *OPC command provides notification that all overlapped commands have been completed.
- all triggered actions are completed and the trigger system returns to the Idle state.

*OPC does not prevent processing of subsequent commands but Bit 0 will not be set until all pending operations are completed. The query causes the interface to place an ASCII "1" in the Output Queue when all pending operations are completed.

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

*OPT?

This query requests the ac source to identify any options that are installed. Options are identified by 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 as follows (see Chapter 4 under “Programming the Status Registers” for register details):

1 or **ON** Prevents the register contents from being saved, causing them to be cleared at power-on. This prevents a PON event from clearing SRQ at power-on.

0 or **OFF** Saves the contents of the Service Request Enable and the Standard Event Status enable registers in non-volatile memory and recalls them at power-on. This allows a PON event to generate SRQ at power-on.

The query returns the current state of *PSC.

Command Syntax *PSC <Bool>
Parameters 0 | 1 | OFF | ON
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)

NOTE: 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 to 15
Example
```
*RCL 3
```
Related Commands *PSC *RST *SAV

*RST

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

CAL:STAT	OFF	[SOUR:]FUNC	SIN
DISP:STAT	ON	[SOUR:]FUNC:CSIN	100%
DISP:MODE	TEXT	[SOUR:]LIST:COUN	1
INIT:CONT	OFF	[SOUR:]LIST:STEP	AUTO
INST:COUP	ALL	[SOUR:]PHAS	ϕ1=0 ϕ2=240 ϕ3=120
INST:NSEL	1	[SOUR:]PHAS:TRIG	ϕ1=0 ϕ2=240 ϕ3=120
OUTP	OFF	[SOUR:]PHAS:MODE	FIX
OUTP:COUP	AC	[SOUR:]PULS:COUN	1
OUTP:DFI	OFF	[SOUR:]PULS:DCYC	50%
OUTP:DFI:SOUR	OFF	[SOUR:]PULS:HOLD	WIDT
OUTP:IMP	OFF	[SOUR:]PULS:PER	1
OUTP:IMP:REAL	0	[SOUR:]PULS:WIDT	0.01667s
OUTP:IMP:REAC	0	[SOUR:]VOLT	1
OUTP:PROT:DEL	100ms	[SOUR:]VOLT:TRIG	1
OUTP:RI:MODE	OFF	[SOUR:]VOLT:MODE	FIX
OUTP:TTLT	OFF	[SOUR:]VOLT:OFFS	0
OUTP:TTLT:SOUR	BOT	[SOUR:]VOLT:OFFS:MODE	FIX
SENS:CURR:ACDC:RANG	MAX	[SOUR:]VOLT:OFFS:TRIG	0
SENS:SWE:OFFS:POIN	0	[SOUR:]VOLT:OFFS:SLEW	INF
SENS:SWE:TINT	25μs	[SOUR:]VOLT:OFFS:SLEW:MODE	FIX
SENS:WIND	KBESsel	[SOUR:]VOLT:OFFS:SLEW:TRIG	INF
[SOUR:]CURR	MAX / 1A[1]	[SOUR:]VOLT:PROT	MAX
[SOUR:]CURR:PEAK	6.5A / 13A / 26A [2]	[SOUR:]VOLT:PROT:STAT	OFF
[SOUR:]CURR:PEAK:TRIG	6.5A / 13A / 26A [2]	[SOUR:]VOLT:RANG	MAX
[SOUR:]CURR:PEAK:MODE	FIX	[SOUR:]VOLT:SENS:DET	RTIME
[SOUR:]CURR:PROT:STAT	OFF	[SOUR:]VOLT:SENS:SOUR	INT
[SOUR:]FREQ	60Hz	[SOUR:]VOLT:SLEW	INF
[SOUR:]FREQ:MODE	FIX	[SOUR:]VOLT:SLEW:MODE	FIX
[SOUR:]FREQ:SLEW	INF	[SOUR:]VOLT:SLEW:TRIG	INF
[SOUR:]FREQ:SLEW:MODE	FIX	TRIG:DEL	0
[SOUR:]FREQ:SLEW:TRIG	INF	TRIG:SOUR	BUS
[SOUR:]FREQ:TRIG	60Hz	TRIG:SEQ2:SOUR	IMM
[SOUR:]FUNC:MODE	FIX	TRIG:SEQ2:PHAS	0
[SOUR:]FUNC:TRIG	SIN	TRIG:SEQ3:SOUR	BUS

[1]Max for Keysight 6811B/6812B/6813B; 1A for Keysight 6814B/6834B/6843A
[2]6.5A for Keysight 6811B; 13A for Keysight 6812B; 26A for Keysight 6813B

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

*SRE

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (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 4 under “Programming the Status 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 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 4 under “Programming the Status 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 has one or more reasons for requesting service.

Bit Configuration of Status Byte Register

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

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?
Parameters None
Returned Parameters <NR1> (register value)
Related Commands *SRE *ESR *ESE

*TRG

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

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

*TST?

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

Query Syntax TST?
Parameters None
Returned Parameters <NR1> 0 indicates the ac source has passed selftest.
Non-zero indicates an error code (see appendix C)

*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 . 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 a GPIB DCL (Device Clear) command.

Command Syntax	WAI?
Parameters	None
Related Commands	*OPC

4

Programming Examples

Introduction

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

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

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

Programming the Output

Power-on Initialization

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

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

*RST is a convenient way to program all parameters to a known state. Refer to the *RST command in Chapter 3 to see how each programmable parameter is set by *RST. Refer to the *PSC command in Chapter 3 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
```

NOTE: In the three-phase model, 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 3.

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.

For Keysight models 6814B, 6834B and 6843A, you cannot program a voltage that produces a higher volt-second on the output than a 300 Vrms sinewave.

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

Voltage Ranges (Keysight 6814B/6834B/6843A only)

The Keysight 6814B, 6834B and 6843A have 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).

NOTE: On the Keysight 6814B, 6834B and 6843A, 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 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".

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

Frequency Slew

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

```
FREQuency:SLEW 30
```

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

Waveform Shapes

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

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

Square Waveform

To select the square output waveform, use:

```
FUNCtion:SHAPe SQUare
```

Clipped Waveform

To select a clipped sine waveform use:

```
FUNCtion:SHAPe CSINusoid
```

To set the clipping level to 50%, use:

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

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

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

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

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

User-Defined Waveform

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

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

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

Data points can be in any arbitrary units. The ac source scales the data to an internal format that removes the dc component and ensures that the correct ac rms voltage is output when the waveform is selected. When queried, trace data is returned as normalized values in the range of 1. Waveform data is stored in nonvolatile memory and is retained when input power is removed. Up to 12 user defined waveforms may be created and stored.

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

Individual Phases (Keysight 6834B only)

In the Keysight 6834B model, 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 model. 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.

NOTE: The INSTrument:COUPle command has no effect on queries. In the Keysight 6834B three-phase model, 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 Keysight 6834B three-phase model, 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.

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

Current Limit

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

```
CURRent <n>
```

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

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

The ac source can be programmed to turn off its output if the rms current limit is reached. This protection feature is implemented with the CURRent:PROTection:STATe command as explained in Chapter 3.

NOTE: On the Keysight 6814B, 6834B and 6843A, 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.

Peak Current Limit (Keysight **6811B/6812B/6813B only)**

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

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

where <n> is the peak current in amperes.

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

DC Output (Keysight 6811B/6812B/6813B only)

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

To enable the dc output, use:

```
OUTPut:COUPling DC
```

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

```
VOLTage:OFFSet 5
```

When the command

```
OUTPut:COUPling AC
```

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

NOTE: Because the voltage offset commands are coupled with the voltage commands, the dc output voltage will affect the maximum output voltage and vice-versa. When the dc output is non-zero, the maximum ac voltage that can be programmed is reduced to a value that limits the total ac + dc peak amplitude to 425 V. Similarly, when the ac output is non-zero the maximum dc offset that can be programmed is subject to the same limitation. Refer to "Coupled Commands" for more information.

Coupled Commands

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

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

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

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

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

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

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

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

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

CURRent:LEVel and VOLTage:RANGe (Keysight **6814B/6834B/6843A only)**

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 the above models.

Programming Output Transients

Output transients are used to:

- Synchronize output changes with a particular phase of the voltage waveform.
- Synchronize output changes with internal or external trigger signals.
- Simulate surge, sag, and dropout conditions with precise control of duration and phase.
- Create complex, multi-level sequences of output changes.
- Create output changes that have rapid or precise timing requirements.

The following ac source functions are subject to transient control:

AC output voltage

Frequency

Phase

Waveform shape

AC voltage slew rate

Frequency slew rate

DC output voltage (Keysight 6811B/6812B/6813B only)

Peak current limit (Keysight 6811B/6812B/6813B only)

The following transient modes can be generated:

STEP	Generates a single triggered output change.
PULSe	Generates an output change which returns to its original state after some time period.
LIST	Generates a sequence of output changes, each with an associated dwell time or paced by triggers.
FIXed	Turns off the transient functions, which means that only the IMMediate values are used as the data source for a particular function.

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

Transient System Model

Figure 4-1 is a model of the transient system. The figure shows the transient modes and the source of the data that generates each mode.

When a trigger is received in step or pulse modes, the triggered functions are set from their IMMediate to their TRIGgered value. In Step mode, the triggered value becomes the immediate value. In Pulse mode, the functions return to their immediate value during the low portion of the pulse. If there are no further pulses, the immediate value remains in effect. In List mode, the functions return to their immediate value at the completion of the list.

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

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

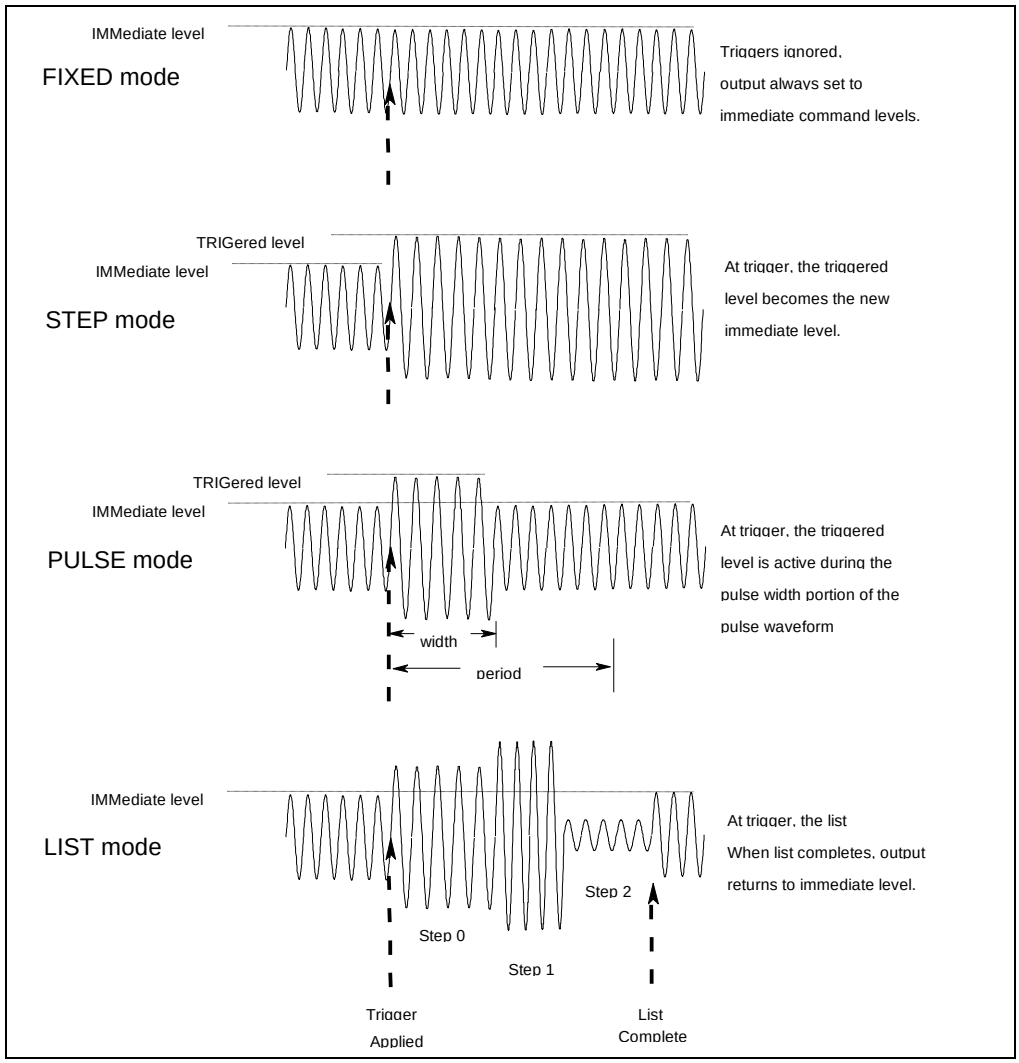


Figure 4-1. Model of Transient System

Step and Pulse Transients

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

```
*RST
VOLTage:MODE STEP
```

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

```
VOLTage:TRIGger 150
```

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

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

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

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

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

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

```
INITiate:IMMediate:SEQuence1
```

To enable the transient system indefinitely use:

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

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

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 GPIB bus trigger.

`*RST`	Begin at power-on state
`VOLT 120`	Set initial output voltage (immediate-level)
`FREQ 60`	Set initial output frequency
`OUTP ON`	Enable the output
`VOLT:MODE PULS`	Enable output to generate pulses when triggered
`VOLT:TRIG 0`	Set the voltage dropout (triggered level)
`PULS:WIDT .03333`	Set pulse width for 2 periods
`TRIG:SOUR BUS`	Respond to GPIB bus triggers
`TRIG:SYNC:SOUR PHAS`	Synchronize triggers to internal phase reference
`TRIG:SYNC:PHAS 90`	Sets internal phase reference point to 90 degrees
`INIT:SEQ1`	Set to Wait-for-trigger state
`<device trigger>`	Send the GPIB bus trigger

List Transients

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

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

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

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

Step 2 Program the list of output values for each function. The list commands take a comma-separated list of arguments. The order in which the arguments are given determines the sequence in which the values will be output. For example, to cycle the voltage through a sequence that includes nominal line, high line, and low line, a list may include the following values:

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

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

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

All lists must have the same number of data values or points, or an error will occur when the transient system that starts the sequence is later initiated. The exception is when a list has only one item or point. In this case the single-item list is treated as if it had the same number of points as the other lists, with all values being equal to the one item. For example:

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

is the same as:

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

Step 3 Determine the time interval that the output remains at each level or point in the list before it advances to the next point. The time is specified in seconds. For example, to specify five dwell intervals, use:

```
LIST:DWELl 1, 1.5, 2, 2.5, 3
```

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

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

```
LIST:COUNt 10
```

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

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

```
LIST:STEP AUTO
```

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

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

```
LIST:STEP ONCE
```

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

Step 6 Use the transient trigger system to trigger the list. This is described in detail under "Triggering Output Changes".

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

SCPI Triggering Nomenclature

In SCPI terms, trigger systems are called sequences. When more than one trigger system exists, they are differentiated by naming them SEQuence1, SEQuence2, ... etc. In the ac source, SEQuence1 is the transient trigger system, SEQuence2 is the phase synchronization trigger system, and SEQuence3 is the measurement trigger system.

The ac source uses aliases with more descriptive names for these sequences. These aliases can be used instead of the sequence forms.

Sequence Form	Alias
SEQuence1	TRANsient
SEQuence2	SYNChronize
SEQuence3	ACQuire

Output Trigger System Model

Figure 4-2 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.

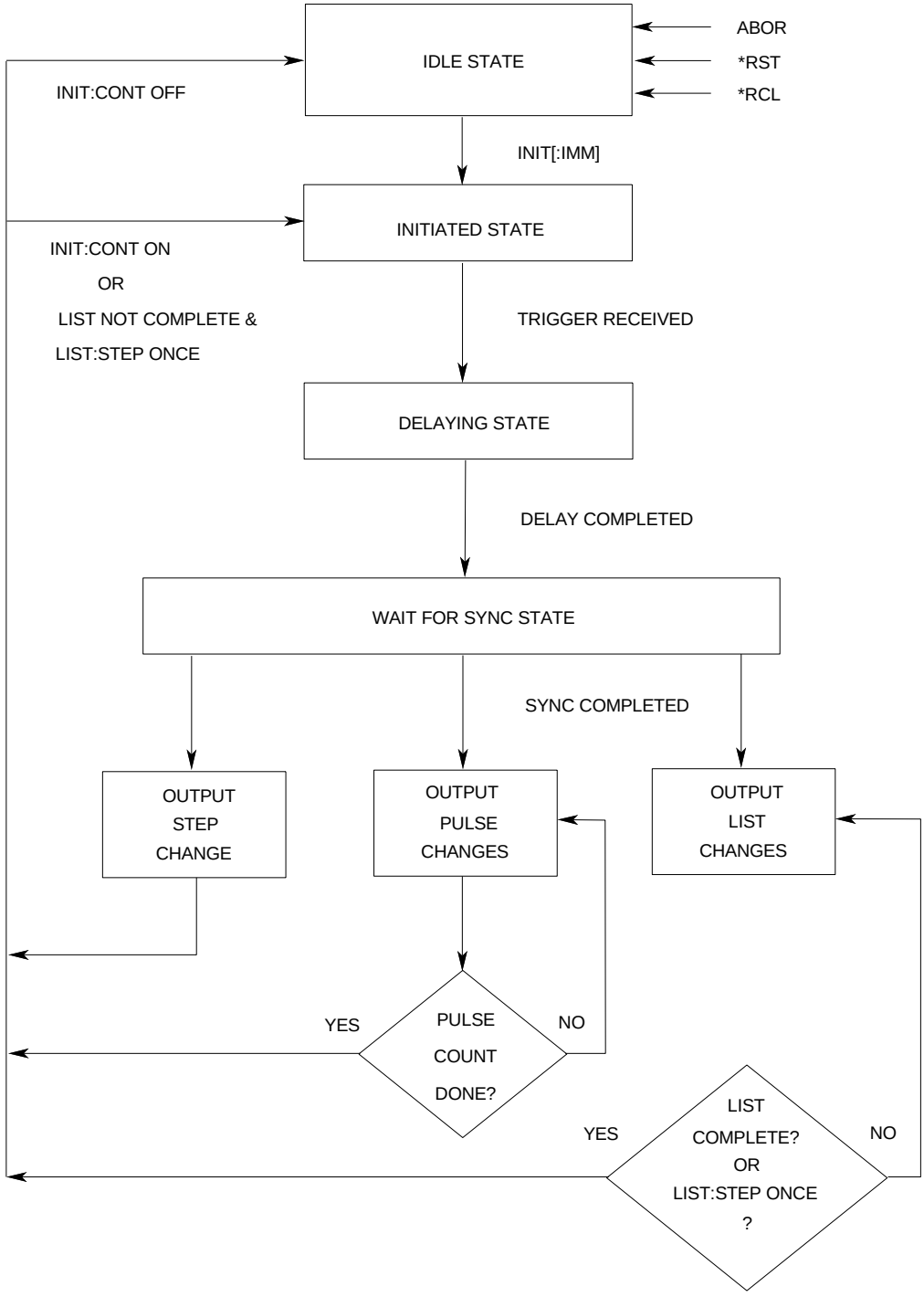


Figure 4-2. Model of Output Triggers

Initiating the Output Trigger System

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

```
ABORt   *RST   *RCL
```

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

```
INITiate:IMMediate:SEQuence1      or
INITiate:IMMediate:NAME TRANsient
```

After a trigger is received and the action completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered action is desired.

To keep a trigger system initiated for multiple actions without having to send an initiate command for each trigger, use:

```
INITiate:CONTinuous:SEQuence1 ONor
INITiate:CONTinuous:NAME TRANsient, ON
```

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

Selecting the Output Trigger Source

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

```
TRIGger:SEQuence1:SOURce EXTernal      or
TRIGger:TRANsient:SOURce EXTernal
```

To select GPIB bus triggers (group execute trigger, device trigger, or *TRG command), use:

```
TRIGger:SEQuence1:SOURce BUS      or
TRIGger:TRANsient:SOURce BUS
```

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

```
TRIGger:SEQuence1:SOURce IMMediate      or
TRIGger:TRANsient:SOURce IMMediate
```

NOTE: The immediate source can be combined with INITiate:CONTinuous:SEQuence1 ON to generate repetitive output transients.

A transition from the Initiated state to the Delay state is made when the trigger signal is received.

Specifying a Trigger Delay

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

```
TRIGger:SEQuence1:DELay .01or
TRIGger:TRANsient:DELay .01
```

which sets a delay time of 10 milliseconds.

NOTE: A trigger delay can only be programmed for SEQuence1 (or TRANsient) triggers.

When the programmed trigger delay has elapsed, the trigger system transitions from the Delay state to the Wait-for-sync state.

Synchronizing Output Changes to a Reference Phase Angle

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

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

```
TRIGger:SEQuence2:SOURce PHASeor
TRIGger:SYNChronize:SOURce PHASe
```

To select the desired phase, use:

```
TRIGger:SEQuence2:PHASe 90or
TRIGger:SYNChronize:PHASe 90
```

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

To turn off transient phase synchronization, use:

```
TRIGger:SYNChronous:SOURce IMMediate
```

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

Generating Output Triggers

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

Single Triggers

- By sending one of the following over the GPIB:

  ```
  TRIGger: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 GPIB:

  ```
  TRIGger:SEQuence1:SOURce IMMediate
  INITiate:CONTinuous:SEQuence1 ON
  ```

When the trigger system enters the Output Change state upon receipt of a trigger (see figure 4-2), the triggered functions are set to their programmed trigger levels. When the triggered actions are completed, the trigger system returns to the Idle state.

Specifying a Dwell Time for Each List Point

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

```
LIST:DWELl <n> {,<n>}
```

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

At the end of the dwell interval, the transition to the next state depends on whether or not the list has completed its sequencing and the state of the LIST:STEP command (see figure 4-2).

If the list is completed, the trigger system returns to the Idle state.

- If the list is not completed, then the system reacts as follows:

  `LIST:STEP ONCE` programs the trigger system to return to the Initiated state to wait for the next trigger.

  `LIST:STEP AUTO` programs the trigger system to immediately execute the next list point.

Making Measurements

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

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

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

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

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

Voltage and Current Measurements

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

```
MEASure:VOLTage:AC?      or
MEASure:CURRent:AC?
```

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

```
MEASure:VOLTage:ACDC?      or
MEASure:CURRent:ACDC?
```

To measure the dc voltage or current components, use:

```
MEASure:VOLTage:DC?      or
MEASure:CURRent:DC?
```

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

```
MEASure:CURRent:AMPLitude:MAXimum?
MEASure:CURRent:CRESTfactor?
```

Power Measurements

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

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

Harmonic Measurements

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

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

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

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

These queries always return 51 data values, from the dc component up to the 50th harmonic. Any harmonics that represent frequencies greater than 12.5kHz are returned as the value 0. To return the percentage of total harmonic distortion in the output voltage or current, use the following commands:

```
MEASure:CURRent:HARMonic:THD?
MEASure:VOLTage:HARMonic:THD?
```

Simultaneous Output Phase Measurements (Keysight 6834B only)

You can return simultaneous measurements from all output phases of the three-phase ac source using the FETCh query. Unlike MEASure queries, FETCh queries do not trigger the acquisition of new data when they are executed.

You must first 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?
```

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?
```

Regulatory-Compliant Measurement of Quasi-Stationary Harmonics

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

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

Triggering Measurements

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

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

SCPI Triggering Nomenclature

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

Sequence Form	**Alias**
SEQuence3	ACQuire

Measurement Trigger System Model

Figure 4-2 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.

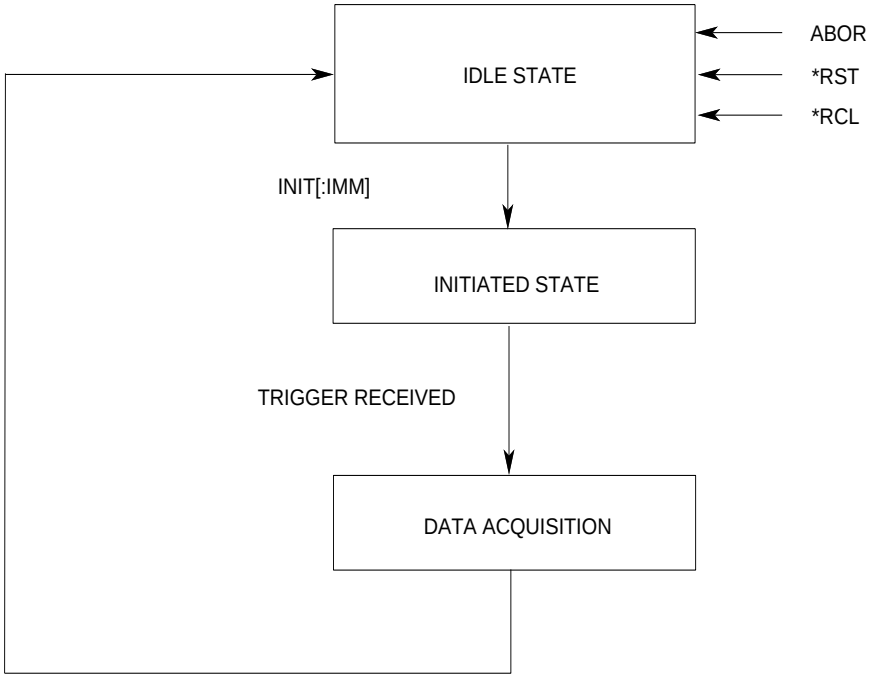


Figure 4-3. Model of Measurement Triggers

Initiating the Measurement Trigger System

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

```
ABORt   *RST   *RCL
```

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

```
INITiate:IMMediate:SEQuence3      or
INITiate:IMMediate:NAME ACQuire
```

After a trigger is received and the data acquisition completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered acquisition is desired.

NOTE: You cannot initiate measurement triggers continuously. Otherwise, the measurement data in the data buffer would continuously be overwritten by each triggered measurement.

Selecting the Measurement Trigger Source

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

```
TRIGger:SEQuence3:SOURce EXTernal      or
TRIGger:ACQuire:SOURce EXTernal
```

To select GPIB bus triggers (group execute trigger, device trigger, or *TRG command), use:

```
TRIGger:SEQuence3:SOURce BUS      or
TRIGger:ACQuire:SOURce BUS
```

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

```
TRIGger:SEQuence3:SOURce TTLTrg      or
TRIGger:ACQuire:SOURce TTLTrg
```

Generating Measurement Triggers

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

- By sending one of the following over the GPIB:

  ```
  TRIGger:SEQuence3:IMMediate
  TRIGger:ACQuire:IMMediate
  *TRG
  ```

  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.

Controlling the Instantaneous Voltage and Current Data Buffers

Varying the Voltage and Current Sampling Rate

At *RST, the output voltage and current sampling rate is 40kHz (period = 25 μs). 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.

Pre-event and Post-event Triggering

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

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

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

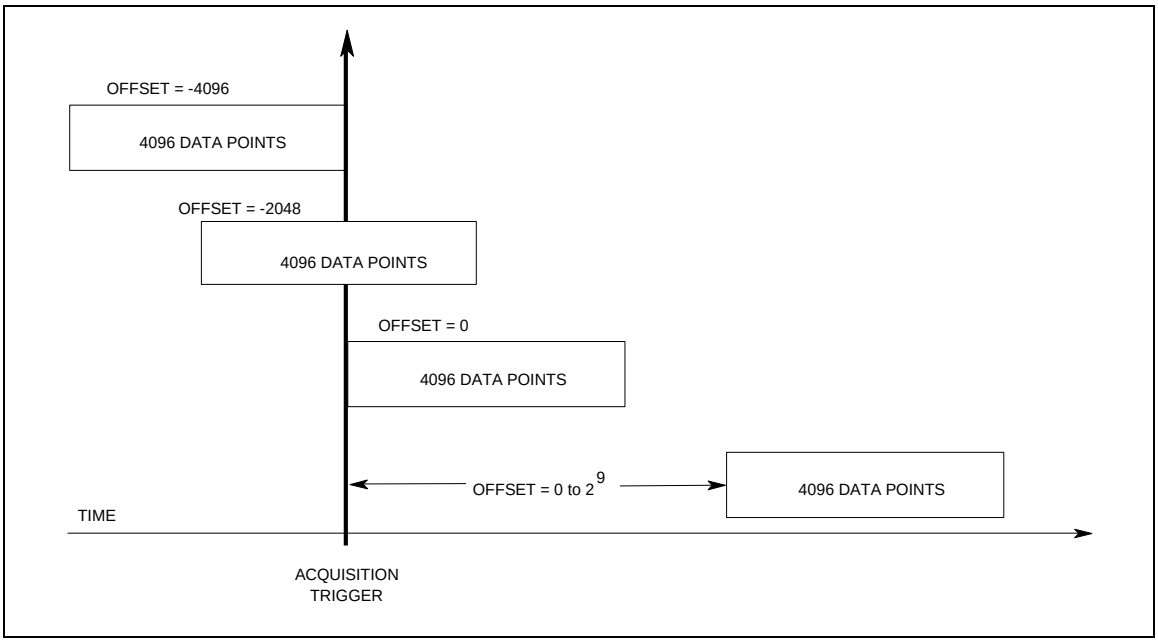


Figure 4-4. Pre-event and Post-event Triggering

Programming the Status Registers

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

Figure 4-5 shows the status register structure of the ac source. Table 4-1 defines the status bits. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard GPIB functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The Operation Status, Questionable Status, and Questionable Instrument Isummary Status registers implement functions that are specific to the ac source.

Power-On Conditions

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

Operation Status Group

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

Register	Command	Description
Condition	`STAT:OPER:COND?`	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	`STAT:OPER:PTR <n>`	A positive transition filter that functions as described under STAT:OPER:NTR|PTR commands in Chapter 3. It is a read/write register.

NTR Filter	`STAT:OPER:NTR <n>`	A negative transition filter that functions as described under STAT:OPER:NTR|PTR commands in Chapter 3. 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.

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

Table 4-1. Bit Configuration of Status Registers

Bit	Signal	Meaning
		Operation Status Group
0	CAL	Interface is computing new cal constants
5	WTG	Interface is waiting for a trigger
8	CV	The output voltage is regulated
		Questionable and Questionable Instrument Isummary Status Groups
0	OV	The overvoltage protection circuit has tripped
1	OCP	The overcurrent protection circuit has tripped
2	SOA	The safe operating area protection has tripped (Keysight 6811B/6812B/6813B) The
3	UNR	output is unregulated
4	OT	An overtemperature condition has occurred
9	RI	The remote inhibit state is active
10	CLpeak	The peak current limit circuit is active (Keysight 6811B/6812B/6813B)
11	Rail	The rail protection circuit has tripped (Keysight 6811B/6812B/6813B)
		The rail is unregulated (Keysight 6814B/6834B/6843A)
12	CLrms	The rms current limit circuit is active
13	Isum	Summary of QUES:INST:ISUM registers (Keysight 6834B)
14	MeasOvld	Current measurement exceeded capability of low range (Keysight 6811B/6812B/6813B)
		Standard Event Status Group
0	OPC	Operation complete
2	QYE	Query error
3	DDE	Device-dependent error
4	EXE	Execution error
5	CME	Command error
7	PON	Power-on
		Status Byte and Service Request Enable Registers
3	QUES	Questionable status summary bit
4	MAV	Message Available summary bit
5	ESB	Event Status Summary bit
6	MSS	Master Status Summary bit
	RQS	Request Service bit
7	OPER	Operation status summary bit

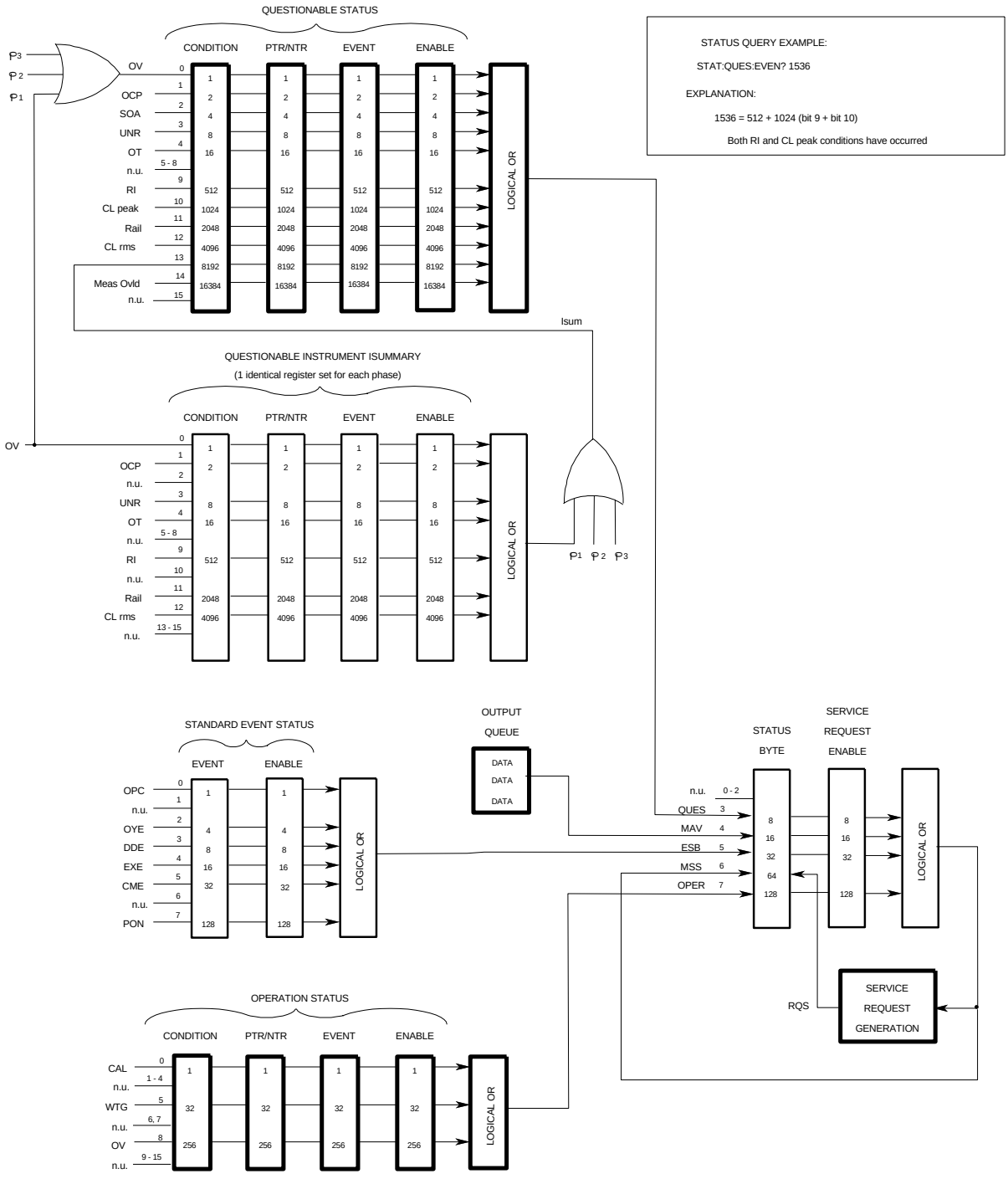


Figure 4-5. Ac Source Status Model

Questionable Status Group

The Questionable Status registers record signals that indicate abnormal operation of the ac source. As shown in the figure 4-5, the group consists of the same type of registers as the Status Operation group.

Register	Command	Description
Condition	`STAT:QUES:COND?`	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	`STAT:QUES:PTR <n>`	A positive transition filter that functions as described under STAT:QUES:NTR|PTR commands in Chapter 3. It is a read/write register.
NTR Filter	`STAT:QUES:NTR <n>`	A negative transition filter that functions as described under STAT:QUES:NTR|PTR commands in Chapter 3. It is a read/write register.
Event	`STAT:QUES:EVEN?`	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	`STAT:QUES:ENAB? <n>`	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

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

NOTE: In a **three-phase ac source**, each signal that is fed into the Questionable Status Condition register is logically-ORed from three corresponding status signals that originate from each phase. Figure 4-5 illustrates this for the OV bit; the same illustration also applies to the other bits in the Condition register.

Questionable Instrument Isummary Status Group

Although only one group of Questionable Instrument Isummary Status registers is shown in figure 4-4, there are actually three identical register groups. With the exception of bit 13, the register structure is the same as the Questionable Status group. These three register groups monitor the status signals of each individual phase of the three-phase ac source. To determine which phase of the ac source is currently selected, use:

```
INSTrument:NSELect?
```

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

```
INSTrument:NSELect <n>
```

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

Register	Command	Description
Condition	STAT:QUES:INST:ISUM:COND?	A register that holds real-time status of the circuits being monitored. It is a read-only register.
PTR Filter	STAT:QUES:INST:ISUM:PTR <n>	A positive transition filter that functions as described under STAT:QUES:INST:ISUM:NTR|PTR commands in Chapter 3. It is a read/write register.
NTR Filter	STAT:QUES:INST:ISUM:NTR <n>	A negative transition filter that functions as described under STAT:QUES:INST:ISUM:NTR|PTR commands in Chapter 3. It is a read/write register.
Event	STAT:QUES:INST:ISUM:EVEN?	A register that latches any condition that is passed through the PTR or NTR filters. It is a read-only register that is cleared when read.
Enable	STAT:QUES:INST:ISUM:ENAB? <n>	A register that functions as a mask for enabling specific bits from the Event register. It is a read/write register.

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

Standard Event Status Group

This group consists of an Event register and an Enable register that are programmed by Common commands. The Standard Event register latches events relating to interface communication status (see figure 4-5). It is a read-only register that is cleared when read. The Standard Event Enable register functions similarly to the enable registers of the Operation and Questionable status groups.

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

The PON (Power On) Bit

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

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

Status Byte Register

This register summarizes the information from all other status groups as defined in the *IEEE 488.2 Standard Digital Interface for Programmable Instrumentation*. The bit configuration is shown in Table 4-1.

Command	Action
*STB? -	reads the data in the register but does not clear it (returns MSS in bit 6)
serial poll -	reads and clears the data in the register (returns RQS in bit 6)

The MSS Bit

This is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the ac source has one or more reasons for requesting service. *STB? reads the MSS in bit position 6 of the response but does not clear any of the bits in the Status Byte register.

The RQS Bit

The RQS bit is a latched version of the MSS bit. Whenever the ac source requests service, it sets the SRQ interrupt line true and latches RQS into bit 6 of the Status Byte register. When the controller does a serial poll, RQS is cleared inside the register and returned in bit position 6 of the response. The remaining bits of the Status Byte register are not disturbed.

The MAV bit and Output Queue

The Output Queue is a first-in, first-out (FIFO) data register that stores ac source-to-controller messages until the controller reads them. Whenever the queue holds one or more bytes, it sets the MAV bit (4) of the Status Byte register.

Examples

Determining the Cause of a Service Interrupt

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

Step 1 Determine which summary bits are active. Use:

```
*STB?
```

or

serial poll

Step 2 Read the corresponding Event register for each summary bit to determine which events caused the summary bit to be set. Use:

```
STATus:QUEStionable:EVENt?
STATus:OPERation:EVENt?
ESR?
```

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

Step 3 Remove the specific condition that caused the event. If this is not possible, the event may be disabled by programming the corresponding bit of the status group Enable register or NTR|PTR filter. A faster way to prevent the interrupt is to disable the service request by programming the appropriate bit of the Service Request Enable register.

Servicing Questionable Status Events

This example assumes you want a service request generated whenever the ac source's overvoltage, overcurrent, or overtemperature circuits have tripped. From figure 4-5, note the required path for Questionable Status conditions at bits 0, 1, and 4 to generate a service request (RQS) at the Status Byte register. The required register programming is as follows:

Step 1 Program the Questionable Status PTR register to allow a positive transition at bits 0, 1, or 4 to be latched into the Status Event register. Use:

```
STATus:QUEStionable:PTR 19
```
(1 + 2 + 16 = 19)

Step 2 Program the Questionable Status Enable register to allow the latched events to be summed into the QUES summary bit. Use:

```
STATus:QUEStionable:ENABle 19
```

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

```
*SRE 8
```

Step 4 When you service the request, read the event register to determine which Questionable Status Event register bits are set and clear the register for the next event. Use:

```
STATus:QUEStionable:EVENt?
```

Monitoring Both Phases of a Status Transition

You can monitor a status signal for both its positive and negative transitions. For example, to generate RQS when the ac source either enters the CLrms (rms current limit) condition or leaves that condition, program the Questionable Status PTR/NTR filter as follows:

```
STATus:QUEStionable:PTR 4096;NTR 4096
STATus:QUEStionable:ENABle 4096;*SRE 8
```

The PTR filter will cause the QUES summary bit to set RQS when CLrms occurs. When the controller subsequently reads the event register with STATus:QUEStionable:EVEN?, the register is cleared. When CLrms subsequently goes false, the NTR filter causes the QUES summary bit to again set RQS.

Programming the Trigger In and Trigger Out BNC Connectors

The ac source has two bnc connectors labeled Trigger In and Trigger Out (see figure 4-6). Refer to "Operating Characteristics" in appendix A of the User's Guide for the electrical parameters.

Trigger In BNC

This chassis-referenced digital input can be selected as a source for transient or measurement triggers. This allows an action to be synchronized to an external signal. The trigger is recognized on a high-to-low transition of the input signal. The minimum pulse width of the signal is 1 microsecond. To select the Trigger In connector as the trigger source, use:

```
TRIGger:SEQuence1:SOURce EXTernal
```
or
```
TRIGger:TRANsient:SOURce EXTernal
```

```
TRIGger:SEQuence3:SOURce EXTernal
```
or
```
TRIGger:ACQuire:SOURce EXTernal
```

Trigger Out BNC

This chassis-referenced digital output can be programmed to supply a pulse output at the leading or trailing edge of a step or pulse, or at the leading edge of any point in a list sequence. The output signal is nominally a 10 microsecond low-true pulse. To enable the Trigger Out connector, use:

```
OUTPut:TTLTrg:STATe ON
```

At *RST, the Trigger Out connector is off.

To select a trigger source for the Trigger Out connector, use:

```
OUTPut:TTLTrg:SOURce BOT | EOT | LIST
```

BOT - specifies that the pulse is output at the beginning of a transient. This is the *RST setting.
EOT - specifies that the pulse is output at the end of a transient.
LIST - specifies that the pulse position is defined by the LIST:TTLTrg command.

You can also specify the Trigger Out connector as a trigger source for measurement trigger sequences. Use:

```
TRIGger:SEQuence3:SOURce TTLTor
TRIGger:ACQuire:SOURce TTLT
```

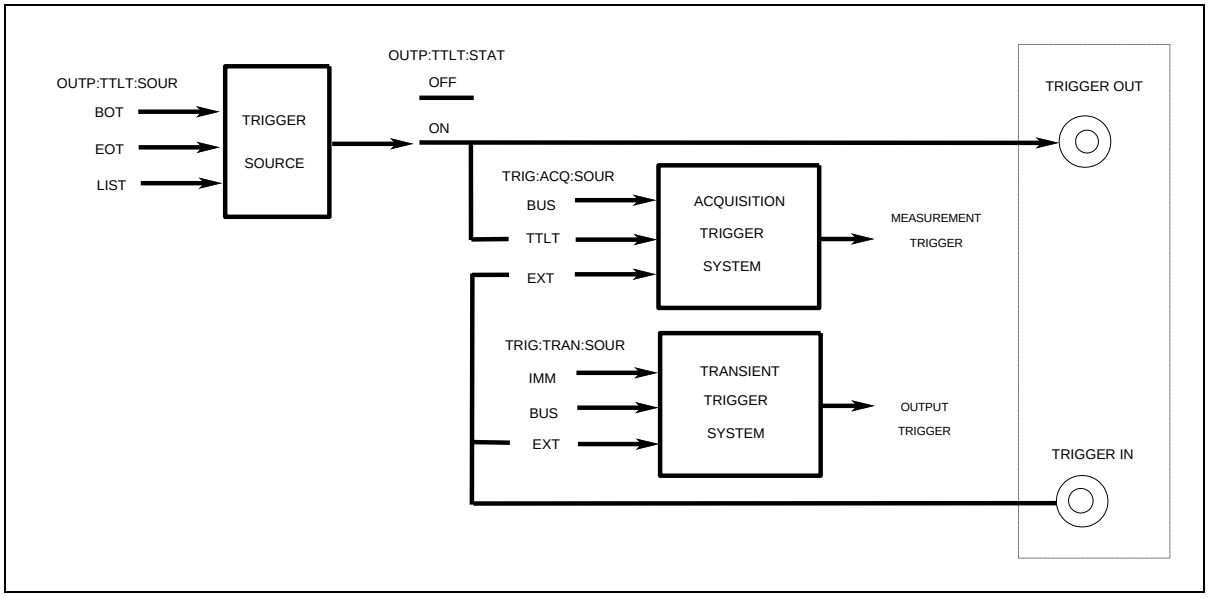


Figure 4-6. BNC Connector Trigger Model

Remote Inhibit and Discrete Fault Indicator

The remote inhibit and discrete fault indicators are implemented through the respective INH and FLT connections on the rear panel. Refer to "Operating Characteristics" in appendix A of the User's Guide for the electrical parameters.

Remote Inhibit (RI)

Remote inhibit is an external logic signal routed through the rear panel INH connection, which allows an external device to signal a fault. To select an operating modes for the remote inhibit signal, use:

```
OUTPut:RI:MODE LATChing | LIVE | OFF
```

LATChing -causes a low-true signal on the INH input to disable the output. The only way to clear the latch is by sending an OUTPut:PROTection:CLEAR command while the INH input is false.
LIVE - allows the RI input to disable the output in a non-latching manner. When INH is low true, the output is disabled. When INH is high, it has no effect on the output.
OFF - disables the INH input.

Discrete Fault Indicator (DFI)

The discrete fault indicator is a chassis-referenced, open-collector logic signal connected to the rear panel FLT connection, that can be used to signal external devices when a fault condition is detected. To select the internal fault source that drives this signal, use:

```
OUTPut:DFI:SOURce QUEStionable | OPERation | ESB | RQS | OFF
```

QUEStionable - selects the Questionable event summary bit (bit 3 of the Status Byte Register)
OPERation - selects the Operation Event summary bit (bit 7 of the Status Byte Register)
ESB - selects the Standard Event summary bit (bit 5 of the Status Byte Register)
RQS - selects the Request Service bit (bit 6 of the Status Byte Register)
OFF - selects no DFI source

To enable or disable the DFI output, use:

```
OUTPut:DFI:STATe ON | OFF
```

SCPI Command Completion

SCPI commands sent to the ac source are processed either sequentially or in parallel. Sequential commands finish execution before a subsequent command begins. Parallel commands allow other commands to begin executing while the parallel command is still executing. Commands that affect list and trigger actions are among the parallel commands.

The *WAI, *OPC, and *OPC? common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. The syntax and parameters for these commands are described in Chapter 3. 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.

A

SCPI Command Tree

Command Syntax

ABORt

CALibrate
- :CURRent
  - :AC
  - :MEASure
- :DATA <n>
- :IMPedance
- :LEVel P1 | P2 | P3 | P4
- :PASSword <n>
- :PWM
  - :FREQuency <n>
  - :RAMP <n>
- :SAVE
- :STATe <bool> [, <n>]
- :VOLTage
  - :AC
  - :DC
  - :OFFSet
  - :PROTection
  - :RTIMe

DATA | TRACe
- :CATalog?
- [:DATA] <trace_name>, <n> {, <n>}
- :DEFine <trace_name>[,<trace_name> | 1024]
- :DELete
  - [:NAME] <trace_name>

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 | OUTPut3

FETCh | MEASure
- [:SCALar]
  - :CURRent
    - [:DC]?
    - :AC?
    - :ACDC?
    - :AMPLitude
      - :MAX?
    - :CRESTfactor?
    - :HARMonic
      - [:AMPLitude]? <n>
      - :PHASe? <n>
      - :THD?
    - :NEUTral
      - [:DC]?
      - :AC?
      - :ACDC?
      - :HARMonic
        - [:AMPLitude]? <n>
        - :PHASe? <n>
  - :FREQuency?
  - :POWer
    - [:DC]?
    - :AC
      - [:REAL]?
      - :APParent?
      - :REACtive?
      - :PFACtor?
      - :TOTal?
  - :VOLTage
    - [:DC]?
    - :AC?
    - :ACDC?
    - :HARMonic
      - [:AMPLitude]? <n>
      - :PHASe? <n>
      - :THD?
- :ARRay
  - :CURRent
    - [:DC]?
    - :HARMonic
      - [:AMPLitude]?
      - :PHASe?
    - :NEUTral
      - [:DC]?
      - :HARMonic
        - [:AMPLitude]?
        - :PHASe?
  - :VOLTage
    - [:DC]?
    - :HARMonic
      - [:AMPLitude]?
      - :PHASe?

OUTPut
- [:STATe] <bool>
- :COUPling DC | AC
- :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
- :TTLTrg
  - [: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>
- :PEAK
  - [:IMMediate] <n>
  - :MODE FIXed | STEP | PULSe | LIST
  - :TRIGgered <n>
- :PROTection
  - :STATe <bool>

FREQuency
- [:CW | :IMMediate] <n>
- :MODE FIXed | STEP | PULSe | LIST
- :SLEW
  - [:IMMediate] <n> | INFinity
  - :MODE FIXed | 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> | INFinity
- :CURRent <n> {, <n>}
  - :POINts?
- :DWELl <n> {,<n>}
  - :POINts ?
- :FREQuency
  - [:LEVel] <n> {,<n>}
    - :POINts?
  - :SLEW <n> {,<n>}
    - :POINts?
- :PHASe <n> {,<n>}
  - :POINts?
- :SHAPe <shape> {, <shape>}
  - :POINts?
- :STEP ONCE | AUTO
- :TTLTrg <bool> {,<bool>}
  - :POINts?
- :VOLTage
  - [:LEVel] <n> {,<n>}
    - :POINts?
  - :SLEW <n> {,<n>}
    - :POINts?
  - :OFFSet <n> {,<n>}
    - :POINts?
    - :SLEW <n> {,<n>}
      - :POINts?

PHASe
- [:IMMediate | :ADJust] <n>
- :MODE FIXed | STEP | PULSe | LIST
- :TRIGgered <n>

PULSe
- :COUNt <n> | INFinity
- :DCYCle <n>
- :HOLD WIDTh | DCYCle
- :PERiod <n>
- :WIDTh <n>

VOLTage
- [:LEVel]
  - [:IMMediate]
    - [:AMPLitude] <n>
  - :TRIGgered
    - [:AMPLitude] <n>
- :SENSe | :ALC
  - :DETector RTIMe | RMS
  - :SOURce INT | EXT
- :MODE FIXed | STEP | PULSe | LIST
- :OFFSet <n>
  - [:IMMediate] <n>
  - :MODE FIXed | STEP | PULSe | LIST
  - :TRIGgered <n>
  - :SLEW
    - [:IMMediate] <n> | INFinity
    - :MODE FIXed | STEP | PULSe | LIST
    - :TRIGgered <n> | INFinity
- :PROTection
  - [:LEVel] <n>
  - :STATe <bool>
- :RANGe 150 | 300
- :SLEW
  - [:IMMediate] <n> | INFinity
  - :MODE FIXed | STEP | PULSe | LIST
  - :TRIGgered <n> | INFinity

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

SYSTem
- :CONFigure
  - :NOUTputs 1 | 3
- :ERRor?
- :VERSion?
- :LANGuage SCPI | E9012
- :LOCal
- :REMote
- :RWLock

TRIGger
- [:TRANsient | :SEQuence1]
  - [:IMMediate]
  - :SOURce BUS | EXTernal | IMMediate
  - :DELay <n>
- :SYNChonize | :SEQuence2
  - :SOURce PHASe | IMMediate
  - :PHASe <n>
- :ACQuire | :SEQuence3
  - [:IMMediate]
  - :SOURce BUS | EXTernal | TTLTrg
- :SEQuence1
  - :DEFine TRANsient
- :SEQuence2
  - :DEFine SYNChronize
- :SEQuence3
  - :DEFine ACQuire

SCPI Common Commands

*CLS	*IDN?	*PSC <bool>	*SAV <value>	*TRG
*ESE <value>	*OPC	*PSC?	*SRE <value>	*TST?
*ESE?	*OPC?	*RCL <value>	*SRE?	*WAI
*ESR?	*OPT?	*RST	*STB?	

B

SCPI Conformance Information

The ac source responds to SCPI Version 1992.0

SCPI Confirmed Commands

ABOR
CAL:DATA
CAL:STAT
DISP[:WIND][:STAT]
DISP[:WIND]:TEXT[:DATA]
INIT[:IMM]
INIT[:IMM]:SEQ | NAME
INIT:CONT:SEQ | NAME
INST:COUP
INST:NSEL
MEAS | FETC[:SCAL]:CURR[:DC]?
MEAS | FETC[:SCAL]:CURR:AC?
MEAS | FETC[:SCAL]:FREQ?
MEAS | FETC[:SCAL]:POW[:DC]?
MEAS | FETC[:SCAL]:POW:AC[:REAL]?
MEAS | FETC[:SCAL]:VOLT[:DC]?
MEAS | FETC[:SCAL]:VOLT:AC?
MEAS | FETC:ARR:CURR[:DC]?
MEAS | FETC:ARR:VOLT[:DC]?
OUTP[:STAT]
OUTP:COUP
OUTP:IMP[:STAT]
OUTP:PROT:CLE
OUTP:PROT:DEL
OUTP:TTLT[:STAT]
SENS:CURR:ACDC:RANG[:UPP]
SENS:SWE:OFFS:POIN
SENS:SWE:TINT
SENS:WIND
[SOUR]:CURR[:LEV][:IMM][:AMPL]
[SOUR]:CURR:PROT:STAT
[SOUR]:FREQ[:CW | :IMM]
[SOUR]:FREQ:MODE
[SOUR]:FUNC[:SHAP][:IMM]
[SOUR]:LIST:COUN
[SOUR]:LIST:CURR
[SOUR]:LIST:CURR:POIN?
[SOUR]:LIST:DWEL
[SOUR]:LIST:DWEL:POIN?
[SOUR]:LIST:FREQ[:LEV]
[SOUR]:LIST:FREQ[:LEV]:POIN?
[SOUR]:LIST:VOLT[:LEV]
[SOUR]:LIST:VOLT[:LEV]:POIN?
[SOUR]:PHAS[:IMM]
[SOUR]:PULS:COUN
[SOUR]:PULS:DCYC
[SOUR]:PULS:HOLD
[SOUR]:PULS:PER
[SOUR]:PULS:WIDT
[SOUR]:VOLT:ALC | SENS:SOUR
[SOUR]:VOLT[:LEV][:IMM][:AMPL]
[SOUR]:VOLT[:LEV][:TRIG][:AMPL]
[SOUR]:VOLT:MODE
[SOUR]:VOLT:PROT[:AMPL]
[SOUR]:VOLT:RANG
[SOUR]:VOLT:SLEW[:IMM]
STAT:OPER[:EVEN]?
STAT:OPER:COND?
STAT:OPER:ENAB
STAT:OPER:NTR
STAT:OPER:PTR
STAT:PRES
STAT:QUES[:EVEN]?
STAT:QUES:COND?
STAT:QUES:ENAB
STAT:QUES:NTR
STAT:QUES:PTR
STAT:QUES:INST:ISUM[:EVEN]?
STAT:QUES:INST:ISUM:COND?
STAT:QUES:INST:ISUM:ENAB
STAT:QUES:INST:ISUM:NTR
STAT:QUES:INST:ISUM:PTR
SYST:ERR?
SYST:LANG
SYST:VERS?
TRAC | DATA:CAT?
TRAC | DATA[:DATA]
TRAC | DATA:DEF
TRAC | DATA:DEL[:NAME]
TRIG[:SEQ1 | :TRAN][:IMM]
TRIG[:SEQ1 | :TRAN]:DEL
TRIG[:SEQ1 | :TRAN]:SOUR
TRIG:SEQ2 | SYNC:SOUR
TRIG:SEQ3 | ACQ[:IMM]
TRIG:SEQ3 | ACQ:SOUR
TRIG:SOUR

*CLS
*ESE *ESE? *ESR?
*IDN?
*OPC *OPC? *OPT?
*PSC *PSC?
*RCL *RST
*SAV *SRE *STB?
*TRG *TST?
*WAI

Non SCPI Commands

CAL:CURR:AC
CAL:CURR:DC
CAL:LEV
CAL:PASS
CAL:PWM:FREQ
CAL:PWM:RAMP
CAL:SAVE
CAL:VOLT:AC
CAL:VOLT:DC
CAL:VOLT:OFFS
DISP[:WIND]:MODE
MEAS | FETC[:SCAL]:CURR:ACDC?
MEAS | FETC[:SCAL]:CURR:AMPL:MAX?
MEAS | FETC[:SCAL]:CURR:CRES?
MEAS | FETC[:SCAL]:CURR:HARM[:AMPL]?
MEAS | FETC[:SCAL]:CURR:HARM:PHAS?
MEAS | FETC[:SCAL]:CURR:HARM:THD?
MEAS | FETC[:SCAL]:CURR:NEUT[:DC]?
MEAS | FETC[:SCAL]:CURR:NEUT:AC?
MEAS | FETC[:SCAL]:CURR:NEUT:DC?
MEAS | FETC[:SCAL]:CURR:NEUT:ACDC?
MEAS | FETC[:SCAL]:CURR:NEUT:HARM[:AMPL]?
MEAS | FETC[:SCAL]:CURR:NEUT:HARM:PHAS?
MEAS | FETC[:SCAL]:POW:AC:APP?
MEAS | FETC[:SCAL]:POW:AC:REAC?
MEAS | FETC[:SCAL]:POW:AC:PFAC?
MEAS | FETC[:SCAL]:POW:AC:TOT?
MEAS | FETC[:SCAL]:VOLT:ACDC?
MEAS | FETC[:SCAL]:VOLT:HARM[:AMPL]?
MEAS | FETC[:SCAL]:VOLT:HARM:PHAS?
MEAS | FETC[:SCAL]:VOLT:HARM:THD?
MEAS | FETC:ARR:CURR:HARM[:AMPL]?
MEAS | FETC:ARR:CURR:HARM:PHAS?
MEAS | FETC:ARR:CURR:NEUT[:DC]?
MEAS | FETC:ARR:CURR:NEUT:HARM[:AMPL]?
MEAS | FETC:ARR:CURR:NEUT:HARM:PHAS?
MEAS | FETC:ARR:VOLT:HARM[:AMPL]?
MEAS | FETC:ARR:VOLT:HARM:PHAS?
OUTP:DFI[:STAT]
OUTP:DFI:SOUR
OUTP:IMP:REAL
OUTP:IMP:REAC
OUTP:RI:MODE
OUTP:TTLT:SOUR
[SOUR]:CURR:PEAK[:IMM]
[SOUR]:CURR:PEAK:MODE
[SOUR]:CURR:PEAK:TRIG
[SOUR]:FREQ:SLEW[:IMM]
[SOUR]:FREQ:SLEW:MODE
[SOUR]:FREQ:SLEW:TRIG
[SOUR]:FREQ:TRIG
[SOUR]:FUNC[:SHAP]:MODE
[SOUR]:FUNC[:SHAP]:TRIG
[SOUR]:FUNC[:SHAP]:CSIN
[SOUR]:LIST:FREQ:SLEW
[SOUR]:LIST:FREQ:SLEW:POIN?
[SOUR]:LIST:PHAS
[SOUR]:LIST:PHAS:POIN?
[SOUR]:LIST:SHAP
[SOUR]:LIST:SHAP:POIN?
[SOUR]:LIST:STEP
[SOUR]:LIST:TTLT
[SOUR]:LIST:TTLT:POIN?
[SOUR]:LIST:VOLT:SLEW
[SOUR]:LIST:VOLT:SLEW:POIN?
[SOUR]:LIST:VOLT:OFFS
[SOUR]:LIST:VOLT:OFFS:POIN?
[SOUR]:PHAS:MODE
[SOUR]:PHAS:TRIG
[SOUR]:VOLT:ALC | SENS:DET
[SOUR]:VOLT:OFFS[:IMM]
[SOUR]:VOLT:OFFS:MODE
[SOUR]:VOLT:OFFS:TRIG
[SOUR]:VOLT:SLEW:MODE
[SOUR]:VOLT:SLEW:TRIG
SYST:CONF:NOUT
SYST:LOC
SYST:REM
SYST:RWL
TRIG:SEQ2 | SYNC:PHAS
TRIG:SEQ:DEF

C

Error Messages

Error Number List

This appendix gives the error numbers and descriptions that are returned by the ac source. Error numbers are returned in two ways:

- Error numbers are displayed on the front panel
- Error numbers and messages are read back with the SYSTem:ERRor? query. SYSTem:ERRor? returns the error number into a variable and returns two parameters, an NR1 and a string.

The following table lists the errors that are associated with SCPI syntax errors and interface problems. It also lists the device dependent errors. Information inside the brackets is not part of the standard error message, but is included for clarification. When errors occur, the Standard Event Status register records them in bit 2, 3, 4, or 5:

Table C-1. Error Numbers

Error #	Error String [Description/Explanation/Examples]
	Command Errors –100 through –199 (sets Standard Event Status Register bit #5)
–100	Command error [generic]
–101	Invalid character
–102	Syntax error [unrecognized command or data type]
–103	Invalid separator
–104	Data type error [e.g., "numeric or string expected, got block data"]
–105	GET not allowed
–108	Parameter not allowed [too many parameters]
–109	Missing parameter [too few parameters]
–112	Program mnemonic too long [maximum 12 characters]
–113	Undefined header [operation not allowed for this device]
–121	Invalid character in number [includes "9" in octal data, etc.]
–123	Numeric overflow [exponent too large; exponent magnitude >32 k]
–124	Too many digits [number too long; more than 255 digits received]
–128	Numeric data not allowed
–131	Invalid suffix [unrecognized units, or units not appropriate]
–138	Suffix not allowed
–141	Invalid character data [bad character, or unrecognized]
–144	Character data too long
–148	Character data not allowed
–150	String data error
–151	Invalid string data [e.g., END received before close quote]
–158	String data not allowed
–160	Block data error
–161	Invalid block data [e.g., END received before length satisfied]
–168	Block data not allowed

–170	Expression error
–171	Invalid expression
–178	Expression data not allowed
	Execution Errors –200 through –299 (sets Standard Event Status Register bit #4)
–200	Execution error [generic]
–221	Settings conflict [check current device state]
–222	Data out of range [e.g., too large for this device]
–223	Too much data [out of memory; block, string, or expression too long]
–224	Illegal parameter value [device-specific]
–225	Out of memory
–270	Macro error
–272	Macro execution error
–273	Illegal macro label
–276	Macro recursion error
–277	Macro redefinition not allowed
	System Errors –300 through –399 (sets Standard Event Status Register bit #3)
–310	System error [generic]
–350	Too many errors [errors beyond 9 lost due to queue overflow]
	Query Errors –400 through –499 (sets Standard Event Status Register bit #2)
–400	Query error [generic]
–410	Query INTERRUPTED [query followed by DAB or GET before response complete]
–420	Query UNTERMINATED [addressed to talk, incomplete programming message received]
–430	Query DEADLOCKED [too many queries in command string]
–440	Query UNTERMINATED [after indefinite response]
	Selftest Errors 0 through 99 (sets Standard Event Status Register bit #3)
0	No error
1	Non-volatile RAM RD0 section checksum failed
2	Non-volatile RAM CONFIG section checksum failed
3	Non-volatile RAM CAL section checksum failed
4	Non-volatile RAM WAVEFORM section checksum failed
5	Non-volatile RAM STATE section checksum failed
6	Non-volatile RAM LIST section checksum failed
7	Non-volatile RAM RST section checksum failed
10	RAM selftest
11 - 31	DAC selftest error, expected <n>, read <reading> Errors 11, 12, 13, 14, 15 apply to DAC12 1A and 1B Errors 16, 17, 18 apply to DAC12 2A Errors 19, 20, 21 apply to DAC12 2B Errors 22, 23 apply to DAC12 4A Errors 24, 25 apply to DAC12 4B Errors 26, 27, 28 apply to DAC12 3A and 3B Errors 29, 30, 31 apply to DAC12 5A and 5B
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
70	Fan voltage failure
80	Digital I/O selftest error
	Device-Dependent Errors 100 through 32767 (sets Standard Event Status Register bit #3)
200	Outgrd not responding
201	Front panel not responding
210	Ingrd receiver framing error
211	Ingrd uart overrun status
212	Ingrd received bad token
213	Ingrd receiver buffer overrun
214	Ingrd input buffer overrun
215	Outgrd output buffer overrun
216	RS-232 receiver framing error
217	RS-232 receiver parity error
218	RS-232 receiver overrun error
219	Ingrd inbuf count sync error
220	Front panel uart overrun
221	Front panel uart framing
222	Front panel uart parity
223	Front panel buffer overrun
224	Front panel timeout
401	CAL switch prevents calibration
402	CAL password is incorrect
403	CAL not enabled
404	Computed 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:SHAP modes incompatible
608	Measurement overrange
609	Output buffer overrun
610	Command cannot be given with present SYST:CONF setting

D

Elgar Model 9012 Compatibility

Elgar Model 9012 Plug-in Programmer Compatibility

The ac source interface has a language switch command that allows it to emulate the Elgar Model 9012 Plug-in Programmer. The command to switch languages is:

```
SYSTem:LANGuage SCPI | E9012
```

The language selection is stored in non-volatile memory and is retained after power-off. Regardless of which language is selected, the current selection can be queried using

```
SYSTem:LANGuage?
```

When E9012 is selected, all SCPI commands are disabled and the interface responds only to Elgar commands. Many ac source features which do not have corresponding Elgar commands are disabled in this mode, as are their front panel controls. The basic capabilities of the Elgar PIP are control of voltage, frequency, and current limit, and readback of frequency, rms voltage, current, and power. Transient capabilities include dropout and some phase synchronization of the output voltage changes.

Main Board W1 Jumper Option Emulation

Some Elgar 9012 programming commands are affected by the installation of an "alternate" W1 jumper configuration. This behavior can be programmed in E9012 language using the following command:

```
JUMPer1 NORMal | ALTernate
```

The JUMPer1 setting is stored in non-volatile memory and is retained after power off. The jumper affects the phase synchronization of the PEAK, ZERO, and DROP commands.

Syntax Compatibility

The instrument accepts a superset of Elgar of the E9012 PIP commands. It should correctly accept any command string that the Elgar PIP accepts. However, it is less strict in checking white space between headers and arguments, and it accepts more combinations of commands within a single message unit. Any number of Elgar commands can be combined into a single program message unit. The command separator is a comma, and serves the same purpose as the semicolon in SCPI. The format of query responses is identical to that of the E9012 PIP. The response terminator is a newline with EOI asserted.

Status Model

In E9012 language, status information is provided through the serial poll response byte and the error queue. The error queue operates as it does in SCPI language, providing error status of selftest and other runtime errors. The SYSTem:ERRor? query operates identically in SCPI and E9012 languages.

The E9012 language provides an abbreviated status model consisting entirely of the serial poll response byte. Other SCPI status registers (questionable, operation, and standard event) are not defined for this language. The serial poll responses are limited to the following:

Byte	Serial Poll Response
64	SOA, Overtemperature, or Rail fault has tripped
67	Overvoltage protection has tripped
71	Overcurrent protection has tripped on phase 1
72	Overcurrent protection has tripped on phase 2
73	Overcurrent protection has tripped on phase 3
74	Syntax error
75	Command error (value out of range, improper mode, etc)
76	Query interrupted, query unterminated, or deadlocked error
78	Measurement error
79	Measurement complete

Reading the serial poll response byte clears it to 0. There is no queue of responses, so the value read will indicate the most recent event.

Power-on State

The ac power source is set to the E9012 power-on state when any of the following occur.

- The power source is turned on with the E9012 language selected.
- The SYST:LANG E9012 command is given and the language had been set to SCPI.
- A GPIB Device Clear or Selected Device Clear command is sent to the power source while the E9012 language was selected.

The power-on state in E9012 language is equivalent to giving the following commands:

Command	Description
VOLTS 0	0V output voltage
CURL 0	over-current protection disabled, current limit set to MAX
FREQ 400	400Hz output frequency
RNG 0	low voltage range
RNGF 2	frequency range to 1200Hz
CLS	output relay closed

All power source functions not set by the above commands go to the state defined by the SCPI *RST command, with the following exceptions:

```
OUTPut:STATe ON
VOLT:SENSe:SOURce EXTernal
VOLT:SENSe:DETector RMS
```

Protection

The SOA fault, overtemperature, rail fault, overvoltage, and overcurrent protection features are operational in E9012 language and are reported through the serial poll response byte. Clearing the protection latch is done by programming an output voltage using the VOLTS command.

There is no command to set the overvoltage threshold in E9012 language. The value is fixed at MAXimum (500V). The overcurrent protection can be enabled by programming a non-zero value for the current limit. Setting the value to 0 disables overcurrent protection and sets the current limit function to allow maximum load current.

Front Panel Operation

Many front panel keys are re-defined when E9012 language is selected.

System Keys

The Local, Address, and Error keys are identical in SCPI and E9012 languages. The Recall and Save keys are not operational in E9012 language.

Function Keys

The Meter, Harmonic, Index, Phase Select, and Output on/off keys are identical in SCPI and E9012 languages. The Output, Phase, Protect, Status, Shape, Trigger and List keys are not operational in E9012 language.

Voltage key functions:

Display Format	**Description**
VOLTS <value>	Set AC output voltage
RNG <value>	Set voltage range (0 | 1)

Current key functions:

Display Format	**Description**
CURL <value>	Set current limit. A value of 0 turns off over-current protection

Freq key functions:

Display Format	**Description**
FREQ <value>	Set output frequency
RNGF <value>	Set frequency range (0, 1, or 2)

Input key functions:

Display Format	Description
INP:COUP <enum>	Set coupling for front panel measurements (AC | DC | ACDC)

Trigger Control key functions:

Display Format	Description
JUMPER1 <enum>	Set W1 emulation jumper (NORM | ALT)

Pulse key functions:

Display Format	Description
DROP <value>	Dropout for <value> half cycles (JUMPER1 NORM) or for <value> full cycles (JUMPER1 ALT)

Entry Keys

The Calibration key is not operational in E9012 language. All other Entry keys are identical in SCPI and E9012 languages.

E9012 Language Command Summary

Command	Description
`<Device Clear>`	Set the instrument to the power-on state and clear the error queue and the serial poll response byte.
`VOLTS <n>`	Set the rms output voltage.
`CURL <n>`	Set the rms output current limit. A value of 0 disables over-current protection and sets the current limit function to allow maximum load current.
`FREQ <n>`	Sets the output frequency.
`RNG 0 | 1`	Set the output voltage range. 0 = low range 1 = high range If the voltage range is changing, the output voltage is set to 0. If the voltage range is not changing, the output voltage is unaffected.
`RNGF 0 | 1 | 2`	Set output frequency range. 0 = to 99.99Hz 1 = to 999.9Hz 2 = to 1200Hz If the frequency range is changing, the frequency is set to 60Hz for range 0, and to 400Hz for ranges 1 and 2. If the frequency range is not changing, the output frequency is unaffected.
`CLS`	Closes the output relay
`OPN`	Opens the output relay
`LOCK`	Disables fault shutdown
`UNLK`	Enables fault shutdown

ZERO	The next voltage or frequency change is at 0 degrees phase (JUMPer1 NORMal), or 180 degrees phase (JUMPer1 ALTernate). The VOLTS or FREQ command must be part of the same program message unit.
PEAK	The next voltage or frequency change is at 90 degrees phase (JUMPer1 NORMal), or 270 degrees phase (JUMPer1 ALTernate). The VOLTS or FREQ command must be part of the same program message unit.
OVER	No action (relaxes voltage and frequency limits in Elgar PIP)
DROP <n>	Sets dropout for <n> half cycles (JUMPer1 NORMal), or <n> full cycles (JUMPer1 ALTernate).
OFF	Sets output voltage to 0 volts at 0 degrees phase sync.
ON 1 | 2	Restores output voltage to previous setting before OFF was given. 0 = restores immediately, with arbitrary phase 1 = restores at 0 degrees phase sync 2 = restores at 90 degrees phase sync
JUMPer1 NORMal | ALTernate	Sets emulation of Elgar main board W1 jumper option. This setting is saved in non-volatile memory and is retained after power off. This command is not available in the E9012 PIP.
SYSTem:LANGuage SCPI | E9012	Sets the programming language. The language setting is saved in non-volatile memory and is retained after power off. This command is not available in the E9012 PIP.
SYSTem:LANGuage?	Returns the selected programming language. This query is not available in the E9012 PIP.
SYSTem:ERRor?	Returns an error number and error string. This query is not available in the E9012 PIP.
TEST 0	Returns the output frequency, format: F=nnnn.H
TEST 1 | 2 | 3	Returns the rms output voltage, phase 1, 2, or 3, format: A=nnn.nV
TEST 4 | 5 | 6	Returns the rms output current, phase 1, 2, or 3, format: A=nn.nnA
TEST 7 | 8 | 9	Returns the output power, phase 1, 2, or 3, format: A=nnnn.W

E

IEC Mode Command Summary

Introduction

The Keysight 6812B, 6813B, and 6843A ac sources are designed to operate in Normal as well as IEC mode. In Normal mode, the units respond to all of the commands that program ac source operation. Normal mode commands are documented in this Programmer's Guide. In IEC mode, when used in conjunction with the Keysight 14761A Harmonic and Flicker Emissions Tests software, ac source provides full EN 61000-3-2/EN 60555 Part 2 and EN 61000-3-3 compliance test capability. The SYSTem CONFigure command details the differences between Normal and IEC mode.

When a Keysight 6812B, 6813B, or 6843A ac source is being used in IEC mode, the Keysight 14761A Harmonic and Flicker Emissions Tests software handles all of the communication between the user and the ac source. The Keysight 14761A software must be loaded and running in Microsoft Windows on a personal computer that is connected to the ac source.

The IEC commands that are described in this appendix are for those users who need to directly program the IEC functions of the Keysight 14761A without using the Keysight 14761A software. Be aware that these commands will return "raw" data from the ac source. It is the programmer's responsibility to interpret the data according to the IEC standards.

Using the SENSe:CURRent:ACDC:RANGe command

The SENSe:CURRent:ACDC:RANGe command is documented earlier in this Programmer's Guide. When using this command in IEC mode, you must always initialize it before making any Array measurements by sending a Meas:Curr? command. For example:

```
SENSe:CURRent:ACDC:RANGe
MEASure:CURRent?
ENTER
```

Command Syntax

CALCulate	
:INTegral	
:TIME <Nrf+>	selects the Pst integration time for flicker measurements
:LIMit	
:UPPer	sets various limits associated with rms voltage
[:DATA] <Nrf+>	fluctuation testing for IEC 1000-3-3
:SMOothing <bool>	turns the 1.5 second smoothing filter on or off
FORMat	
[:DATA] <type>	specifies the response data format (ASCii | REAL)
:BORDer <type>	sets the byte order of the floating point values returned (NORMal | SWAPped)
MEASure	
:ARRay	
:CURRent	
:HARMonic? <NRf+>	returns an array of current harmonic magnitudes
:VOLTage	
:FLUCtuations	
:FLICker? <NRf+>	returns rms and instantaneous flicker values
:PST? <NRf+>	returns Pst summary values
:ALL? <NRf+>	returns both rms/flicker and Pst summary values
SENSe	
:CURRent	sets the phase reference for current harmonic phase
:PREFerence <type>	measurements (VOLTage | CURRent)
:WINDow	
[:TYPE] <type>	selects the window function used in the harmonic measurements (HANNing | KBESsel | RECTangular)
SYSTem	
:CONFigure <mode>	selects the operating mode of the ac source (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 IEC 868. The command will be accepted and may be queried, but will have no meaningful function unless the ac source is placed in IEC 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: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 IEC 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?

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[1] that defines "steady-state". At *RST this value is set to 0.003[2]. Note that this number is not specified by IEC 1000-3-3.
dmax limit	Sets the maximum relative voltage[1] change allowed before a dmax error is flagged. At *RST this value is set to 0.04 (see note [2]).
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 [2]).
dt tlimit	Sets the maximum time in seconds that the relative voltage[1] 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[1] 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 [2]).

[1]The expression "relative voltage" as used above is the measured rms voltage divided by the programmed voltage.

[2]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>

FORMat

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

MEASure:ARRay:CURRent:DC?
MEASure:ARRay:VOLTage:DC?
MEASure:ARRay:CURRent:HARMonic[:AMPLitude]?
MEASure:ARRay:VOLTage:FLUCutations:ALL?
MEASure:ARRay:VOLTage:FLUCutations:FLICker?
MEASure:ARRay:VOLTage:FLUCutations: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:FLICker? MEAS:ARR:VOLT:FLUC:PST?

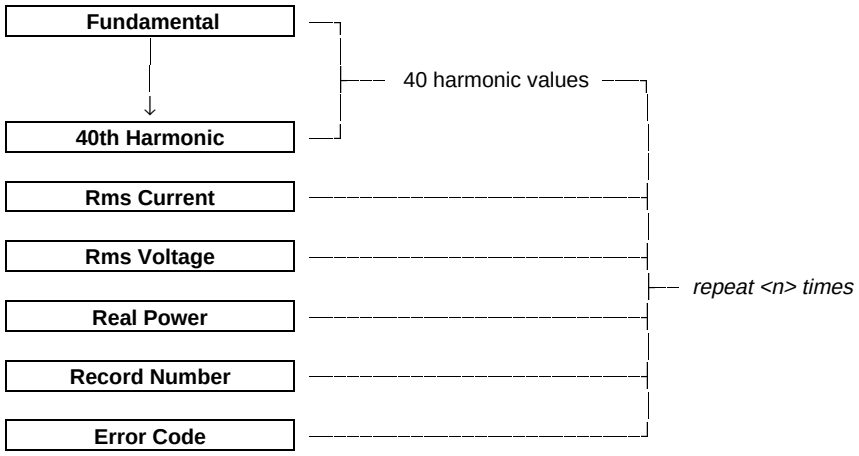
MEASure:ARRay:CURRent:HARMonic?

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 (ie 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 impacts availability of the RMS Current, RMS Voltage, 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 60555 Part 2 and related regulations). 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 one. 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?

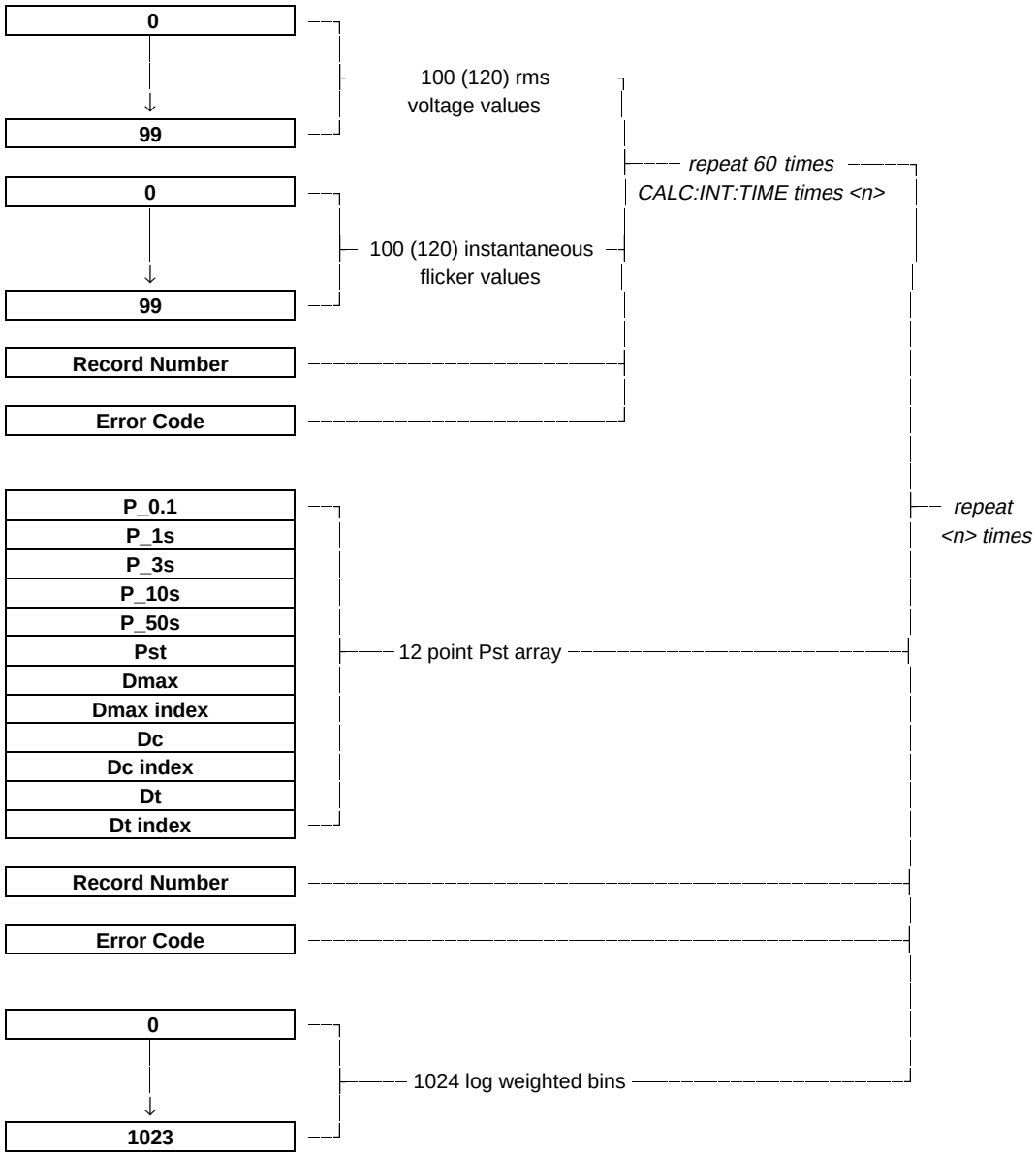
This query measures voltage fluctuations in accordance with the IEC 868 standard. It is only available when IEC mode is selected with SYSTem:CONFigure. The parameter specifies the number of Pst integration periods during which data will be returned in response to the query.

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 itself, various RMS voltage values (Dmax, Dc, and Dt), together with indices for these RMS values that give their approximate 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 set of logarithmically weighted ranges of instantaneous flicker is returned for each Pst integration period. The array covers a flicker perceptibility range of 0.01 to 10000 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 (ie 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 50Hz 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:



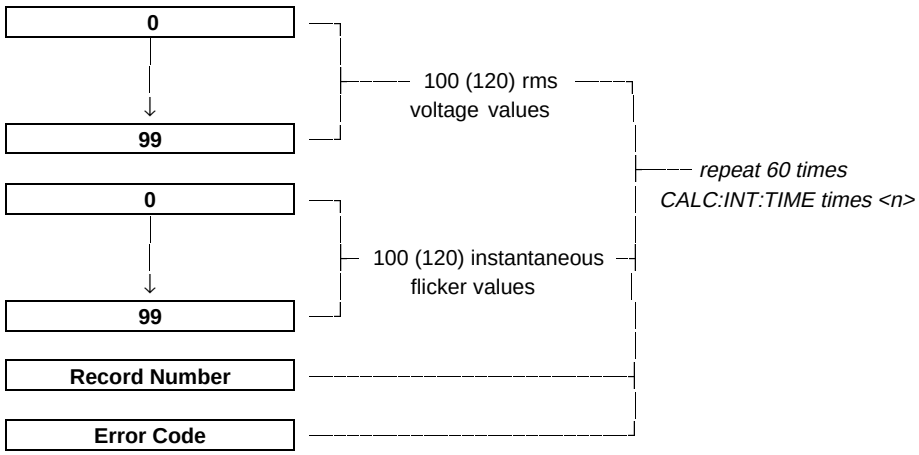
Command Syntax MEASure:ARRay:VOLTage:FLUCtuations:ALL? <NRf+>
Parameters 1 to 1008
Examples MEAS:ARR:VOLT:FLUC:ALL? 12
Returned Parameters 13,158 to 220,588,704 values
Related Commands ABORt SYSTEM:CONF INST:NSEL
MEAS:ARR:VOLT:FLUC:PST?
MEAS:ARR:VOLT:FLUC:ALL?

MEASure:ARRay:VOLTage:FLUCtuations:FLICker?

This query measures voltage fluctuations in accordance with the IEC 868 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 contained 202 (50Hz) or 242 (60Hz) 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 50Hz 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 one. The figure below defines the structure of the data returned by this query:



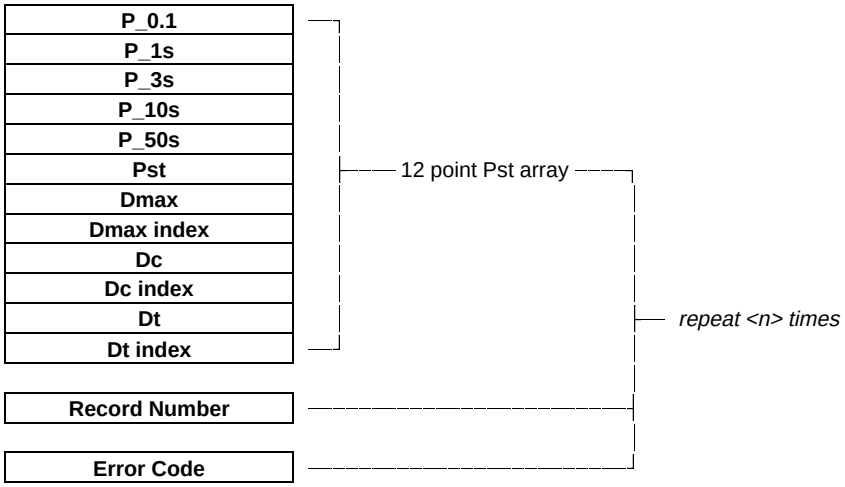
Command Syntax MEASure:ARRay:VOLTage:FLUCtuations:FLICker? <NRf+>
Parameters 1 to 1008
Examples MEASure:ARRay:VOLTage:FLUCtuations:FLICker? 12
Returned Parameters 12120 to 219,542,400 values
Related Commands ABORt SYSTEM:CONF INST:NSEL
MEAS:ARR:VOLT:FLUC:PST?
MEAS:ARR:VOLT:FLUC:ALL?

MEASure:ARRay:VOLTage:FLUCtuations:PST?

This query measures voltage fluctuations in accordance with the IEC 868 standard. It is only available when IEC mode is selected with SYSTem:CONFigure. The parameter specifies the number of Pst integration periods for 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 itself, various RMS voltage values (Dmax, Dc, and Dt), together with indices for these RMS values that give their approximate 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 one. 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	ABORt SYSTEM: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

SENSe:WINDow

This command sets the window function which is used in harmonic measurements. The choice of parameters is affected by the SYSTem:CONF command. If NORMal is selected, HANNing, KBESsel, or RECTangular may be selected. IF IEC mode is selected, only HANNing and RECTangular may be selected. KBESsel is the preferred window and should be used for most measurements in NORMal mode. HANNing and RECTangular are available for making harmonic current measurements that comply with the regulatory requirements.

Command Syntax	SENSe:WINDow [:TYPE] <window>
Parameters	HANNing | KBESsel | RECTangular
***RST Value**	KBESsel (RECTangular is the default setting when the Keysight 14761A software is run)
Examples	SENS:WIND RECT
Query Syntax	SENSe:WINDow?
Returned Parameters	<CRD>
Related Commands	MEASure:ARRay:CURRent:HARMonic? SYSTem:CONF MEASure:CURRent:HARMonic? MEASure:ARRay:VOLTage:HARMonic? MEASure:VOLTage:HARMonic?

SYSTem:CONFigure

Keysight **6812B, 6813B, 6843A Only**

This command sets the overall operating mode of the ac source. The choices are Normal mode, which causes the product to closely mimic the operating characteristics of standard ac sources, 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:

	NORMAL MODE	**IEC MODE**
Base Sampling Rate	39.920792 KHz	38.400000 KHz
Output Frequency	DC - 1000 Hz	50 Hz & 60 Hz Only
Freq/Window/Fs Mode	Independent	Coupled
Transient System	FIXEd/STEP/PULSe/LIST Modes	FIXEd Mode Only
Slew Operation	AC & DC Voltage; Frequency	AC Voltage Only
MEASure:ARRay:CURRent :HARM?	DC, Fundamental, and Harmonics to 50th	Fundamental and Harmonics to 40th plus RMS Current, RMS Voltage & Real Power
CALCulate:SMOothing	Not/Available	1.5 Second Smoothing on/off
MEASure:ARRay:VOLTage :FLUCtuations	Not/Available	FLICker | PST | ALL?
CALCulate:INTegral:TIME	Not/Available	1 | 5 | 10 | 15 MINutes

Transmission of a SYSTem:CONFigure command implies ABORt and terminates any transient or measurement actions previously initiated.

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

Index

—A—

ac voltage, 122
assigning HP-IB address in programs, 24

—B—

BNC connectors, 148

—C—

calibration subsystem, 29
  CALibrate CURRent AC, 29
  CALibrate CURRent MEASure, 30
  CALibrate DATA, 30
  CALibrate IMPedance, 30
  CALibrate LEVel, 30
  CALibrate PASSword, 31
  CALibrate PWM FREQuency, 31
  CALibrate PWM RAMP, 31
  CALibrate SAVE, 31
  CALibrate STATe, 32
  CALibrate VOLTage AC, 32
  CALibrate VOLTage DC, 32
  CALibrate VOLTage OFFSet, 33
  CALibrate VOLTage PROTection, 33
  CALibrate VOLTage RTIMe, 33
  syntax, 29
character data, 23
Combing commands
  root specifier, 19
Combining commands
  from different subsystems, 19
  root specifier, 20
command completion, 150
command summary
  format, 168, 171
    border, 171
  system
    configure, 169, 171
Command tree
  active header path, 18
commands
  coupled, 20
  EN 61000-3-3, 167
common commands, 113
  *CLE, 114
  *ESE, 114
  *ESR?, 115
  *IDN?, 115
  *OPC, 115
  *OPT?, 116
  *PSC, 116
  *RCL, 116
  *RST, 117
  *SAV, 118
  *SRE, 118
  *STB?, 119
  *TRG, 119
  *TST?, 119
  *WAI, 120
  syntax, 113
Conventions used in this manual, 17
coupled commands, 20, 127
current data
  instantaneous, 139
current limit, 125
  peak, 125
current measurements, 137
current sampling rate
  varying, 141

—D—

dc output, 126
delay, 135
DFI indicator, 149
discrete fault, 149
display subsystem, 34
  DISPlay, 34
  DISPlay MODE, 34
  DISPlay TEXT, 34
  syntax, 34
DOS driver types, 24
dwell
  specifying, 136

—E—

enabling the output, 121
error handling, 25
error numbers, 157
errors
  program execution, 25
event triggers, 141

—F—

FETCh, 137
flicker, 168, 174, 175, 176
FLT connection, 149
frequency, 122

—H—

harmonic measurements, 138, 139
headers, 21
  convention, 21
  long form, 21
  optional, 21
  separator, 21

short form, 21
history, 2
HP 82335A driver, 24
HP BASIC controllers, 25
HP-IB
capabilities of ac source, 15
command library for MS DOS, 12
controller programming, 12
IEEE Std for standard codes, 12
IEEE Std for standard digital interface, 12
HP-IB Address, 15
HP-IB References, 12
HP-IB source address, 24

—I—

INH
connection, 149
instrument subsystem, 35
INSTrument COUPle, 35
INSTrument NSELect, 36
INSTrument SELect, 36
syntax, 35

—L—

language dictionary, 27
list transients, 131

—M—

MAV bit, 147
maximum coupled voltage, 126
maximum voltage, 122
MEASure, 137
measurement
triggers, 139
measurement subsystem
FETCh ARRay CURRent HARMonic PHASe?, 38
FETCh ARRay CURRent HARMonic?, 38
FETCh ARRay CURRent NEUTral HARMonic PHASe?, 40
FETCh ARRay CURRent NEUTral HARMonic?, 39
FETCh ARRay CURRent NEUTral?, 39
FETCh ARRay CURRent?, 37
FETCh ARRay VOLTage HARMonic PHASe?, 41
FETCh ARRay VOLTage HARMonic?, 41
FETCh ARRay VOLTage?, 40
FETCh CURRent AC?, 43
FETCh CURRent ACDC?, 43
FETCh CURRent AMPLitude MAXimum?, 43
FETCh CURRent CREStfactor?, 44
FETCh CURRent HARMonic PHASe?, 45
FETCh CURRent HARMonic THD?, 45
FETCh CURRent HARMonic?, 44
FETCh CURRent NEUTral AC?, 46
FETCh CURRent NEUTral ACDC?, 46
FETCh CURRent NEUTral HARMonic PHASe?, 47
FETCh CURRent NEUTral HARMonic?, 46
FETCh CURRent NEUTral?, 45
FETCh CURRent?, 42
FETCh POWer AC APParent?, 50
FETCh POWer AC PFACtor?, 50
FETCh POWer AC REACtive?, 50
FETCh POWer AC TOTal?, 51
FETCh POWer AC?, 49
FETCh POWer?, 49
FETCh VOLTage AC?, 52
FETCh VOLTage ACDC?, 53
FETCh VOLTage HARMonic PHASe?, 54
FETCh VOLTage HARMonic THD?, 54
FETCh VOLTage HARMonic?, 53
FETCh VOLTage?, 52
MEASure ARRay CURRent HARMonic PHASe?, 38
MEASure ARRay CURRent HARMonic?, 38
MEASure ARRay CURRent NEUTral HARMonic PHASe?, 40
MEASure ARRay CURRent NEUTral HARMonic?, 39
MEASure ARRay CURRent NEUTral?, 39
MEASure ARRay CURRent?, 37
MEASure ARRay VOLTage HARMonic PHASe?, 41
MEASure ARRay VOLTage HARMonic?, 41
MEASure ARRay VOLTage?, 40
MEASure CURRent AC?, 43
MEASure CURRent ACDC?, 43
MEASure CURRent AMPLitude MAXimum?, 43
MEASure CURRent CREStfactor?, 44
MEASure CURRent HARMonic PHASe?, 45
MEASure CURRent HARMonic THD?, 45
MEASure CURRent HARMonic?, 44
MEASure CURRent NEUTral AC?, 46
MEASure CURRent NEUTral ACDC?, 46
MEASure CURRent NEUTral HARMonic PHASe?, 47
MEASure CURRent NEUTral HARMonic?, 46
MEASure CURRent NEUTral?, 45
MEASure CURRent?, 42
MEASure POWer AC APParent?, 50
MEASure POWer AC PFACtor?, 50
MEASure POWer AC REACtive?, 50
MEASure POWer AC TOTal?, 51
MEASure POWer AC?, 49
MEASure POWer?, 49
MEASure VOLTage AC?, 52
MEASure VOLTage ACDC?, 53
MEASure VOLTage HARMonic PHASe?, 54
MEASure VOLTage HARMonic THD?, 54
MEASure VOLTage HARMonic?, 53
MEASure VOLTage?, 52
measurement subsystem, arrays, 37
syntax, 37
measurement subsystem, current, 42
syntax, 42
measurement subsystem, FETCh FREQuency?, 48
measurement subsystem, frequency, 48
syntax, 48
measurement subsystem, MEASure FREQuency?, 48
measurement subsystem, power, 49
syntax, 49
measurement subsystem, voltage, 52
syntax, 52
measurement system, 137

measurement trigger system
  initiating, 140
  model, 139
measurement triggers
  generating, 141
  selecting, 140
message terminator, 22
  end or identify, 22
  newline, 22
message unit separator, 22
Message Units
  combing message units, 21
Moving among subsystems, 19
MSS bit, 147
multipliers, 23

—N—

National Instruments GP-IB driver, 24
NTR filter, 148
numerical data formats, 23

—O—

offset, 126
operation status group, 142
Optional header
  LEVel, 19
Optional Headers
  effect of, 19
output
  triggers, 132
output phase measurements
  simultaneous, 138
output phases, 124
  programming, 125
  selecting, 125
output queue, 147
output subsystem, 55
  OUTPut, 55
  OUTPut COUPling, 56
  OUTPut DFI, 56
  OUTPut DFI SOURce, 56
  OUTPut IMPedance, 57
  OUTPut IMPedance REACtive, 57
  OUTPut IMPedance REAL, 57
  OUTPut PON STATe, 58
  OUTPut PROTection CLEar, 58
  OUTPut PROTection DELay, 58
  OUTPut RI MODE, 59
  OUTPut TTLTrg, 59
  OUTPut TTLTrg SOURce, 59
  syntax, 55
output transients, 128
  model, 129
  synchronizing, 135
output trigger system
  initiating, 134
  model, 133
output triggers
  generating, 136
  selecting, 134

—P—

peak current limit, 125
post-event triggers, 141
power measurements, 138
power-on bit, 146
power-on initialization, 121
power-on status conditions, 142
pre-event triggers, 141
print date, 2
program commands
  measure array current harmonic?, 173
  measure array voltage fluctuations all?, 174
  measure array voltage fluctuations flicker?, 176
  measure array voltage fluctuations pst?, 177
  sense current preference, 178
  sense window, 178
  system configure, 179
programming commands
  calculate integral time, 169
  calculate limit upper, 170
  calculate smoothing, 169
  format, 171
  format border, 172
programming the output, 121
pst integration time, 168, 169
PTR filter, 148
pulse transients, 130

—Q—

quasi-stationary harmonics, 139
queries, 20
query indicator, 22
questionable instrument status group, 145
questionable instrument summary registers
  bit configuration, 99
questionable status condition registers
  bit configuration, 97
questionable status events, 148
questionable status group, 145

—R—

remote inhibit, 149
RI indicator, 149
root level, 18
Root specifier, 22
  combing commands, 19
RQS bit, 147
RS-232
  capabilities, 15
  data format, 15
  program example, 16
  troubleshooting, 16

—S—

safety summary, 2
sampling rate, 141
SCPI
  command syntax, 27
  data formats, 23
  subsystem commands, 28
  triggering nomenclature, 132, 139
SCPI command completion, 150
SCPI command tree, 18
SCPI common commands, 17
SCPI message structure, 20
SCPI message types, 17
SCPI message unit, 20
SCPI parser, 19
SCPI program message, 17
SCPI References, 12
SCPI response message, 17
SCPI subsystem commands, 17
sense subsystem, 60
  SENSe CURRent
    ACDC
      RANGe, 60
  SENSe SWEep
    OFFSet
      POINts, 61
    TINTerval, 61
  SENSe WINDow, 61
  syntax, 60
service interrupt, 147
service request
  generating, 148
slew rate
  frequency, 123
  voltage, 123
source subsystem
  CURRent, 62
  CURRent PEAK, 63
  CURRent PEAK MODE, 63
  CURRent PEAK TRIGgered, 64
  CURRent PROTection STATe, 64
  FREQuency, 65
  FREQuency MODE, 65
  FREQuency SLEW, 66
  FREQuency SLEW MODE, 66
  FREQuency SLEW TRIGgered, 66
  FREQuency TRIGgered, 67
  FUNCtion, 68
  FUNCtion CSINusoid, 70
  FUNCtion MODE, 69
  FUNCtion TRIGgered, 69
  LIST COUNt, 72
  LIST CURRent, 72
  LIST CURRent POINts?, 72
  LIST DWELl, 73
  LIST DWELl POINts?, 73
  LIST FREQuency, 73
  LIST FREQuency POINts?, 74
  LIST FREQuency SLEW, 74
  LIST FREQuency SLEW POINts?, 74
  LIST PHASe, 74
  LIST PHASe POINts?, 75
  LIST SHAPe, 75
  LIST SHAPe POINts?, 75
  LIST STEP, 76
  LIST TTLTrg, 76
  LIST TTLTrg POINts?, 76
  LIST VOLTage, 77
  LIST VOLTage OFFSet, 78
  LIST VOLTage OFFSet POINts?, 78
  LIST VOLTage POINts?, 77
  LIST VOLTage SLEW, 77, 79
  LIST VOLTage SLEW POINts?, 78, 79
  PHASe, 80
  PHASe MODE, 81
  PHASe TRIGgered, 81
  PULSe COUNt, 82
  PULSe DCYCle, 82
  PULSe HOLD, 83
  PULSe PERiod, 84
  PULSe WIDTh, 84
  VOLTage, 86
  VOLTage ALC DETector, 91
  VOLTage ALC SOURce, 92
  VOLTage MODE, 87
  VOLTage OFFSet, 87
  VOLTage OFFSet MODE, 88
  VOLTage OFFSet SLEW, 89
  VOLTage OFFSet SLEW MODE, 89
  VOLTage OFFSet SLEW TRIGgered, 90
  VOLTage OFFSet TRIGgered, 88
  VOLTage PROTection, 90
  VOLTage PROTection STATe, 90
  VOLTage RANGe, 91
  VOLTage SENSe DETector, 91
  VOLTage SENSe SOURce, 92
  VOLTage SLEW, 92
  VOLTage SLEW MODE, 93
  VOLTage SLEW TRIGgered, 93
  VOLTage TRIGgered, 86
source subsystem, current, 62
  syntax, 62
source subsystem, frequency, 65
  syntax, 65
source subsystem, function, 68
  syntax, 68
source subsystem, list, 71
  syntax, 71
source subsystem, phase, 80
source subsystem, pulse, 82
  syntax, 82
source subsystem, voltage, 85
  syntax, 85
SRQ
  determining cause of, 147
standard event status enable register
  bit configuration, 114
standard event status group, 146
status byte register, 147
  bit configuration, 119
status operation registers

bit configuration, 95
status registers
programming, 142
status subsystem, 94
QUEStionable INSTrument ISUMmary CONDition?, 100
QUEStionable INSTrument ISUMmary ENABle?, 100
QUEStionable INSTrument ISUMmary?, 99
STATus OPERation CONDition?, 95
STATus OPERation ENABle?, 95
STATus OPERation NTRansition, 96
STATus OPERation PTRansition, 96
STATus OPERation?, 95
STATus PRESet, 94
STATus QUEStionable CONDition?, 97, 98
STATus QUEStionable INSTrument ISUMmary NTRansition, 101
STATus QUEStionable INSTrument ISUMmary PTRansition, 101
STATus QUEStionable NTRansition, 98
STATus QUEStionable PTRansition, 98
STATus QUEStionable?, 97
syntax, 94
status transition, monitoring, 148
step transients, 130
suffixes, 23
system commands, 102
syntax, 102
SYSTem CONFgure, 102
SYSTem CONFigure NOUTputs, 103
SYSTem ERRor?, 103
SYSTem LANGuage, 104
SYSTem LOCal, 104
SYSTem REMote, 104
SYSTem RWLock, 104
SYSTem VERSion?, 103
system considerations, 24
system errors, 157

—T—

trace subsystem, 105
DATA, 105
DATA CATalog?, 106
DATA DEFine, 106
DATA DELete, 106
syntax, 105
TRACe, 105
TRACe CATalog?, 106
TRACe DEFine, 106
TRACe DELete, 106
transients
programming, 128, 130, 131
synchronizing, 135
trigger delay
selecting, 135
Trigger In, programming, 148
Trigger Out, programming, 148
trigger subsystem, 107
ABORt, 108
INITiate CONTinuous NAME, 109
INITiate CONTinuous SEQuence, 109
INITiate NAME, 108
INITiate SEQuence, 108
syntax, 107
TRIGger, 109
TRIGger ACQuire, 111
TRIGger ACQuire SOURce, 112
TRIGger DELay, 109
TRIGger SEQuence1 DEFine, 112
TRIGger SEQuence2 DEFine, 112
TRIGger SEQuence2 PHASe, 111
TRIGger SEQuence2 SOURce, 110
TRIGger SEQuence3, 111
TRIGger SEQuence3 DEFine, 112
TRIGger SEQuence3 SOURce, 112
TRIGger SOURce, 110
TRIGger SYNChronize PHASe, 111
TRIGger SYNChronize SOURce, 110
triggers
event, 141
generating, 136, 141
measurement, 139
multiple, 136
output, 132
programming, 133, 134, 139, 140
selecting, 134, 140
single, 136
Types of SCPI commands, 17

—U—

using queries, 20

—V—

voltage, 122
voltage data
instantaneous, 139
voltage measurements, 137
voltage sampling rate
varying, 141

—W—

waveform
clipped, 123
shapes, 123
square, 123
user defined, 123

Keysight Sales and Support Offices

For more information about Keysight Technologies test and measurement products, applications, services, and for a current sales office listing, visit our web site: http://www.keysight.com/find/tmdir

You can also contact one of the following centers and ask for a test and measurement sales representative.

United States:
Keysight Technologies
Test and Measurement Call Center
P.O. Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

Canada:
Keysight Technologies Canada
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

Europe:
Keysight Technologies
Test & Measurement European Marketing Organisation
P.O. Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

Japan:
Keysight Technologies Japan Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840
Technical data is subject to change.

Latin America:
Keysight Technologies
Latin American Region Headquarters
5200 Blue Lagoon Drive, Suite #950
Miami, Florida 33126
U.S.A.
(tel) (305) 267 4245
(fax) (305) 267 4286

Australia/New Zealand:
Keysight Technologies Australia Pty Ltd
347 Burwood Highway
Forest Hill, Victoria 3131
(tel) 1-800 629 485 (Australia)
(fax) (61 3) 9272 0749
(tel) 0 800 738 378 (New Zealand)
(fax) (64 4) 802 6881

Asia Pacific:
Keysight Technologies
24/F, Cityplaza One, 1111 King's Road,
Taikoo Shing, Hong Kong
tel: (852)-3197-7777
fax: (852)-2506-9284

This information is subject to change without notice.
© Keysight Technologies 1996 - 2014
Edition 4, December 2014



5962-0889
www.keysight.com