

Agilent Technologies
Modular
Solar Array Simulator
Series E4360

Programmer's Reference
Guide



Legal Notices

© Agilent Technologies, Inc. 2011

No part of this document may be photocopied, reproduced, or translated to another language without the prior agreement and written consent of Agilent Technologies, Inc. as governed by United States and international copyright laws.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Manual Editions

Part Number E4360-90902

Version 2.4, December 2011

Available in electronic format only.

Certification

Agilent Technologies certifies that this product met its published specifications at time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE THE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. AGILENT TECHNOLOGIES SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

This product comes with the standard product warranty. Warranty options, extended support contacts, product maintenance agreements and customer assistance agreements are also available. Contact your nearest Agilent Technologies Sales and Service office for further information on Agilent Technologies' full line of Support Programs.

Technologies Licenses

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

U.S. Government Restricted Rights

Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data – Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

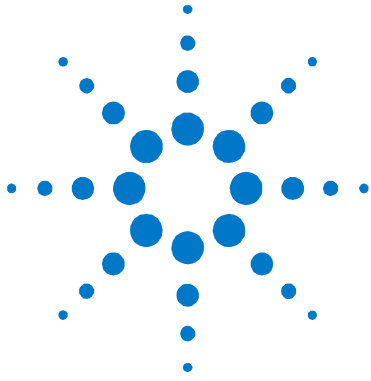
Trademarks

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

Contents

1 - Introduction	5
Instrument Drivers and IO Libraries.....	6
Instrument Web Server.....	6
Agilent E4360 Product Documentation	7
2 - Introduction to the SCPI Language	9
SCPI Commands.....	10
SCPI Messages.....	12
SCPI Conventions and Data Formats.....	14
SCPI Command Completion	16
3 - Commands by Subsystem	17
ABORt Commands Introduction	18
CALibrate Subsystem Introduction	22
Common Commands Introduction.....	34
DISPlay Commands Introduction.....	54
FETCh Subsystem Introduction	57
FORMat Commands Introduction.....	63
INITiate Subsystem Introduction	66
MEASure Subsystem Introduction.....	72
MEMory Subsystem Introduction	75
OUTPut Subsystem Introduction.....	85
SENSe Subsystem Introduction	93
SOURce Subsystem Introduction	96
STATus Subsystem Introduction.....	168
SYSTem Commands Introduction.....	192
TRIGger Subsystem Introduction	209
4 - Command Quick Reference	223
SCPI Command Summary.....	224
Non-Volatile Settings	229
Reset Settings (*RST).....	230
5 - Programming Examples	231
Output Programming Example (Fixed Mode only).....	232
List Programming Example.....	235
6 - SCPI Error Messages	241
Introduction.....	242
Error List	242

7 - Agilent E4350 Compatibility249
 Programming Differences 250



1 Introduction

Instrument Drivers and IO Libraries	6
Instrument Web Server	6
Agilent E4360 Product Documentation	7

This manual contains reference information to help you program the Agilent E4360 Modular Solar Array Simulator over the remote interface using the SCPI programming language. The Agilent E4360 supports the SCPI programming language on all of its remote I/O interfaces.

For additional information on the Agilent E4360, be sure to browse the Product Reference CD-ROM shipped with the instrument or go to www.agilent.com/find/E4360.

Instrument Drivers and IO Libraries

We have included the following drivers and libraries on the Product Reference CD-ROM shipped with your Agilent E4360.

- 14360 System Control Tools
- IVI-COM Drivers

Agilent IO Libraries are included on a separate *Automation-Ready CD-ROM*, which is also shipped with your Agilent E4360.

You can also download the drivers and libraries from the Agilent web site at www.agilent.com/find/inst_drivers.

Instrument Web Server

The Agilent E4360 has a built-in Web server that lets you control them directly from an internet browser on your computer. With the Web server, you can access the front panel control functions including the LAN configuration parameters.

NOTE

The built-in Web server only operates over the LAN interface. It requires Internet Explorer 6+ or Netscape 6.2+. You also need the Java (Sun) plug-in. This is included in the Java Runtime Environment. Refer to Sun Microsystem's website www.sun.com.

The Web server is enabled when shipped. To launch the Web server:

1. Establish a LAN interface connection from your computer to the Agilent E4360.
2. To launch the Web server, enter the instrument's IP address or fully-qualified hostname into the browser's Address field.
3. To begin controlling your instrument, click on the Browser Web Control button in the navigation bar on the left.
4. For additional Help, click on the Help with this Page button.

Agilent E4360 Product Documentation

The following additional documents are available on the Agilent E4360 Product Reference CD-ROM, shipped with the Agilent E4360.

- E4360 User's Guide
- E4360 Family Specifications Guide
- E4360 Service Guide
- 14360 User's Guide

NOTE

You can contact Agilent Technologies at one of the following telephone numbers for warranty, service, or technical support information.

In the United States: (800) 829-4444

In Europe: 31 20 547 2111

In Japan: 0120-421-345

Or use our Web link for information on contacting Agilent in your country or specific location: www.agilent.com/find/assist

Or contact your Agilent Technologies Representative.

The web contains the most up to date version of the manuals. Go to www.agilent.com/find/E4360 to get the latest version of the manuals.



2

Introduction to the SCPI Language

SCPI Commands	10
SCPI Messages	12
SCPI Conventions and Data Formats	14
SCPI Command Completion	16

This chapter contains a brief introduction to the SCPI Programming language.

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling test and measurement instruments. SCPI provides instrument control with a standardized command syntax and style, as well as a standardized data interchange format.

SCPI Commands

SCPI has two types of commands, common and subsystem.

- **Common commands** are defined by the IEEE 488.2 standard to perform common interface 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 instrument 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. Subsystem commands are arranged alphabetically according to the function they perform. The following figure shows a portion of a subsystem command tree, from which you access the commands located along the various paths. Some [optional] commands have been included for clarity.

```
:OUTput
  [:STATe] <Bool>, (@<chanlist>)
  :COUPle
    [STATe] <Bool>
      :CHANnel <NR1>{ ,<NR1>}
  :INHibit
    :MODE <mode>

:STATus
  :OPERation
    [:EVENT] ? (@<chanlist>)
    :CONDition? (@<chanlist>)
```

Multiple Commands in a Message

Multiple SCPI commands can be combined and sent as a single message with one message terminator. There are two important considerations when sending several commands within a single message:

- Use a semicolon to separate commands within a message.
- There is an implied header path that affects how commands are interpreted by the instrument.

The header path can be thought of as a string that gets inserted **before** each command within a message. For the first command in a message, the header path is a null string. For each subsequent command the header path is defined as the characters that make up the headers of the previous command in the message up to and including the last colon separator. An example of a message with two commands is:

```
OUTPut:STATe ON, (@1);PROTection:CLEar (@1)
```

which shows the use of the semicolon separating the two commands, and also illustrates the header path concept. Note that with the second command, the leading header "OUTPut" was omitted because

after the "OUTPut:STATe ON" command, the header path became defined as "OUTPut" and thus the instrument interpreted the second command as:

```
OUTPut:PROTection:CLEar (@1)
```

In fact, it would have been syntactically incorrect to include the "OUTPut" explicitly in the second command, since the result after combining it with the header path would be:

```
OUTPut:OUTPut:PROTection:CLEar (@1)
```

which is incorrect.

Moving Among Subsystems

In order to combine commands from different subsystems, you need to be able to reset the header path to a null string within a message. You do this by beginning the command with a colon (:), which discards any previous header path. For example, you could clear the output protection and check the status of the Operation Condition register in one message by using a root specifier as follows:

```
OUTPut:PROTection:CLEar (@1)::STATus:OPERation:CONDition? (@1)
```

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

```
VOLTage:LEVel 7.5,(@1);PROTection 10,(@1)::CURRENT:LEVel 0.5,(@1)
```

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

Including Common Commands

You can combine common commands with subsystem 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 header path*; you may insert them anywhere in the message.

```
OUTPut OFF,(@1);*RCL 1;OUTPut ON,(@1)
```

Using Queries

Observe the following precautions with queries:

- Add a blank space between the query indicator (?) and any subsequent parameter such as a channel list.
- Allocate the proper number of variables for the returned data.
- Read back all the results of a query before sending another command to the instrument. 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 another command. The following commands are coupled:

```
[SOURce:]CURRent:SAS:ISC, [SOURce:]CURRent:SAS:IMP,
[SOURce:]VOLTage:SAS:VMP, and [SOURce:]VOLTage:SAS:VOC.
```

For example, by sending

```
CURR:SAS:ISC 4, (@1); IMP 3, (@1); VOLT:SAS:VMP 40, (@1);
VOC 55, (@1); <NL>
```

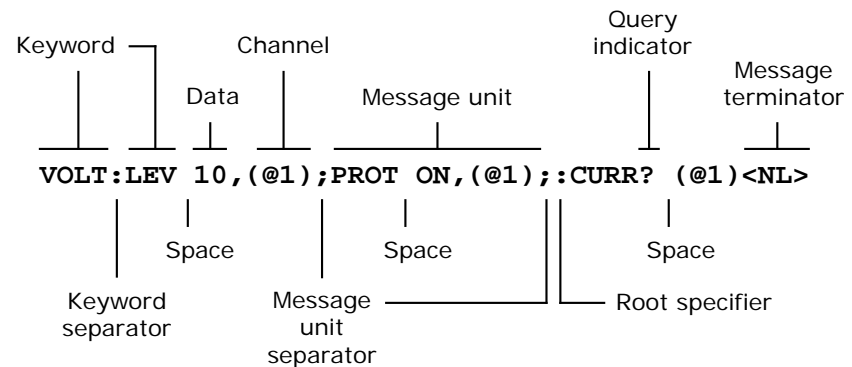
the instrument will be able to determine whether all four values are within acceptable parameter ranges. If one of the values is incorrect, no table will be generated.

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 instrument. The message, which may be sent at any time, requests the instrument to perform some action.
- A **response message** consists of data in a specific SCPI format sent from the instrument to the controller. The instrument sends the message only in response to a query header.

The following figure illustrates the SCPI message structure.



The Message Unit

The simplest SCPI command is a single message unit consisting of a command header (or keyword) followed by a message terminator such as a newline. The message unit may include a parameter after the header. The parameter can be numeric or a string.

```
*RST<NL>
VOLTage 20, (@1) <NL>
```

Channel List Parameter

The channel parameter is used to address one or more channels. This is an optional parameter. If a channel is not specified, the command is automatically directed to output channel 1. Additionally, if the output channels in a mainframe have been grouped to act as a single channel, commands sent without a channel parameter will program both grouped channels. You will need to use the channel parameter if you are programming output channels in a mainframe that are not grouped.

The channel parameter has the following syntax:

```
(@<channel> [ ,<channel> ])
```

You can also specify a range of sequential channels as follows:

```
(@<start_channel> : <end_channel>)
```

For example, (@2) specifies channel 2 and (@1:2) specifies channels 1 through 2. The channel list, shown as <chanlist> throughout this document, must be preceded with the @ symbol and must be enclosed in parentheses (). A maximum of 2 channels may be specified. Query results are channel list order-sensitive. Results are returned in the order they are specified in the list.

NOTE

When adding a channel list parameter to a query, you must include a space character between the query indicator (?) and the channel list parameter. Otherwise error -103, Invalid separator will occur.

Headers

Headers, also referred to as keywords, are instructions recognized by the instrument. Headers may be in the long form or in the short form. In the long form, the header is completely spelled out, such as VOLTAGE, STATUS, and DELAY. In the short form, the header has only the first three or four letters, such as VOLT, STAT, and DEL.

When the long form notation is used in this document, the capital letters indicate the equivalent short form. For example, MEASure is the long form, and MEAS indicates the short form equivalent.

Query Indicator

Following a header with a question mark turns it into a query (VOLTage?). The ? is the query indicator. If a query contains parameters, place the query indicator at the end of the last header, before the parameters.

```
VOLTage? MAX, (@1)
```

Message Unit Separator

When two or more message units are combined into a compound message, separate the units with a semicolon.

```
STATus:OPERation? (@1);QUEStionable? (@1)
```

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.

Message Terminator

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

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

In the examples of this guide, there is an assumed message terminator at the end of each message.

SCPI Conventions and Data Formats

Conventions

The following SCPI conventions are used throughout 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, VOLT CURR indicates that either "VOLT" or "CURR" can be used as a parameter.
Square brackets []	Items within square brackets are optional. The representation [SOURce:]VOLTage means that SOURce: may be omitted.
Parentheses ()	Items within parentheses are used in place of the usual parameter types to specify a channel list. The notation (@1,2) specifies a channel list that includes only channels 1 and 2.
Braces { }	Braces indicate parameters that may be repeated zero or more times. It is used especially for showing arrays. The notation <A>{} shows that parameter "A" must be entered, while parameter "B" may be omitted or may be entered one or more times.

Data Formats

Data programmed or queried from the instrument is ASCII. The data may be numerical or character string.

Numeric and Character Data Formats

Symbol	Description
<NR1>	Numeric value with an implied decimal point assumed at the right of the least-significant digit. Examples: 273
<NR2>	Numeric value with an explicit decimal point. Example: 27.3
<NR3>	Numeric value with an explicit decimal point and an exponent. Example: 2.73E+02
<NRf>	Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273 27.3 2.73E+02
<NRf+>	Expanded decimal format that includes <NRf> and MIN, MAX. Examples: 273 27.3 2.73E+02 MAX. MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter.
<Bool>	Boolean Data. They can be numeric (0, 1), or named (OFF, ON).
<SPD>	String Program Data. Predefined symbolic string parameter enclosed in single or double quotes.
<CPD>	Character Program Data. Predefined symbolic discrete parameter. Accepts both the short form and the long form.
<SRD>	String Response Data. Predefined symbolic string parameter enclosed in single or double quotes.
<CRD>	Character Response Data. Predefined symbolic discrete parameter. Only the short form of the parameter is returned.
<AARD>	Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
<Block>	Arbitrary Block Response Data. Permits the return of definite length and indefinite length arbitrary response data. This data type has an implied message terminator.

Suffixes and Multipliers

Class	Suffix	Unit	Unit with Multiplier
Current	A	ampere	MA (milliampere)
Amplitude	V	volt	UV (microvolt)
Time	S	second	KS (kilosecond)
Common Multipliers			
1E3	K	kilo	
1E-3	M	milli	
1E-6	U	micro	

SCPI Command Completion

SCPI commands sent to the instrument 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.

The following is a list of parallel commands. You should use some form of command synchronization as discussed in this section before assuming that these commands have completed.

OUTPut:STATE	INITiate
VOLTage	OUTPut:PROTEction:CLEar
CURRent	FUNCTion:MODE

The *WAI, *OPC, and *OPC? common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. Some practical considerations for using these commands are as follows:

- *WAI** This command prevents the instrument from processing subsequent commands until all pending operations are completed. For example, the *WAI command can be used to make a voltage measurement after an OUTPut ON command has completed:

```
OUTPut ON, (@1); *WAI; :MEASure:VOLTage? (@1)
```

- *OPC?** This command 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 command 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.

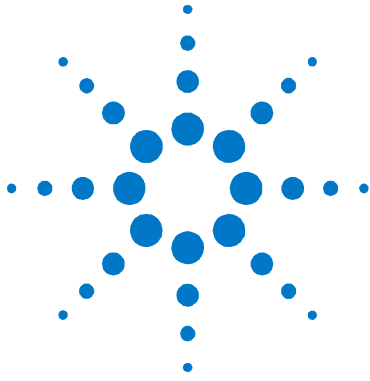
NOTE

The trigger subsystem must be in the Idle state for the status OPC bit to be true. As far as triggers are concerned, OPC is false whenever the trigger subsystem is in the Initiated state.

Device Clear

You can send a Device Clear at any time to abort a SCPI command that may be hanging up the GPIB interface. Device Clear clears the input and output buffers of the instrument and prepares the instrument to accept a new command string. The status registers, error queue, and all configuration states are left unchanged by Device Clear. The following statement shows how to send a device clear over the GPIB interface using Agilent Basic:

```
CLEAR 705 IEEE-488 Device Clear
```

3

Commands by Subsystem

ABORt Commands Introduction	18
CALibrate Subsystem Introduction	22
Common Commands Introduction	34
DISPlay Commands Introduction	54
FETCh Subsystem Introduction	57
FORMat Commands Introduction	63
INITiate Subsystem Introduction	66
MEASure Subsystem Introduction	72
MEMory Subsystem Introduction	75
OUTPut Subsystem Introduction	85
SENSe Subsystem Introduction	93
SOURce Subsystem Introduction	96
STATus Subsystem Introduction	168
SYSTem Commands Introduction	192
TRIGGer Subsystem Introduction	209

This section gives the syntax and parameters for all the IEEE 488.2 SCPI commands and the Common commands used by the instrument. It is assumed that you are familiar with the material in chapter 2, which explains the terms, symbols, and syntactical structures used here and gives an introduction to programming. You should also be familiar with chapter 4 in the User's Guide, in order to understand how the instrument functions.

Subsystem commands are specific to instrument 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 subsystem commands are arranged alphabetically according to the function they perform.

Common commands are defined by the IEEE 488.2 standard to perform common interface functions. They begin with an * and consist of three letters (command) or three letters and a ? (query). Common commands are grouped along with the subsystem commands according to the function they perform.

ABORt Commands Introduction

Abort commands cancel any triggered actions.

SCPI Command	Description
ABORt	
[:TRANsient] [(@chanlist)]	Resets the transient trigger system to the Idle state
:ACQuire [(@chanlist)]	Resets the measurement trigger system to the Idle state
:DLOG [(@chanlist)]	Stops the currently running datalog

ABORt[:TRANsient]

Syntax

```
ABORt[:TRANsient][(@<chanlist>)]
```

Description

This command cancels any triggered actions and returns the transient system back to the Idle state. ABORt:TRANsient also resets the WTG-tran bit in the Operation Condition Status register.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

If INITiate:CONTinuous ON has been turned on, ABORt does not turn off continuous triggers.

ABORt[:TRANsient] is also executed at power-on and upon execution of the [*RST](#) command.

Example

The following command aborts the triggered action of channel 1.

```
ABOR (@1)
```

See Also

[*RST](#)

[ABORt:ACQuire](#)

[INITiate:CONTinuous](#)

ABORt:ACQuire

Syntax

```
ABORt:ACQuire [(@<chanlist>)]
```

Description

This command cancels any triggered measurements and returns the trigger system back to the Idle state. It also resets the WTG-meas bit in the Condition Status register.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

ABORt:ACQuire is also executed at power-on and upon execution of the [*RST](#) command.

Example

The following command aborts the triggered action of channel 1.

```
ABOR:ACQ (@1)
```

See Also

[*RST](#)

[ABORt:TRANsient](#)

ABORt:DLOG

Syntax

```
ABORt:DLOG [(@<chanlist>)]
```

Description

This command stops the currently running data log and returns the trigger system back to the Idle state. It also resets the WTG-dlog bit in the Condition Status register and clears Busy-Dlog. Any remaining data can be fetched until the buffer is emptied.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Example

The following command stops the currently running data log of channel 1.

```
ABOR:DLOG (@1)
```

CALibrate Subsystem Introduction

This subsystem lets you calibrate the instrument. Only one channel can be calibrated at a time.

If calibration mode has not been enabled with CALibrate:STATE, the calibration commands will generate an error. Use CALibrate:SAVE to save any changes, otherwise all changes will be lost when you exit calibration mode.

NOTE

Refer to Chapter 3 in the Service Guide for a detailed description of the calibration procedure.

SCPI Command	Description
CALibrate	
:CURRent	
[:DATA] <NRf>, (@<channel>)	Enters the calibration value
:LEVel P1 P2, (@<channel>)	Calibrate the output current programming
:PROTection (@<channel>)	Calibrates the over-current protection
:DATE <"date">, (@<channel>)	Sets the calibration date
:PASSword <NRf>	Sets the numeric calibration password
:SAVE	Saves the new cal constants in non-volatile memory
:STATE <Bool> [,<NRf>]	Enables/disables calibration mode
:VOLTage	
[:DATA] <NRf>, (@<channel>)	Enters the calibration value
:LEVel P1 P2, (@<channel>)	Calibrate the output voltage programming
:PROTection (@<channel>)	Calibrates the over-voltage protection

CALibrate:CURRent[:DATA]**Syntax**

```
CALibrate:CURRent[:DATA] <value>, (@<channel>)
```

Description

This command enters a calibration value that you obtain by reading an external meter. You must first select a calibration level (with CALibrate:CURRent:LEVel) for the value being entered. Data values are expressed in amperes.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<value>	NRf	The measured current value.	Required parameter
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command enters a value measured by an external meter.

```
CAL:CURR 5.01, (@1)
```

See Also

[CALibrate:CURRent:LEVel](#)

CALibrate:CURRent:LEVel**Syntax**

```
CALibrate:CURRent:LEVel <level>, (@<channel>)
```

Description

This command selects the minimum or maximum calibration points. P1 and P2 are accepted as aliases for MIN and MAX respectively.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<level>	CPD	MIN P1 MAX P2	Required parameter
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command selects the minimum calibration point of channel 1.

```
CAL:CURR:LEV MIN, (@1)
```

See Also

[CALibrate:CURRent\[:DATA\]](#)

CALibrate:CURRent:PROTection

Syntax

```
CALibrate:CURRent:PROTection (@<channel>)
```

Description

This command calibrates the over-current protection. The peak current limit is calibrated automatically, at the same time as over-current protection.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command starts the over-current protection calibration of channel 1.

```
CAL:CURR:PROT (@1)
```

CALibrate:DATE

Syntax

```
CALibrate:DATE <"date">, (@<channel>)
```

```
CALibrate:DATE? (@<channel>)
```

Description

This command stores the date that the output module was last calibrated. The calibration date is stored in nonvolatile memory. Enter any ASCII string up to 16 characters. The query returns the date.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<"date">	SPD	String program data. String parameters must be enclosed in single or double quotes.	Required parameter
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Remarks

The firmware does not interpret the string format. The information is not used by the firmware. The command is only provided to store the calibration date.

Return Format

```
<SRD>
```

The query command returns an ASCII string enclosed in double quotes. If no date is stored, an empty quoted string ("") is returned.

Example

The following command enters a date for channel 1.

```
CAL:DATE "3/01/08", (@1)
```

The following query returns the calibration date.

```
CAL:DATE? (@1)
```

Typical Response: "3/22/09"

CALibrate:PASSword

Syntax

CALibrate:PASSword <password>

Description

This command lets you change the calibration password. The 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 calibration mode is unrestricted. The factory-default password 0 (zero).

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<password>	NRf	A numeric value up to 15 digits in length.	Required parameter

Example

The following command sets the password to a value of 1234.

```
CAL:PASS 1234
```

CALibrate:SAVE

Syntax

```
CALibrate:SAVE
```

Description

This command saves calibration constants in non-volatile memory after the calibration procedure has been completed. If calibration mode is exited by programming CALibration:STAtE OFF without first saving the new constants, the previous constants are restored.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Example

The following command saves the calibration values.

```
CAL:SAVE
```

See Also

[CALibrate:STAtE](#)

CALibrate:STATe

Syntax

```
CALibrate:STATe <Bool>, [<password>]
```

```
CALibrate:STATe?
```

Description

This command enables or disables calibration mode. Calibration mode must be enabled for the instrument to accept any calibration commands. The first parameter specifies the ON (1) or OFF (0) state. The second parameter is the password.

A numeric password is required if 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.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF
<password>	NRf	A numeric value up to 15 digits in length.	Required if password not 0

Remarks

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

Return Format

```
<NR1>
```

The query returns only the calibration state, not the password. It returns either 0 (OFF), or 1 (ON).

Example

The following command enables calibration mode; the password is 1234.

```
CAL:STAT ON, 1234
```

The following query returns the calibration state.

```
CAL:STAT?
```

Typical Response: 1

See Also

[CALibrate:SAVE](#)

CALibrate:VOLTage[:DATA]**Syntax**

```
CALibrate:VOLTage[:DATA] <value>, (@<channel>)
```

Description

This command enters a calibration value that you obtain by reading an external meter. You must first select a calibration level (with CALibrate:VOLTage:LEVel) for the value being entered. Data values are expressed in volts.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<value>	NRf	The measured voltage value.	Required parameter
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command enters a value measured by an external meter.

```
CAL:VOLT 65.1, (@1)
```

See Also

[CALibrate:VOLTage:LEVel](#)

CALibrate:VOLTage:LEVel**Syntax**

```
CALibrate:VOLTage:LEVel <level>, (@<channel>)
```

Description

This command selects the minimum or maximum calibration points. P1 and P2 are accepted as aliases for MIN and MAX respectively.

NOTE

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<level>	CPD	MIN P1 MAX P2	Required parameter
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command selects the minimum calibration point of channel 1.

```
CAL:VOLT:LEV MIN, (@1)
```

See Also

[CALibrate:VOLTage\[:DATA\]](#)

CALibrate:VOLTage:PROTection

Syntax

CALibrate:VOLTage:PROTection (@<channel>)

Description

This command calibrates the over-voltage protection.

NOTE

The output will go near full-scale during this procedure. The output should be left open-circuit.

Refer to Calibration in the Service Guide for a detailed description of the calibration procedure.

Parameters

Name	Type	Range of Values	Default Value
<channel>	NR1	The output channel that is being calibrated.	Required parameter

Example

The following command starts the over-voltage protection calibration of channel 1.

```
CAL:VOLT:PROT (@1)
```

Common Commands Introduction

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

SCPI Command	Description
*CLS	Clear status
*ESE <NRf>	Standard event status enable
*ESR?	Return event status register
*IDN?	Return instrument identification
*OPC	Enable "operation complete" bit in ESR
*OPT?	Return option number
*RCL <NRf>	Recalls a saved instrument state
*RDT?	Return output channel descriptions
*RST	Reset
*SAV <NRf>	Saves an instrument state
*SRE <NRf>	Set service request enable register
*STB?	Return status byte
*TRG	Trigger
*TST?	Selftest
*WAI	Pauses additional command processing until all device commands are done

CLS*Syntax**

*CLS

Description

This command causes the following actions on the status system:

Clears the Standard Event Status, Operation Status Event, and Questionable Status Event registers.

Clears the Status Byte and the Error Queue.

If *CLS immediately follows a program message terminator (<NL>), then the output queue and the MAV bit are also cleared.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Example

The following command clears the event register bits.

```
*CLS
```

ESE*Syntax**

*ESE <value>

*ESE?

Description

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.

The query reads the enable register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NRf	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Required parameter.

For example, to enable bit 2 (decimal value = 4), bit 3 (decimal value = 8), and bit 7 (decimal value = 128), the corresponding decimal value would be 140 (4 + 8 + 128).

Remarks

The following table lists the bit definitions for the Standard Event Register.

Bit Number	Decimal Value	Definition
0 Operation Complete	1	All commands prior to and including *OPC have been executed.
1 Not Used	Not Used	0 is returned.
2 Query Error	4	The instrument tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3 Device-Dependent Error	8	A self-test or calibration error occurred. For a complete listing of the error messages, see SCPI Error Messages .

Bit Number	Decimal Value	Definition
4 Execution Error	16	An execution error occurred (an error in the -200 range has been generated).
5 Command Error	32	A command syntax error occurred (an error in the -100 range has been generated).
6 Not Used	Not Used	0 is returned.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

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.

The [*CLS](#) (clear status) command will not clear the enable register but it does clear all bits in the event register.

A [*STATUS:PRESet](#) command does not clear the bits in the Status Byte register.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 3 (decimal value = 8) and bit 7 (decimal value = 128) are enabled, the query command will return 136.

Example

The following command enables bit 4 (decimal value = 16) in the enable register.

```
*ESE 16
```

The following query returns which bits are enabled in the register.

```
*ESE?
```

Typical Response: +16

See Also

[*ESR?](#)

ESR?*Syntax**

*ESR?

Description

This query returns the value of the *Event* register of the Standard Event Status group. The Event register is a read-only register, which stores (latches) all standard events. Reading the Standard Event Status *Event* register clears it.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Remarks

The following table lists the bit definitions for the Standard Event Register.

Bit Number	Decimal Value	Definition
0 Operation Complete	1	All commands prior to and including *OPC have been executed.
1 Not Used	Not Used	0 is returned.
2 Query Error	4	The instrument tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3 Device-Dependent Error	8	A self-test or calibration error occurred. For a complete listing of the error messages, see SCPI Error Messages .
4 Execution Error	16	An execution error occurred (an error in the -200 range has been generated).
5 Command Error	32	A command syntax error occurred (an error in the -100 range has been generated).
6 Not Used	Not Used	0 is returned.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

To be reported to the Standard Event Register, the corresponding bits in the event register must be enabled using the [*ESE](#) command.

Once a bit is set, it remains set until cleared by reading the event register or the [*CLS](#) (clear status) command.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 3 (decimal value = 8) and bit 7 (decimal value = 128) are enabled, the query command will return 136.

Example

The following command reads the event register (bits 3 and 4 are set).

```
*ESR?
```

Typical Response: +24

See Also

[*ESE](#)

*IDN?

Syntax

*IDN?

Description

This command reads the instrument's (mainframe) identification string which contains four comma-separated fields. The first field is the manufacturer's name, the second field is the instrument model number, the third field is the serial number, and the fourth field is the firmware revision.

Agilent Technologies	Manufacturer
E4360A	Mainframe model number
MYxxxxxxx	Mainframe serial number if available, or zero
X.xx.xx	Revision levels of firmware

Return Format

<AARD>

The command returns an arbitrary ASCII string with four comma-separated fields. To read the string into your computer, be sure to dimension a string variable with at least 50 characters.

Example

The following query returns the instrument's identification string.

```
*IDN?
```

Typical Response:

```
AGILENT TECHNOLOGIES,E4360A,MY00123456,A.00.00
```


OPC*Syntax**

*OPC

*OPC?

Description

The command is mainly used for program synchronization. It causes the instrument to set the OPC bit (bit 0) of the Standard Event Status register when the instrument has completed all pending operations sent before *OPC. Pending operations are complete when:

1. All commands sent before *OPC, including paralleled commands, have been completed. Most commands are sequential and are completed before the next command is executed. Commands that affect output voltage, current, or state, relays, and trigger actions are executed in parallel with subsequent commands. *OPC provides notification that all parallel commands have completed.
2. All triggered actions are completed.

The query returns 1 to the output buffer.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Remarks

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

*OPC? causes the instrument to place a 1 in the output buffer when all pending operations are completed. *OPC? does not suspend processing of commands.

Return Format

<NR1>

Returns 1 to the output buffer.

Example

The following command sets the operation complete bit.

*OPC

The following command waits until the completion of the current command and then sends 1 to the output buffer.

*OPC?

***OPT?**

Syntax

*OPT?

Description

This query requests the mainframe to identify any installed options.

Remarks

A 0 indicates no options are installed.

Return Format

<CRD>

The option number.

Example

The following command returns the option number.

```
*OPT?
```

Typical Response: +0

*RCL

Syntax

```
*RCL 0 | 1
```

Description

This command restores the instrument to a state that was previously stored in memory locations 0 through 1 with the *SAV command. All instrument states are recalled except for the following:

The trigger system is set to the Idle state by an implied ABORt command (this cancels any uncompleted trigger actions).

Calibration is disabled by setting CALibration:STATe to OFF.

All list settings are set to their *RST values.

Remarks

The device state stored in location 0 is automatically recalled at power turn-on when the Output Power-On state is set to RCL 0.

A Factory Reset (*RST command) does not affect the configurations stored in memory. Once a state is stored, it remains until it is overwritten.

Example

The following command recalls the instrument state previously stored in location 1.

```
*RCL 1
```

See Also

[*SAV](#)

[*RST](#)

*RDT?

Syntax

*RDT?

Description

This query returns a description of all the output modules (channels) installed in a mainframe. Semicolons separate multiple channel descriptions.

CHAN<c>:description <c> refers to the channel number.
Description returns the model number of the output modules.

Return Format

<AARD>

The command returns an arbitrary ASCII string with up to four semicolon-separated fields. To read the string into your computer, be sure to dimension a string variable with at least 50 characters.

Example

The following query identifies the mainframe's installed output modules.

```
*RDT?
```

Typical Response: CHAN1:E4361A;CHAN2:E4361A

*RST

Syntax

```
*RST
```

Description

This command resets the volatile memory of the instrument to a factory-defined state. Refer to [Factory Reset State](#) for a complete listing of the instrument's factory configuration.

Remarks

*RST also forces the ABORT:ACQuire and ABORT:TRANsient commands. This cancels any measurement or output trigger actions presently in process, and resets the two WTG bits in the Status Operation Condition register.

Example

The following command resets the instrument.

```
*RST
```

See Also

[ABORT:ACQuire](#)

[ABORT:TRANsient](#)

*SAV

Syntax

```
*SAV 0 | 1
```

Description

This command stores the present state of the instrument to the specified location in non-volatile memory. Up to 2 states can be stored - in locations 0 and 1. Any state previously stored in the same location will be overwritten. Use the [*RCL](#) command to retrieve instrument states. Refer to [Factory Reset State](#) for a list of instrument settings that can be saved.

CAUTION

This command causes a write cycle to nonvolatile memory. Nonvolatile memory has a finite maximum number of write cycles. Programs that repeatedly cause write cycles to nonvolatile memory can eventually exceed the maximum number of write cycles and cause the memory to fail.

Remarks

If a particular state is desired at power-on, it should be stored in location 0. It will then be automatically recalled at power turn-on if the Output Power-On state is set to RCL0.

User tables, list data and the calibration state is **NOT** saved as part of the *SAV operation. This means that all table and list data that is sent to the instrument will be lost when the instrument is turned off.

Data saved in non-volatile memory, described in the Non-volatile Factory Settings, is not affected by the *SAV command.

A Factory Reset ([*RST](#) command) does not affect the configurations stored in memory. Once a state is stored, it remains until it is overwritten.

Example

The following command stores the current instrument state in location 1.

```
*SAV 1
```

See Also

[*RCL](#)

[*RST](#)

SRE*Syntax**

*SRE <value>

*SRE?

Description

This command sets the value of the Service Request Enable register. This register determines which bits from the Status Byte register are summed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in the bit position enables the corresponding event.

The query reads the enable register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NRf	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Required parameter.

For example, to enable bit 1 (decimal value = 2), bit 3 (decimal value = 8), and bit 6 (decimal value = 64), the corresponding decimal value would be 74 (2 + 8 + 64).

Remarks

The following table lists the bit definitions for the Status Byte Register.

Bit Number	Decimal Value	Definition
0 Not Used	Not Used	0 is returned.
1 Not Used	Not Used	0 is returned.
2 Error Queue not empty	4	One or more errors have been stored in the Error Queue (see SYSTEM:ERRor? command).
3 Questionable Status summary	8	One or more bits are set in the Questionable Data Register (bits must be enabled, see STATus:QUEStionable:ENABle command).
4 Message Available	16	Data is available in the instrument's output buffer.

3 Commands by Subsystem

Bit Number	Decimal Value	Definition
5 Event Status Byte summary	32	One or more bits are set in the Standard Event Register (bits must be enabled, see *ESE command).
6 Master Status summary (Request for Service)	64	One or more bits are set in the Status Byte Register (bits must be enabled, see *SRE command). Also used to indicate a request for service.
7 Operation Status summary	128	One or more bits are set in the Standard Operation Register (bits must be enabled, see STATus:OPERation:ENABle command).

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. All such enabled bits are then logically ORed to cause the MSS bit (bit 6) of the Status Byte Register to be set.

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 power system cannot generate an SRQ to the controller.

The [*CLS](#) (clear status) command will not clear the enable register but it does clear all bits in the event register.

A [STATus:PRESet](#) command does not clear the bits in the Status Byte register.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 3 (decimal value = 8) and bit 7 (decimal value = 128) are enabled, the query command will return "+136".

Examples

The following command enables bit 4 (decimal value = 16) in the enable register.

```
*SRE 16
```

The following query returns which bits are enabled in the register.

```
*SRE?
```

Typical Response: +16

See Also

[*STB?](#)

***STB?**

Syntax

*STB?

Description

This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. The Status Byte is a read-only register and the bits are *not* cleared when it is read.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Remarks

The following table lists the bit definitions for the Status Byte Register.

Bit Number	Decimal Value	Definition
0 Not Used	Not Used	0 is returned.
1 Not Used	Not Used	0 is returned.
2 Error Queue not empty	4	One or more errors have been stored in the Error Queue (see SYSTEM:ERROR? command).
3 Questionable Status summary	8	One or more bits are set in the Questionable Data Register (bits must be enabled, see STATus:QUEStionable:ENABle command).
4 Message Available	16	Data is available in the instrument's output buffer.
5 Event Status Byte summary	32	One or more bits are set in the Standard Event Register (bits must be enabled, see *ESE command).
6 Master Status summary (Request for Service)	64	One or more bits are set in the Status Byte Register (bits must be enabled, see *SRE command). Also used to indicate a request for service.
7 Operation Status summary	128	One or more bits are set in the Standard Operation Register (bits must be enabled, see STATus:OPERation:ENABle command).

The input summary bits are cleared when the appropriate event registers are read. The MAV bit is cleared at power-on, by *CLS, or when there is no more response data available.

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 instrument has one or more reasons for requesting service.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 1 (decimal value = 2) and bit 4 (decimal value = 16) are set (and the corresponding bits are enabled), this command will return 18.

Example

The following command reads the condition register (bits 3 and 4 are set).

```
*STB?
```

Typical Response: +24

See Also

[*SRE](#)

TRG*Syntax**

*TRG

Description

This common command generates a trigger when the trigger subsystem has BUS selected as its source. The command has the same affect as the Group Execute Trigger (<GET>) command.

Example

The following command generates a trigger.

```
*TRG
```

See Also

[ABORt:ACQuire](#)

[ABORt:TRANsient](#)

[INITiate\[:IMMediate\]:ACQuire](#)

[INITiate\[:IMMediate\]:TRANsient](#)

*TST?

Syntax

*TST?

Description

This query causes the instrument to do a self-test and report any errors. A 0 indicates the instrument passed self-test. If all tests pass, you can have a high confidence that the instrument is operational.

Remarks

If one or more tests fail, a 1 is returned and an error is stored in the error queue. For a complete listing of the error messages related to self-test failures, see [SCPI Error Messages](#).

If one or more tests fail, see the Service Guide for instructions on returning the instrument to Agilent for service.

*TST? also forces an *RST command.

Return Format

<NR1>

The command returns 0 (all tests passed) or 1 (one or more tests failed).

Example

The following command performs a self-test and returns a pass/fail indication.

```
*TST?
```

Typical Response: +0

*WAI

Syntax

```
*WAI
```

Description

This command instructs the instrument not to process any further commands until all pending operations are completed. Pending operations are as defined under the [*OPC](#) command.

Remarks

*WAI can be aborted only by sending the instrument a [Device Clear](#) command.

Example

The following command waits for pending operations to complete.

```
*WAI
```

See Also

[*OPC](#)

DISPlay Commands Introduction

The Display commands let you control the front panel display.

SCPI Command	Description
DISPlay	
[:WINDow]	
:TEXT <SPD>	Sends the text to be displayed on the front panel
:VIEW CHAN1 CHAN2 ALL TEXT	Selects the channels displayed in meter view
ENABle <Bool>	Enables/disables the front panel display

DISPlay[:WINDow]:TEXT

Syntax

```
DISPlay[:WINDow]:TEXT <text>
```

```
DISPlay[:WINDow]:TEXT?
```

Description

This command sends a character string to the display when the [DISPlay:VIEW](#) is set to TEXT. The string can include any ASCII 7-bit character, including nonprintable characters. Nonprintable characters will be displayed as the “?” character. The string must be enclosed in either single (‘) or double (“) quotes. The maximum length of the string is 254 characters, the display is capable of showing up to approximately 160 proportional font characters. Only the first four lines of auto-wrap text are displayed.

Parameters

Name	Type	Range of Values	Default Value
<“text”>	SPD	String program data	Required parameter

Return Format

```
<CRD>
```

The query command returns the text string.

Example

The following command sends a string with double quotes enclosed in single quotes.

```
DISP:TEXT 'String with double quotes ("...")
enclosed in single quotes'
```

The following command sends a string with single quotes enclosed in double quotes.

```
DISP:TEXT "String with single quotes ('...')
enclosed in double quotes"
```

The following query returns the front panel test string.

```
DISP:TEXT?
```

Typical Response: "TEXT"

DISPlay[:WINDow]:VIEW**Syntax**

```
DISPlay[:WINDow]:VIEW <view>
```

```
DISPlay[:WINDow]:VIEW?
```

Description

This command selects the information to be displayed on the front panel. CHAN1 displays output channel 1. CHAN2 displays output channel 2. ALL displays both output channels. Text displays the text specified by the [DISPlay:TEXT](#) command.

Parameters

Name	Type	Range of Values	Default Value
<view>	CPD	CHAN1 CHAN2 ALL TEXT	ALL

Return Format

```
<CRD>
```

The query command returns CHAN1, CHAN2, ALL, or TEXT.

Example

The following command sets the display to two-channel view.

```
DISP:VIEW ALL
```

The following query returns the front-panel display setting.

```
DISP:VIEW?
```

Typical Response: ALL

DISPlay:ENABle

Syntax

DISPlay:ENABle <Bool>

DISPlay:ENABle?

Description

This command turns the front panel display on or off

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	ON

Return Format

<NR1>

The query returns only the display state; either 0 (OFF), or 1 (ON).

Example

The following command turns the front panel display off.

```
DISP:ENAB OFF
```

See Also

[DISPlay\[:WINDow\]:VIEW](#)

FETCh Subsystem Introduction

Fetch commands return measurement data that has been previously acquired. Because output voltage and current are measured simultaneously, you take a voltage measurement and then fetch the current data that was collected along with the voltage measurement. If you take a current measurement, you can then fetch the voltage data. You can also use Fetch commands to return measurement data that has been synchronized with a triggered event.

SCPI Command	Description
FETCh	
[:SCALar]	
:CURRent[:DC]? [(@chanlist)]	Returns the average output current
:VOLTage[:DC]? [(@chanlist)]	Returns the average output voltage
:DLOG [(@chanlist)]	Returns measurements from datalog buffer

FETCh[:SCALar]:CURRent[:DC]?**Syntax**

```
FETCh[:SCALar]:CURRent[:DC]? [(@<chanlist>)]
```

Description

This query returns the DC output current in amperes. The data returned is the result of the last measurement command or triggered acquisition. The data is valid until the next MEASure or INITiate command occurs.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The FETCh query does not perform a new measurement.

Both voltage and current are measured whenever a measurement is executed; therefore, both quantities can also be fetched after any measurement.

Return Format

```
<NR3>
```

The reading is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Example

The following query returns the current measured on channel 1.

```
FETC:CURR? (@1)
```

Typical Response: +1.000000E+01

See Also

[FETCh\[:SCALar\]:VOLTage\[:DC\]?](#)

FETCh[:SCALar]:VOLTage[:DC]?**Syntax**

```
FETCh[:SCALar]:VOLTage[:DC]? [(@<chanlist>)]
```

Description

This query returns the DC output voltage in volts. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition. The data is valid until the next MEASure or INITiate command occurs.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The FETCh query does not perform a new measurement.

Both voltage and current are measured whenever a measurement is executed; therefore, both quantities can also be fetched after any measurement.

Return Format

```
<NR3>
```

The reading is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Example

The following query returns the voltage measured on channel 1.

```
FETCh:VOLT? (@1)
```

Typical Response: +5.000000E+01

See Also

[FETCh\[:SCALar\]:CURRent\[:DC\]?](#)

FETCh:DLOG?

Syntax

```
FETCh:DLOG? <maxrecords> [ ,(@<chanlist>)]
```

Description

This query returns buffered measurements from the datalog buffer. <maxrecords> is the desired number of measurement records to return. If there are fewer than <maxrecords> records available, all available records will be returned. Reading buffered measurements can occur while new measurements are being buffered. Doing this clears up buffer space for new measurements. This allows continuous acquisition to be implemented.

Data is returned in the following order:

```
<meas_volt>,<meas_curr>[,<meas_volt>,<meas_curr>]
```

Parameters

Name	Type	Range of Values	Default Value
<maxrecords>	Numeric	1 - 65535 MAX	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The data can be retrieved in ASCII or BINary format.

The data type for <meas_volt> and <meas_curr> are float32 and the ASCII response is in NR3 format.

Multi-channel query is not supported in ASCII mode. A separate transaction must be used to get data from each channel.

In REAL format, data values are returned in definite length arbitrary block response data (AARD) format.

Return Format

```
<NR3>
```

The reading is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Example

The following query requests 1000 records of logged data on channel 1.

```
FETCh:DLOG? 1000 ,(@1)
```

3 Commands by Subsystem

Typical Response:

```
+6.100000E+01,+3.200000E+00.....,+6.097836E+01,3.199  
57E+00
```

FORMat Commands Introduction

The FORMat commands let you specify the format for transferring datalog measurements.

SCPI Command	Description
FORMat	
[:DATA] ASCII REAL	Selects the transferred data format
:BORDER NORMAL SWAPPED	Specifies how binary data is transferred

FORMat[:DATA]**Syntax**

```
FORMat[:DATA] <type> [ ,(@<chanlist>)]
```

```
FORMat[:DATA]? [(@<chanlist>)]
```

Description

This command specifies the format of the data returned by queries that can return a block of data.

ASCII Numeric data is transferred as ASCII bytes in <NR1>, <NR2>, or <NR3> format as appropriate. The numbers are separated by commas.

REAL Data is transferred in a definite-length block as floating point numbers of the 32 bits.

Parameters

Name	Type	Range of Values	Default Value
<type>	CPD	ASCII REAL	ASCII
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<CRD>
```

The query command returns the data format setting. Multiple responses are separated by commas.

Examples

The following command sets the data format of channel 1 to ASCII.

```
FORM ASCII, (@1)
```

The following query returns the data format setting of channel 1.

```
FORM? (@1)
```

Typical Response: ASCII

See Also

[FORMat:BORDER](#)

FORMat:BORDER

Syntax

FORMat:BORDER <setting> [,(@<chanlist>)]

FORMat:BORDER? [(@<chanlist>)]

Description

This command specifies how binary data is transferred.

NORMAL Binary data is transferred in normal order. The most significant byte is returned first, and the least significant byte is returned last (big-endian).

SWAPped Binary data is transferred in swapped-byte order. The least significant byte is returned first, and the most significant byte is returned last (little-endian).

Parameters

Name	Type	Range of Values	Default Value
<setting>	CPD	NORMAL SWAPped	NORMAL
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<CRD>

The query command returns the binary byte order setting.

Examples

The following command sets the byte order setting of channel 1 to SWAPped.

```
FORM:BORD SWAP , (@1)
```

The following query returns the byte order setting of channel 1.

```
FORM:BORD? (@1)
```

Typical Response: SWAP

See Also

[FORMat\[:DATA\]](#)

INITiate Subsystem Introduction

Initiate commands initialize the trigger system. This enables the trigger system to receive triggers.

SCPI Command	Description
INITiate	
[:IMMEDIATE]	
[:TRANSient] [(@chanlist)]	Enables output triggers
:ACQuire [(@chanlist)]	Enables measurement triggers
:DLOG [(@chanlist)]	Enables the datalogger
:CONTinuous	
[:TRANSient] <Bool>, [(@chanlist)]	Enables/disables continuous transient triggers

INITiate[:IMMediate][:TRANsient]

Syntax

```
INITiate[:IMMediate][:TRANsient] [(@<chanlist>)]
```

Description

This command enables the output trigger system by moving it from the Idle state to the Initiated state. At this time the validity of the list is also checked. Once initiated, a trigger event on the selected trigger source triggers the list and then returns the trigger system to the Idle state. If the trigger system is not initiated, all triggers are ignored.

NOTE

This command only applies when the SAS mode is set to List operation using the [\[SOURCE:\]CURRENT:SAS:MODE](#) command.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

During list validation, the value of each data point in the list as well as the list length is checked. The process can take some time if the list is long. Validation is complete when the WTG-tran status bit is set true.

If the list validation fails, an error will be generated and the trigger system remains in the Idle state.

Example

The following command initiates the transient trigger system on channel 1.

```
INIT (@1)
```

See Also

[INITiate\[:IMMediate\]:ACQUIRE](#)

[ABORT:TRANsient](#)

INITiate[:IMMediate]:ACQuire

Syntax

```
INITiate[:IMMediate]:ACQuire [(@<chanlist>)]
```

Description

This command enables the measurement trigger system by moving it from the Idle state to the Initiated state. Once initiated, a trigger event on the selected trigger source causes the measurement to be triggered. If the trigger system is not initiated, all triggers are ignored.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

It takes a few milliseconds for the instrument to be ready to receive a trigger signal after receiving the INITiate:ACQuire command.

If a trigger occurs before the trigger system is ready for it, the trigger will be ignored. The user can test the WTG_meas bit in the operation status register to know when the instrument is ready to receive a trigger after initiating.

Example

The following command initiates the measurement trigger system on channel 1.

```
INIT:ACQ (@1)
```

See Also

[INITiate\[:IMMediate\]:TRANSient](#)

[ABORt:ACQuire](#)

INITiate[:IMMediate]:DLOG

Syntax

```
INITiate[:IMMediate]:DLOG [(@<chanlist>)]
```

Description

This command enables the data logger by moving it from the Idle state to the Initiated state. Once initiated, a trigger event on the selected data log trigger source causes the data log to be triggered. If the data logger is not initiated, all triggers are ignored.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

It takes a few milliseconds for the instrument to be ready to receive a trigger signal after receiving the INITiate:DLOG command.

If a trigger occurs before the trigger system is ready for it, the trigger will be ignored. The user can test the WTG_dlog bit in the operation status register to know when the instrument is ready to receive a trigger after initiating.

Example

The following command initiates the data logger on channel 1.

```
INIT:DLOG (@1)
```

See Also

[TRIGger:DLOG\[:IMMediate\]](#)

[ABORt:DLOG](#)

INITiate:CONTInuous[:TRANSient]**Syntax**

```
INITiate:CONTInuous[:TRANSient] <Bool>,
[(@<chanlist>)]

INITiate:CONTInuous[:TRANSient]? [(@<chanlist>)]
```

Description

This command continuously initiates the output trigger system. This allows for multiple output transients without having to initiate the transient system after each run. The enabled state is ON (1); the disabled state is OFF (0).

NOTE

This command only applies when the SAS mode is set to List operation using the [\[SOURCE:\]CURRENT:SAS:MODE](#) command.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

If INITiate:CONTInuous is turned off, the output trigger system will return to the Idle state after the list completes.

If INITiate:CONTInuous is turned on, [ABORT:TRANSient](#) will NOT return the transient system to the Idle state. INITiate:CONTInuous must be turned off prior to sending ABORT:TRANSient.

Return Format

```
<NR1>
```

The query command returns 0 if continuous transients are disabled (OFF), and 1 if continuous transients are enabled (ON).

Example

The following command continually initiates the transient trigger system on channel 1.

```
INIT:CONT:TRAN ON, (@1)
```

The following query returns the continuous transient trigger state.

```
INIT:CONT:TRAN? (@1)
```

Typical Response: 0

See Also

[ABORT:TRANsient](#)

MEASure Subsystem Introduction

Measure commands measure the output voltage and current. They trigger the acquisition of new data before returning the reading. Measurements are performed by digitizing the instantaneous output voltage and current for a specified measurement time, storing the results in a buffer, and calculating the average value..

Use Measure commands when the output measurements do not need to be synchronized with any other event. Use INITiate:ACQuire and TRIGger commands when measurements are to be synchronized.

SCPI Command	Description
MEASure	
[:SCALar]	
:CURRent[:DC]? [(@chanlist)]	Takes a measurement; returns the average output current
:VOLTage[:DC]? [(@chanlist)]	Takes a measurement; returns the average output voltage

MEASure[:SCALar]:CURRent[:DC]?

Syntax

```
MEASure[:SCALar]:CURRent[:DC]? [(@<chanlist>)]
```

Description

This query initiates and triggers a measurement, and returns the DC output current in amperes.

NOTE

Output voltage and current is measured simultaneously. You can use [FETCh:VOLTage?](#) to return the voltage data that was collected along with the current measurement.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The MEASure query starts a new measurement immediately. It is not synchronized to any trigger event

Use the INITiate, trigger, and FETCh sequence if a synchronized measurement is necessary.

The default measurement time interval is 1.7 milliseconds.

Return Format

```
<NR3>
```

The reading is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Example

The following query returns the voltage measured on channel 1.

```
MEAS:CURR? (@1)
```

Typical Response: +1.000000E+01

See Also

[MEASure\[:SCALar\]:VOLTage\[:DC\]?](#)

[FETCh\[:SCALar\]:VOLTage\[:DC\]?](#)

MEASure[:SCALar]:VOLTage[:DC]?**Syntax**

```
MEASure[:SCALar]:VOLTage[:DC]? [(@<chanlist>)]
```

Description

This query initiates and triggers a measurement, and returns the DC output voltage in volts.

NOTE

Output voltage and current is measured simultaneously. You can use [FETCh:CURRent?](#) to return the current data that was collected along with the voltage measurement).

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The MEASure query starts a new measurement immediately. It is not synchronized to any trigger event

Use the INITiate, trigger, and FETCh sequence if a synchronized measurement is necessary.

The default measurement time interval is 1.7 milliseconds.

Return Format

```
<NR3>
```

The reading is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Example

The following query returns the voltage measured on channel 1.

```
MEAS:VOLT? (@1)
```

Typical Response: +5.000000E+01

See Also

[MEASure\[:SCALar\]:CURRent\[:DC\]?](#)

[FETCh\[:SCALar\]:CURRent\[:DC\]?](#)

MEMory Subsystem Introduction

MMEMory commands provide mass storage capability for output measurement data. The mass storage may be either internal or external to the instrument. The mass storage device is selected using the full path embedded within the filename. The filename parameter is a string.

SCPI Command	Description
MEMory	
:COPY	
:TABLE <CPD>	Copies the selected table to non-volatile memory
:DElete	
[:NAME] <CPD>	Deletes the specified table in volatile and non-volatile memory
:ALL	Deletes all tables in volatile and non-volatile memory
:TABLE	
:CATalog?	Returns all table names in volatile and non-volatile memory
:CURRent	
[:MAGnitude] <NRf>,{<NRf>}	Programs a list of current points for a new table
:POINts?	Returns the number of current points in the active table
:SElect [<CPD>]	Creates a new table in volatile memory
:VOLTage	
[:MAGnitude] <NRf>,{<NRf>}	Programs a list of voltage points for a new table
:POINts?	Returns the number of voltage points in the active table

MEMory:COpy:TABLE

Syntax

```
MEMory:COpy:TABLE <name>
```

Description

This command copies the table that was selected with [MEMory:TABLE:SElect](#) to non-volatile memory. You can use the same name or a different name. Table names cannot be longer than 12 alphanumeric characters and must start with an alpha character. A maximum of 30 tables can be stored in non-volatile memory. Tables can have from 3 to 4000points.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter

Remarks

If a table with the same name already exists in non-volatile memory, it will be over-written with the new table.

After many writes to non-volatile memory, the time it takes to write to memory may increase up to one second per write operation. If you are copying many tables, this may take up to 10 seconds. To allow faster writes to non-volatile memory when this occurs, you must [sanitize](#) and then [initialize](#) the flash memory.

Example

The following command copies the nable named curve1.

```
MEM:COpy:TABL curve1
```

See Also

[MEMory:TABLE:SElect](#)

MEMory:DElete:ALL

Syntax

```
MEMory:DElete:ALL
```

Description

This command deletes all tables in both volatile and non-volatile mass memory. The command fails if a table is currently selected or activated with the [MEMory:TABLE:SElect](#) or [\[SOURce\]:\[CURRent\]:TABLE:NAME](#) command. If the command fails, no tables are deleted.

Example

The following command deletes all tables in volatile and non-volatile memory.

```
MEM:DEL:ALL
```

See Also

[MEMory:DElete\[:NAME\]](#)

[MEMory:TABLE:SElect](#)

[\[SOURce\]:\[CURRent\]:TABLE:NAME](#)

MEMory:DELeTe[:NAME]**Syntax**

```
MEMory:DELeTe[:NAME] <name>
```

Description

This command deletes the specified table in both volatile and non-volatile memory. This command will delete one table per execution. If a table is stored in volatile memory and also stored in non-volatile memory with the same name, the volatile copy will be deleted first. A second MEM:DEL[:NAME] must be sent to remove the table from non-volatile memory.

NOTE

This command can only be programmed when the instrument is operating in Fixed mode (see [\[SOURCE:\]CURRENT:MODE](#)). An error will be generated if this command is sent when the instrument is operating in SAS or in Table mode..

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter

Example

The following command deletes the table named curve1.

```
MEM:DEL curve1
```

See Also

[MEMory:DELeTe:ALL](#)

[\[SOURCE:\]CURRENT:TABLE:NAME](#)

MEMory:TABLE:CATalog?

Syntax

```
MEMory:TABLE:CATalog?
```

Description

This query returns the names of all user-defined tables in both volatile and non-volatile memory. The table names are enclosed in quotes.

Return Format

```
<SRD>
```

The defined tables in quotes. Multiple responses are separated by commas. ("") is the end of the list.

Example

The following example returns three table names.

```
MEM:TABLE:CAT?
```

Typical Response: "Table1","Table2","Table3",""

MEMory:TABLE:CURRent[MAGNitude]

Syntax

```
MEMory:TABLE:CURRent[:MAGNitude]
<current>{,<current>}
```

Description

This command programs a list of current points for a new table. A table must have at least 3 points. You may, but are not required to send all the table points in one command. Repeatedly sending this command will append additional points to the end of the table, up to a maximum of 4000 points. A maximum of 30 tables can be stored in volatile memory.

To copy a table from volatile to non-volatile memory, use [MEMory:TABLE:SElect](#) and [MEMory:COPY:TABLE](#).

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf	0 to the maximum current rating. Current values must be entered in equal or descending order, otherwise an error will occur.	0

Remarks

<SRD>

If you are programming a follower channel that is paralleled with a master channel, you must send the same values to the follower channel.

Example

The following command programs ten data points in the new table.

```
MEM:TABLE:CURR 7.8,7.7,7.5,7,6.5,5.5,4,3,2,1
```

See Also

[MEMory:TABLE:CURRent\[:MAGNitude\]:POINTs?](#)

MEMory:TABLE:CURRent[:MAGNitude]:POINts?

Syntax

```
MEMory:TABLE:CURRent[:MAGNitude]:POINts?
```

Description

This query returns the number of current points in the active table.

Return Format

```
<NR1>
```

Example

```
MEM:TABLE:CURR:POIN?
```

Typical Response: 10

See Also

[MEMory:TABLE:CURRent\[:MAGNitude\]](#)

MEMory:TABLE:SElect

Syntax

MEMory:TABLE:SElect [<name>]

Description

This command creates a new table in volatile memory to be programmed with a list of voltage and current points. Table names cannot be longer than 12 alphanumeric characters and must start with an alpha character. Table names are converted to upper case.

This command is also used to select an existing table in non-volatile memory to be copied into volatile memory for appending additional voltage and current points. A selected table can be copied into non-volatile memory using [MEMory:COPY:TABLE](#).

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Optional parameter

Remarks

If you send this command without specifying a table name, it will de-select the selected table.

To activate a table when the unit is operating in Table mode, use [SOURce:]CURRent:TABLE:NAME.

Return Format

<SRD>

The presently selected table in quotes. Returns ("") if no table is selected.

Example

The following command selects or creates the table named curve1.

```
MEM:TABLE:SEL curve1
```

The following command de-selects the presently selected table.

```
MEM:TABLE:SEL
```

See Also

[MEMory:COPY:TABLE](#)

[\[SOURce:\]CURRent:TABLE:NAME](#)

MEMory:TABLE:VOLTage[MAGNitude]

Syntax

```
MEMory:TABLE:VOLTage[:MAGNitude]
<voltage>{,<voltage>}
```

Description

This command programs a list of voltage points for a new table. A table must have at least 3 points. You may, but are not required to send all the table points in one command. Repeatedly sending this command will append additional points to the end of the table, up to a maximum of 4000 points. A maximum of 30 tables can be stored in volatile memory.

To copy a table from volatile to non-volatile memory, use [MEMory:TABLE:SElect](#) and [MEMory:COPY:TABLE](#).

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf	0 to the maximum voltage rating. Voltage values must be entered in equal or descending order, otherwise an error will occur.	0

Remarks

<SRD>

If you are programming a follower channel that is paralleled with a master channel, you must send the same values to the follower channel.

Example

The following command programs ten data points in the new table.

```
MEM:TABLE:VOLT 5,35,45,50,52,54,56,57,58,59
```

See Also

[MEMory:TABLE:VOLTage\[:MAGNitude\]:POINTS?](#)

MEMory:TABLE:VOLTage[:MAGNitude]:POINts?

Syntax

```
MEMory:TABLE:VOLTage[:MAGNitude]:POINts?
```

Description

This query returns the number of voltage points in the active table.

Return Format

```
<NR1>
```

Example

```
MEM:TABLE:VOLT:POIN?
```

Typical Response: 10

See Also

[MEMory:TABLE:VOLTage\[:MAGNitude\]](#)

OUTPut Subsystem Introduction

The Output subsystem controls the output, power-on, protection, and relay functions.

SCPI Command	Description
OUTPut	
[:STATe] <Bool> [,@chanlist]	Enables/disables the specified output channel(s)
:COUPle	
[:STATe] <Bool>	Enables/disables channel coupling for output synchronization
:CHANNeL [<NR1> {,<NR1>}]	Selects which channels are coupled
:INHibit	
:MODE LATCHing LIVE OFF	Sets the remote inhibit input
:PON	
:STATe RST RCL0	Programs the power-on state
:PROTection	
:CLEar [(@chanlist)]	Resets latched protection
:COUPle <Bool>	Enables/disables channel coupling for protection faults

OUTPut[:STATe]

Syntax

```
OUTPut[:STATe] <Bool> [ ,(@<chanlist>)]
```

```
OUTPut[:STATe]? [(@<chanlist>)]
```

Description

This command enables or disables the specified output channel(s). The enabled state is ON (1); the disabled state is OFF (0). The state of a disabled output is a condition of zero output voltage and zero source current.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

Because of internal circuit start-up procedures, Output On may take up to 75 milliseconds to complete its function. Conversely, Output Off may take up to 75 milliseconds to complete its function. To mitigate these built-in delays, you can program the output to zero volts rather than using the output on/off function.

Return Format

```
<NR1>
```

The query command returns 0 if the output is OFF, and 1 if the output is ON.

Example

The following command turns the output of channel 2 on.

```
OUTP ON, (@2)
```

The following query returns the output state of channel 2.

```
OUTP? (@2)
```

Typical Response: 0

See Also

[OUTPut:PON:STATe](#)

OUTPut[:STATe]:COUPle[:STATe]

Syntax

```
OUTPut[:STATe]:COUPle[:STATe] <Bool>
```

```
OUTPut[:STATe]:COUPle[:STATe]?
```

Description

This command enables or disables output coupling or synchronization between multiple output channels. When output channels have been coupled and the Digital connector pins have been properly configured, enabling this function allows you to control the output state of multiple outputs in multiple mainframes by controlling the output state of one output channel. Refer to Appendix B in the User's Guide for detailed information.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF

Remarks

When this command is sent, all output channels in the mainframe will go to the output Off state.

This setting is non-volatile and not affected by *RST.

Return Format

```
<NR1>
```

The query command returns 1 if outputs are coupled, and 0 if outputs are not coupled.

Example

The following command turns output coupling on.

```
OUTP:COUP ON
```

The following query returns the output coupling state.

```
OUTP:COUP?
```

Typical Response: 0

See Also

[OUTPut\[:STATe\]:COUPle:CHANnel](#)

[\[SOURce:\]DIGital:PIN<1-7>:FUNCTION](#)

OUTPut[:STATe]:COUPle:CHANnel

Syntax

```
OUTPut[:STATe]:COUPle:CHANnel [<channel>
{,<channel>}]
```

```
OUTPut[:STATe]:COUPle:CHANnel?
```

Description

This command specifies which output channels in the mainframe are controlled by the output synchronization function. The output channels that have been synchronized or coupled will turn on and off together when any one of them is turned on or off, or when a signal is received from a digital connector pin that has been configured as an On couple or an Off couple pin. Refer to Appendix B in the User's Guide for detailed information.

If you do not specify an output channel, this command will uncouple all output channels on the mainframe.

Parameters

Name	Type	Range of Values	Default Value
<channel>	NR1	1 2	Required parameter

Remarks

When this command is sent, all output channels in the mainframe will go to the output Off state.

This setting is non-volatile and not affected by *RST.

Return Format

```
<NR1>
```

The query command returns the channels that are coupled. Multiple responses are separated by commas.

Example

The following command couples channels 1 and 2.

```
OUTP:COUP:CHAN 1,2
```

The following query returns the output coupling state.

```
OUTP:COUP:CHAN?
```

Typical Response: 1,2

See Also

[OUTPut\[:STATe\]:COUPle\[:STATe\]](#)

[\[SOURce:\]DIGital:PIN<1-7>:FUNCTION](#)

OUTPut:INHibit:MODE

Syntax

```
OUTPut:INHibit:MODE <mode>
```

```
OUTPut:INHibit:MODE?
```

Description

This command selects the mode of operation of the Inhibit input (INH). The inhibit function shuts down ALL output channels in response to an external signal on the Inhibit input. If an output channel has been turned off by OUTPut[:STATe], the inhibit function does not affect the output channel while it is in the OFF state. The Inhibit mode setting is stored in non-volatile memory.

The following modes can be selected:

LATChing Causes a logic-true transition on the Inhibit input to disable all outputs. The outputs remain disabled until the Inhibit input is returned to logic-false and the latched INH status bit is cleared by sending the OUTPut:PROTection:CLEar command or a protection clear command from the front panel.

OFF The Inhibit input is ignored.

Parameters

Name	Type	Range of Values	Default Value
<mode>	CPD	LATChing OFF	Required parameter

Return Format

```
<CRD>
```

The query command returns LATC or OFF.

Example

The following command sets the Inhibit Input to Latching mode.

```
OUTP:INH:MODE LATC
```

The following query returns the setting of the Inhibit Input.

```
OUTP:INH:MODE?
```

Typical Response: OFF

See Also

[OUTPut\[:STATe\]](#)

[OUTPut:PROTection:CLEar](#)

OUTPut:PON:STATe

Syntax

```
OUTPut:PON:STATe <state>
```

```
OUTPut:PON:STATe?
```

Description

This command determines if the power-on state is set to the *RST (RST) state or the instrument state stored in memory location 0 (RCL0). The parameter is saved in non-volatile memory. Instrument states can be stored using the *SAV command.

When shipped from the factory, the [*RST](#) command is recalled at power-on.

Parameters

Name	Type	Range of Values	Default Value
<state>	CPD	RST RCL0	Required parameter

Return Format

```
<CRD>
```

The query command returns RST or RCL0.

Example

The following command sets the power-on state to the *RST state.

```
OUTPut:PON:STAT RST
```

The following query returns the setting of the power-on state.

```
OUTPut:PON:STAT?
```

Typical Response: RST

See Also

[*RCL](#)

[*RST](#)

[*SAV](#)

OUTPut:PROTection:CLEar

Syntax

```
OUTPut:PROTection:CLEar [(@<chanlist>)]
```

Description

This command clears the latched protection status that disables the output when an over-voltage, over-current, over-temperature, over-switching, inhibit signal, power-fail condition, or a power-limit condition is detected.

All conditions that generate the fault must be removed before the latched status can be cleared. The output is restored to the state it was in before the fault condition occurred.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Example

The following command clears the latched protection status of channel 1.

```
OUTP:PROT:CLE (@1)
```

See Also

[OUTPut:PROTection:COUple](#)

OUTPut:PROTection:COUPle

Syntax

```
OUTPut:PROTection:COUPle <Bool>
```

```
OUTPut:PROTection:COUPle?
```

Description

This command enables or disables output coupling for protection faults. When enabled, ALL output channels are disabled when a protection fault occurs on any output channel. The enabled state is On (1); the disabled state is Off (0). When disabled, only the affected output channel is disabled when a protection fault is triggered.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF

Return Format

```
<NR1>
```

The query command returns 0 if the protection coupling is OFF (disabled), and 1 if the protection coupling is ON (enabled).

Example

The following command enables protection coupling.

```
OUTP:PROT:COUP ON
```

The following query returns the protection coupling setting.

```
OUTP:PROT:COUP?
```

Typical Response: 0

See Also

[OUTPut:PROTection:CLEar](#)

SENSe Subsystem Introduction

The SENSe commands let you control the datalog measurement buffer.

SCPI Command	Description
SENSe	
:DLOG	
:CLOCK TRIGger TIMer [,@chanlist]	Specifies the measurement capture trigger source
:TINTerval <NRf+> [,@chanlist]	Sets the time interval for capturing measurements

SENSe:DLOG:CLOCK

Syntax

```
SENSe:DLOG:CLOCK <setting> [ ,(@<chanlist>)]
```

```
SENSe:DLOG:CLOCK? [ (@<chanlist>)]
```

Description

This command specifies the trigger source for capturing measurements into the data log.

TIMer Specifies that measurements are captured in the data log using the trigger event specified in the [TRIGger:DLOG:SOURce](#) command.

TRIGger Specifies that measurements are captured in the data log using the internal event timer specified in the [SENSe:DLOG:TINterval](#) command

Parameters

Name	Type	Range of Values	Default Value
<setting>	CPD	TIMer TRIGger	TRIGger
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<CRD>
```

The query command returns the trigger source setting. Multiple responses are separated by commas.

Example

The following command sets the trigger source of channel 1 to TIMer, causing datalog measurements to be triggered by the event timer.

```
SENS:DLOG:CLOC TIM,(@1)
```

The following query returns the trigger source of channel 1.

```
SENS:DLOG:CLOC? (@1)
```

Typical Response: TRIG

See Also

[SENSe:DLOG:TINterval](#)

[TRIGger:DLOG:SOURce](#)

SENSe:DLOG:TINteval

Syntax

```
SENSe:DLOG:TINteval <interval> [ ,(@<chanlist>)]
```

```
SENSe:DLOG:TINteval? [(@<chanlist>)]
```

Description

This command specifies the time interval for capturing measurements into the datalog buffer. The time interval can be specified in seconds, from 0.02 through 65, with a resolution of 0.02 seconds.

Parameters

Name	Type	Range of Values	Default Value
<interval>	NRf	0.02 - 65	0.1
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

This command can only be used if the [SENSe:DLOG:CLOCK TIME](#) setting has been specified.

Return Format

```
<NR3>
```

The query command returns the programmed time interval in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Example

The following command programs a time interval of 1 second on channel 1.

```
SENS:DLOG:TINT 1,(@1)
```

The following query returns the time interval of channel 1.

```
SENS:DLOG:TINT? (@1)
```

Typical Response: +2.000000E-02

See Also

[SENSe:DLOG:CLOCK](#)

SOURce Subsystem Introduction

The Source subsystem programs the current, digital, list, step, and voltage functions.

SCPI Command	Description
[SOURce:]	
CURRent	
[:LEVel]	
[:IMMEdiate][:AMPLitude] <NRf+>, (@chanlist)	Sets the output current in Fixed mode
:DTABle	
:SASimulator	
[:IMMEdiate]? [(@chanlist)]	Returns the calculated DAC table used for the SAS settings
[:IMMEdiate]:IMP? [(@chanlist)]	Returns the calculated DAC table Imp value for the SAS settings
[:IMMEdiate]:ISC? [(@chanlist)]	Returns the calculated DAC table Isc value for the SAS settings
:LIST? <NR1> [,@chanlist]	Returns the calculated DAC table used for the specified step
:LIST:IMP? <NR1> [,@chanlist]	Returns the calculated DAC table Imp value for the specified step
:LIST:ISC? <NR1> [,@chanlist]	Returns the calculated DAC table Isc value for the specified step
:TABLe	
[:IMMEdiate]? <CPD> [,@chanlist]	Returns the calculated DAC table used for the specified table
[:IMMEdiate]:IMP? <CPD> [,@chanlist]	Returns the calculated DAC table Imp value for the specified table
[:IMMEdiate]:ISC? <CPD> [,@chanlist]	Returns the calculated DAC table Isc value for the specified table
:MODE FIXed SAS TABLe [,@chanlist]	Sets the operating mode of the instrument
:DTABle 256 4096 [,@<chanlist>]	Specifies the size of the table in points
:PROGramming INTernal EXTernal [,@<chanlist>]	Specifies the programming source in auto-parallel operation
:PROTection	
[:LEVel] <NRf+> [,@chanlist]	Sets the over-current protection level
:SAS	
:IMP <NRf> [,@chanlist]	Sets the current at the peak power point of the curve
:ISC <NRf> [,@chanlist]	Sets the short-circuit current
:MODE IMMEdiate LIST [,@chanlist]	Selects the source of the curve parameters
:SLIMit	
:HIGH <NRf+> MAXimum [,@chanlist]	Sets the high soft limit for programming the output current
:TABLe	
:NAME <CPD> [,@<chanlist>]	Activates a user-defined table in Table mode
:OFFSet <NRf> [,@<chanlist>]	Adds a current offset when operating in Table mode
DIGital	
:INPut:DATA?	Reads the state of the digital port pins
:OUTPut:DATA <NRf>	Sets the digital port
:PIN<1-7>	
:FUNCTion DIO DINPut TOUTput TINPut FAULt INHibit ONCOuple OFFCOuple	Sets the selected pin's function
:POLarity POSitive NEGative	Sets the selected pin's polarity (List commands only on N676xA, N678xA SMU and Option 054)
LIST	
:COUNT <NRf+> INFInity, (@chanlist)	Sets the list repeat count
:DWELI <NRf> {,<NRf>}, (@chanlist)	Sets the list of dwell times
:POINts? (@chanlist)	Returns the number of dwell list points
:SAS	
:IMP	
[:LEVel] <NRf>{,<NRf>} [,@chanlist]	Sets the list of peak power current points
:POINts? (@chanlist)	Returns the number of peak power current points

SCPI Command	Description
LIST:SAS continued	
:ISC	
[:LEVel] <NRf>{,<NRf>} [,@chanlist]	Sets the list of short-circuit current points
:POINts? (@chanlist)	Returns the number of short-circuit points
:VMP	
[:LEVel] <NRf>{,<NRf>} [,@chanlist]	Sets the list of peak power voltage points
:POINts? (@chanlist)	Returns the number of peak power voltage points
:VOC	
[:LEVel] <NRf>{,<NRf>} [,@chanlist]	Sets the list of open-circuit voltage points
:POINts? (@chanlist)	Returns the number of open-circuit points
:STEP ONCE AUTO, (@chanlist)	Specifies how the list responds to triggers
:TERMinate	
:LAST <Bool>, (@chanlist)	Sets the list termination mode
VOLTage	
[:LEVel]	
[:IMMEDIATE][:AMPLitude] <NRf+>, (@chanlist)	Sets the output voltage in Fixed mode
:DTABLE	
:SASimulator	
[:IMMEDIATE]? [(@chanlist)]	Returns the calculated DAC table used for the SAS settings
[:IMMEDIATE]:VMP? [(@chanlist)]	Returns the calculated DAC table Vmp value for the SAS settings
[:IMMEDIATE]:VOC? [(@chanlist)]	Returns the calculated DAC table Voc value for the SAS settings
:LIST? <NR1> [,@chanlist]	Returns the calculated DAC table used for the specified step
:LIST:VMP? <NR1> [,@chanlist]	Returns the calculated DAC table Vmp value for the specified step
:LIST:VOC? <NR1> [,@chanlist]	Returns the calculated DAC table Voc value for the specified step
:TABLE	
[:IMMEDIATE]? <CPD> [,@chanlist]	Returns the calculated DAC table used for the specified table
[:IMMEDIATE]:VMP? <CPD> [,@chanlist]	Returns the calculated DAC table Vmp value for the specified table
[:IMMEDIATE]:VOC? <CPD> [,@chanlist]	Returns the calculated DAC table Voc value for the specified table
:PROTection	
[:LEVel] <NRf>, (@chanlist)	Sets the over-voltage protection level
:SAS	
:VMP <NRf>, (@chanlist)	Sets the voltage at the peak power point of the curve
:VOC <NRf>, (@chanlist)	Sets the open-circuit voltage
:SLIMit	
:HIGH <NRf+> MAXimum [,@chanlist]	Sets the high soft limit for programming the output voltage
:TABLE	
:OFFSet <NRf> [,@<chanlist>]	Adds a voltage offset when operating in Table mode

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]**Syntax**

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
<current> [,(@<chanlist>)]
```

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
[MIN|MAX,] [(@<chanlist>)]
```

Description

This command sets the immediate current level of the specified output channel. The immediate level is the output current setting.

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf+	0 - maximum MIN MAX The maximum value is dependent on the current rating of the output module.	0 A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

This command is applies in FIXed mode only.

Return Format

```
<NR3>
```

The query command returns the programmed current level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the output current to 3 A on channels 1 and 2.

```
CURR 3,(@1,2)
```

The following query returns the programmed current level on channels 1 and 2.

```
CURR? (@1,2)
```

Typical Response: +3.000000E+00,+3.000000E+00

See Also

[\[SOURce:\]VOLTage\[:LEVel\]\[:IMMediate\]](#)

[SOURce:]CURRent:DTABle:SASimulator[:IMMediate]?**Syntax**

```
[SOURce:]CURRent:DTABle:SASimulator[:IMMediate]?
[(@<chanlist>)]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings. The number of current points returned is based on the [\[SOURce:\]CURRent:MODE:DTABle](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table settings of channel 2.

```
CURR:DTAB:SAS? (@2)
```

Typical Response: 4.8,4.9,5.0,5.1,5.2,5.3, ...

See Also

[\[SOURce:\]CURRent:MODE:DTABle](#)

[SOURce:]CURRent:DTABLE:SASimulator[:IMMediate]:IMP?**Syntax**

```
[SOURce:]CURRent:DTABLE:SASimulator[:IMMediate]:IMP?
[(@<chanlist>)]
```

Description

This query returns the calculated Imp (maximum power current) of the internal DAC table used for the SAS settings. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Imp for channel 2 at a 256 point resolution with an SAS setting of:

```
Vmp = 55 V
Imp = 3.5 A
Voc = 60 V
Isc = 4 A
```

```
CURR:DTAB:SAS:IMP? (@2)
```

Typical Response: +3.620843E+00

See Also

[\[SOURce:\]CURRent::DTABLE:SASimulator:ISC?](#)

[SOURCE:]CURRENT:DTABLE:SASimulator[:IMMEDIATE]:ISC?**Syntax**

```
[SOURCE:]CURRENT:DTABLE:SASimulator[:IMMEDIATE]:ISC?
[(@<chanlist>)]
```

Description

This query returns the calculated Isc (short-circuit current) of the internal DAC table used for the SAS settings. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Isc for channel 2 at a 256 point resolution with an SAS setting of:

```
Vmp = 55 V
Imp = 3.5 A
Voc = 60 V
Isc = 4 A
```

```
CURRENT:DTABLE:SAS:ISC? (@2)
```

Typical Response: +5.10000E+00

See Also

[\[SOURCE:\]CURRENT:DTABLE:SASimulator:IMP?](#)

[SOURCE:]CURRENT:DTABLE:SASimulator:LIST?**Syntax**

```
[SOURCE:]CURRENT:DTABLE:SASimulator:LIST? <step>
[ , (@<chanlist>)]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings of the specified SAS list step. The number of current points returned is based on the [\[SOURCE:\]CURRENT:MODE:DTABLE](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table of the 4th step for channel 1 at a 256 point resolution.

```
CURRENT:DTABLE:SAS:LIST? 3, (@1)
```

Typical Response: 4.8,4.9,5.0,5.1,5.2,5.3, ...

See Also

[\[SOURCE:\]CURRENT:MODE:DTABLE](#)

[SOURce:]CURRent:DTABle:SASimulator:LIST:IMP?**Syntax**

```
[SOURce:]CURRent:DTABle:SASimulator:LIST:IMP? <step>
[ , (@<chanlist>)]
```

Description

This query returns the calculated Imp (maximum power current) of the internal DAC table used for the specified SAS list step. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the Imp value of the 4th step for channel 1 at a 256 point resolution with an SAS setting of

Vmp = 40 V

Imp = 4.08 A

Voc = 60 V

Isc = 5.1 A

```
CURR:DTAB:SAS:LIST:IMP? 3, (@1)
```

Typical Response: +3.847693E+00

See Also

[\[SOURce:\]CURRent:DTABle:SASimulator:LIST:ISC?](#)

[SOURce:]CURRent:DTABle:SASimulator:LIST:ISC?**Syntax**

```
[SOURce:]CURRent:DTABle:SASimulator:LIST:ISC? <step>
[ , (@<chanlist>)]
```

Description

This query returns the calculated Isc (short-circuit current) value of the internal DAC table used for the specified SAS list step. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the Isc value of the 4th step for channel 1 at a 256 point resolution with an SAS setting of

Vmp = 40 V

Imp = 4.08 A

Voc = 60 V

Isc = 5.1 A

```
CURR:DTAB:SAS:LIST:ISC? 3, (@1)
```

Typical Response: +5.10000E+00

See Also

[\[SOURce:\]CURRent:DTABle:SASimulator:LIST:IMP?](#)

[SOURCE:]CURRENT:DTABLE:TABLE[:IMMEDIATE]?**Syntax**

```
[SOURCE:]CURRENT:DTABLE:TABLE[:IMMEDIATE]? <name>
[ , (@<chanlist>)]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings in the specified table. The number of current points returned is based on the [\[SOURCE:\]CURRENT:MODE:DTABLE](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table values for the table named curve1.

```
CURR:DTAB:TABL? curve1 (@1)
```

Typical Response: 4.8,4.9,5.0,5.1,5.2,5.3, ...

See Also

[\[SOURCE:\]CURRENT:MODE:DTABLE](#)

[SOURce:]CURRent:DTABle:TABLE[:IMMediate]:IMP?**Syntax**

```
[SOURce:]CURRent:DTABle:TABLE[:IMMediate]:IMP?
<name> [, (@<chanlist>)]
```

Description

This query returns the calculated Imp (maximum power current) value of the internal DAC table for the specified table. Note that the resolution of the DAC table can affect the accuracy of the calculation..

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Imp for channel 1 at a 256 point resolution for the table named sample2:

```
Voltage Current
1          1.5
4          1.45
8          1.4
12         1.35
16         1.3
20         1.2
24         1
```

```
CURR:DTAB:TABL:IMP? sample2,(@1)
```

Typical Response: +1.10029E+00

See Also

[\[SOURce:\]CURRent::DTABle:TABLE:ISC?](#)

[SOURce:]CURRent:DTABle:TABLE[:IMMEDIATE]:ISC?**Syntax**

```
[SOURce:]CURRent:DTABle:TABLE[:IMMEDIATE]:ISC?
<name> [ ,(@<chanlist>)]
```

Description

This query returns the calculated Isc (short-circuit current) value of the internal DAC table for the specified table. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Isc for channel 1 at a 256 point resolution for the table named sample2:

```
Voltage Current
1         1.5
4         1.45
8         1.4
12        1.35
16        1.3
20        1.2
24        1
```

```
CURR:DTAB:TABL:ISC? sample2,(@1)
```

Typical Response: +1.500000E+00

See Also

[\[SOURce:\]CURRent::DTABle:TABLE:IMP?](#)

[SOURce:]CURRent:MODE**Syntax**

```
[SOURce:]CURRent:MODE <mode> [ ,(@<chanlist>)]
```

```
[SOURce:]CURRent:MODE? [(@<chanlist>)]
```

Description

This command selects the operating mode of the instrument.

- FIXed** In Fixed mode, the output is a fixed rectangular I-V characteristic.
- SAS** Simulator mode uses an exponential model to approximate the I-V curve. It is programmed in terms of its open circuit voltage (Voc), short circuit current (Isc), voltage point (Vmp), and current point (Imp) at approximately the peak power point.
- TABLE** Table mode sets the output characteristics by a user-defined table of voltage/current points. A curve is generated by connecting the points by straight lines. Therefore, the more points that are provided, the more the connected points will resemble a curve.

Parameters

Name	Type	Range of Values	Default Value
<mode>	CPD	FIXed SAS TABLE	FIXed
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

When selecting Table mode, you must first activate a table using the [\[SOURce:\]CURRent:TABLE:NAME](#) command. Otherwise Error 103 is generated if no table has been activated.

When a table has been selected, the combination of Table value + Offset for voltage and current points are checked to not exceed any maximum. If validity check passes, the table will be used.

Return Format

```
<CRD>
```

The query command returns the operating mode for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the current mode of channel 2.

```
CURR:MODE SAS, (@2)
```

The following query returns the operating mode of channel 1.

```
CURR:MODE? (@1)
```

Typical Response: FIX

See Also

[\[SOURCE:\]CURRENT:TABLE:NAME](#)

[SOURce:]CURRent:MODE:DTABLE**Syntax**

```
[SOURce:]CURRent:MODE:DTABLE <tablesize>
[ , (@<chanlist>)]

[SOURce:]CURRent:MODE:DTABLE? [(@<chanlist>)]
```

Description

This command specifies the size of the data table used in SAS and Table modes. Allowable sizes are 256 points or 4096 points. The larger table allows greater resolution of the I-V characteristic, but takes longer to calculate and activate. The smaller table is faster to calculate and activate, at the expense of I-V curve resolution..

NOTE

This command can only be programmed when the instrument is operating in Fixed mode (see [\[SOURce:\]CURRent:MODE](#)). An error will be generated if this command is sent when the instrument is operating in SAS or in Table mode

Parameters

Name	Type	Range of Values	Default Value
<tablesize>	NR1	256 4096	4096
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

Not all of the points in the DAC table are available for approximating the I-V curve. Because of circuit requirements, portions of the DAC table are reserved for internal use. The number of points available for curve approximation is about 3600 points for high resolution mode and 230 points for low resolution mode. For all practical purposes however, the entire DAC table at its respective resolution is used in SAS mode.

Return Format

<CRD>

The query command returns the current mode for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the data table to 256 points.

```
CURR:MODE:DTAB 256 , (@2)
```

The following query returns the size of the data table.

```
CURR:MODE:DTAB? (@1)
```

Typical Response: 4096

See Also

[\[SOURCE:|CURRENT:MODE](#)

[SOURce:]CURRent:MODE:PROGramming**Syntax**

```
[SOURce:]CURRent:MODE:PROGramming <source>
[ , (@<chanlist>)]

[SOURce:]CURRent:MODE:PROGramming? [ (@<chanlist>)]
```

Description

When multiple output channels are connected for auto-parallel operation, this command specifies the source of the control signal used for current programming. This determines if the is channel is a "master" or "follower" channel.

INTernal The control signal for current programming is sourced from the internal circuits of the channel. This designates the master channel in auto-parallel operation. This control signal is available on the SAS pin of the master channel's analog connector for controlling the follower channels

EXTernal The control signal for current programming is sourced from the IP+ pin on the analog connector of the follower channel in auto-parallel operation

NOTE

This command can only be programmed when the instrument is operating in Fixed mode (see [\[SOURce:\]CURRent:MODE](#)). An error will be generated if this command is sent when the instrument is operating in SAS or in Table mode

Parameters

Name	Type	Range of Values	Default Value
<source>	CPD	INTernal FIXed	INTernal
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The same Table values, SAS values, or SAS lists must be sent to all master and follower channels.

Return Format

<CRD>

The query command returns the source of the control signal for each channel specified. Multiple responses are separated by commas.

Examples

The following command specifies channel 1 as the master channel in an auto-parallel configuration.

```
CURR:MODE:PROG INT, (@1)
```

The following query returns the source of the control signal.

```
CURR:MODE:PROG? (@1)
```

Typical Response: INT

[SOURce:]CURRent:PROTection[:LEVel]**Syntax**

```
[SOURce:]CURRent:PROTection[:LEVel] <current>
[,(@<chanlist>)]

[SOURce:]CURRent:PROTection[:LEVel]? [MIN|MAX,]
[(@<chanlist>)]
```

Description

This command sets the over-current protection level. The values are programmed in amperes. If the output current exceeds the OCP level, the output is disabled and the Questionable Condition status register OC bit is set. Over-current protection is active in all modes of operation.

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf+	0 - maximum MIN MAX The maximum value is dependent on the current rating of the output module.	1.1 X MAX
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

An over-current condition can be cleared with the OUTPut:PROTection:CLEar command after the condition that caused the OVP trip is removed.

Return Format

<NR3>

The query command returns the protection level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the protection level to 3 A on channels 1 and 2.

```
CURR:PROT:LEV 3,(@1,2)
```

The following query returns the protection level on channels 1 and 2.

```
CURR:PROT:LEV? (@1,2)
```

Typical Response: +3.000000E+00,+3.000000E+00

See Also

[OUTPut:PROTection:CLEar](#)

[SOURce:]CURRent:SAS:IMP**Syntax**

```
[SOURce:]CURRent:DTABLE:SAS:IMP <current>
[ , (@<chanlist>)]

[SOURce:]CURRent:DTABLE:SAS:IMP? [ (@<chanlist>)]
```

Description

This command sets the current at the peak power point of the curve when operating in SAS mode. The values are programmed in amperes. If you are programming a follower channel that is paralleled with a master channel, you must set Isc and Imp to zero (0). This configures the channel to act as a follower..

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf	must be less than or equal to Isc	0.8 X maximum rated value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

When a single curve parameter is sent (Isc, Imp, Voc, or Vmp), it is checked along with the three unchanged parameters to determine whether the parameters and resulting curve are within acceptable limits. A curve is then programmed using the new parameter. If a parameter value is outside the limits, an execution error occurs and no new curve is programmed. If all four parameters are programmed individually, there will be four curves produced as each parameter is programmed.

If you are programming an entirely new SAS curve, it is best to send all of the curve parameters on the same program line. When sending all curve parameters on the same program line, the instrument will determine whether all four parameters are within acceptable limits. The curve parameters can be sent in any order. Any parameter not programmed will take on the value of the previously programmed curve.

Return Format

```
<NR1>
```

The query command returns the peak power current value for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the peak power current to 4.5 A on channels 1 and 2.

```
CURR:SAS:IMP 4.5,(@1,2)
```

The following command sends all four of the curve parameters on the same program line on channel 1.

```
CURR:SAS:ISC 5,(@1);IMP 4.5,(@1);:VOLT:SAS:VOC  
100,(@1);VMP 90,(@1)
```

The following query returns the peak power current on channels 1 and 2.

```
CURR:SAS:IMP? (@1,2)
```

Typical Response: 4.5, 4.5

See Also

[\[SOURCE:\]CURRENT::SAS:ISC](#)

[SOURce:]CURRent:SAS:ISC**Syntax**

```
[SOURce:]CURRent:DTABLE:SAS:ISC <current>
[ , (@<chanlist>)]

[SOURce:]CURRent:DTABLE:SAS:ISC? [ (@<chanlist>)]
```

Description

This command sets the short-circuit current when operating in SAS mode. The values are programmed in amperes. If you are programming a follower channel that is paralleled with a master channel, you must set Isc and Imp to zero (0). This configures the channel to act as a follower..

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf	0 to maximum rated value	maximum rated value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

When a single curve parameter is sent (Isc, Imp, Voc, or Vmp), it is checked along with the three unchanged parameters to determine whether the parameters and resulting curve are within acceptable limits. A curve is then programmed using the new parameter. If a parameter value is outside the limits, an execution error occurs and no new curve is programmed. If all four parameters are programmed individually, there will be four curves produced as each parameter is programmed.

If you are programming an entirely new SAS curve, it is best to send all of the curve parameters on the same program line. When sending all curve parameters on the same program line, the instrument will determine whether all four parameters are within acceptable limits. The curve parameters can be sent in any order. Any parameter not programmed will take on the value of the previously programmed curve.

Return Format

```
<NR1>
```

The query command returns the short-circuit current value for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the short-circuit current to 5 A on channels 1 and 2.

```
CURR:SAS:ISC 5,(@1,2)
```

The following command sends all four of the curve parameters on the same program line on channel 1.

```
CURR:SAS:ISC 5,(@1);IMP 4.5,(@1);:VOLT:SAS:VOC  
100,(@1);VMP 90,(@1)
```

The following query returns the short-circuit current on channels 1 and 2.

```
CURR:SAS:ISC? (@1,2)
```

Typical Response: 5, 5

See Also

[\[SOURCE:\]CURRENT::SAS:IMP](#)

[SOURCE:]CURRENT:SAS:MODE**Syntax**

```
[SOURCE:]CURRENT:SAS:MODE <mode> [ ,(@<chanlist>)]
[SOURCE:]CURRENT:SAS:MODE? [(@<chanlist>)]
```

Description

This command selects the source of the curve parameters when operating in SAS mode.

IMMEDIATE In IMMEDIATE mode, the curve parameters are specified by the [\[SOURCE:\]CURRENT:SAS:IMP](#), [\[SOURCE:\]CURRENT:SAS:ISC](#), [\[SOURCE:\]VOLTAGE:SAS:VMP](#), and [\[SOURCE:\]VOLTAGE:SAS:VOC](#) commands.

LIST In List mode the curve parameters are specified by the programmed list..

Parameters

Name	Type	Range of Values	Default Value
<mode>	CPD	IMMEDIATE LIST	IMMEDIATE
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<CRD>

The query command returns the SAS mode for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the SAS mode of channel 2.

```
CURR:SAS:MODE LIST ,(@2)
```

The following query returns the SAS mode of channel 1.

```
CURR:SAS:MODE? (@1)
```

Typical Response: IMM

See Also

[SOURCE:]LIST commands

[SOURce:]CURRent:SLIMit:HIGH

Syntax

```
[SOURce:]CURRent:SLIMit:HIGH <highlimit>
[ , (@<chanlist>)]

[SOURce:]CURRent:SLIMit:HIGH? [ (@<chanlist>)]
```

Description

This command sets the high limit for programming the output current. Units are in amperes. This software limit provides protection against inadvertent over programming of the output current. If a programming command attempts to set the output above the limit, a “value out of range” error is generated and the command is ignored. The instrument does not go into protection mode.

Programming errors generated due to the high limit setting are not reported to the instrument's status subsystem. Your application program should include an error trapping routine to detect programming limit errors..

- FIXed** In Fixed mode, the software limit is applied to the [\[SOURce:\]CURRent:LEVel\[:IMMEDIATE\]](#) command.
- SAS** In Immediate mode, the software limit is applied to the [\[SOURce:\]CURRent:SAS:ISC](#) command.
In List mode, the limit is applied to the lsc list when the list system is activated by the [\[SOURce:\]CURRent:SAS:MODE](#) command.
- TABLE** In Table mode, the combination of each current point and current offset is checked against the limit when the table is activated using [\[SOURce:\]CURRent:TABLE:NAME](#), and when the current offset is changed with [\[SOURce:\]CURRent:TABLE:OFFset](#) while the table is active. The table values are not checked when the table is being programmed.

Parameters

Name	Type	Range of Values	Default Value
<highlimit>	NRf+	0 - maximum MAXimum The maximum value is dependent on the current rating of the output module	N/A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The limit is stored in non-volatile memory. It applies to all channels and all stored states.

The *RST command has no affect on the limit; it does not get restored to any pre-defined value.

Return Format

<NR3>

The query command returns the software current limit. Multiple responses are separated by commas.

Examples

The following command sets a software current limit.

```
CURR:SLIM:HIGH 5
```

The following query returns the software current limit.

```
CURR:SLIM:HIGH?
```

Typical Response: MAX

See Also

[\[SOURCE:VOLTage:SLIMit:HIGH\]](#)

[SOURce:]CURRent:TABLE:NAME**Syntax**

```
[SOURce:]CURRent:TABLE:NAME [<name>]
[ , (@<chanlist>)]

[SOURce:]CURRent:TABLE:NAME? [(@<chanlist>)]
```

Description

This command activates a user-defined table when the unit is operating in Table mode. If the unit is not in Table mode, the specified table becomes the active table when Table mode is invoked.

If you send this command without specifying a table name, it de-selects the active table. This allows the [MEMory:DELeTe:ALL](#) command to delete all tables in both volatile and non-volatile memory. When operating in Table mode, an error is generated if this command is sent without a table name..

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<CRD>
```

The query command returns the active table name.

Examples

The following command activates the table named curve1 on channel 1.

```
CURR:TABLE:NAME curve1 (@1)
```

The following command de-selects the active table

```
CURR:TABLE:NAME (@1)
```

The following query returns the active table name on channel 2.

```
CURR:TABLE:NAME? (@2)
```

Typical Response: curve1

See Also

[MEMory:TABLE:SElect](#)

[MEMory:DELeTe:ALL](#)

[SOURce:]CURRent:TABLE:OFFSet**Syntax**

```
[SOURce:]CURRent:TABLE:OFFSet <offset>
[,(@<chanlist>)]

[SOURce:]CURRent:TABLE:OFFSet? [(@<chanlist>)]
```

Description

This command adds a current offset when operating in Table mode. Units are in amperes. The offset is added to all of the points in the table and each point is range checked. If the resulting table is valid, then it will be activated immediately. Offsets can be positive or negative..

Parameters

Name	Type	Range of Values	Default Value
<offset>	NRf	-Imax to +Imax The maximum value is dependent on the current rating of the output module.	0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NR1>

The query command returns the table offset.

Examples

The following command sets the table offset on channel 1.

```
CURR:TABLE:OFFS 0.5,(@1)
```

The following query returns the table offset of channel 1.

```
CURR:TABLE:OFFS? (@1)
```

Typical Response: 0.5

See Also

[\[SOURce:\]VOLTage:TABLE:OFFSet](#)

[SOURce:]DIGital:INPut:DATA?

Syntax

```
[SOURce:]DIGital:INPut:DATA?
```

Description

This query reads the state of the digital control port.

Return Format

```
<NR1>
```

The query returns the binary-weighted value of the state of pins 1 through 7 in bits 0 through 6 respectively.

Example

The following query returns the state of the digital control port pins.

```
DIG:INP:DATA?
```

Typical Response: +0

See Also

[\[SOURce:\]DIGital:OUTput:DATA](#)

[SOURce:]DIGital:OUTPut:DATA**Syntax**

```
[SOURce:]DIGital:OUTPut:DATA <value>
```

```
[SOURce:]DIGital:OUTPut:DATA?
```

Description

This command sets the output data on the digital control port when that port is configured for Digital I/O operation. The port has seven signal pins and a digital ground pin. In the binary-weighted value that is written to the port, the pins are controlled according to the following bit assignments:

Pin	Bit Number	Decimal Value
1	0	1
2	1	2
3	2	4
4	3	8
5	4	16
6	5	32
7	6	64

Parameters

Name	Type	Range of Values	Default Value
<value>	NRf	0 - 127	0

Return Format

```
<NR1>
```

The query returns the last programmed value of the bits. To read the actual state of the pin, use [\[SOURce:\]DIGital:INPut:DATA?](#)

Example

The following command programs pins 1, 3, and 5 on. (bit 0 = 1; bit 2 = 4; bit 4 = 16) (1 + 4 + 16 = 21)

```
DIG:OUTP:DATA 21
```

The following query returns the state of the digital control port pins.

```
DIG:OUTP:DATA?
```

Typical Response: +21

See Also

[\[SOURce:\]DIGital:INPut:DATA?](#)

[SOURce:]DIGital:PIN<1-7>:FUNction

Syntax

[SOURce:]DIGital:PIN<1-7>:FUNction <function>

[SOURce:]DIGital:PIN<1-7>:FUNction?

Description

This command sets the functions of the digital port pins. The pin functions are saved in non-volatile memory.

DIO	<i>Applies to all pins.</i> The pin is a general-purpose ground-referenced digital input/output. The output can be set with [SOURce:]DIGital:OUTPut:DATA .
DINPut	<i>Applies to all pins.</i> The pin is in digital input-only mode. The digital output data of the corresponding pin is ignored.
TOUTput	<i>Applies only to pins 3-7.</i> The pin is configured as a trigger output. When configured as a trigger output, the pin will only generate output triggers if an output trigger source has been selected from the specified pin. See TRIGger:PIN<3-7>:SOURce
TINPut	<i>Applies only to pins 3-7.</i> The pin is configured as a trigger input. When configured as a trigger input, the pin can be selected as the source for measurement and transient trigger signals. See TRIGger:ACQuire:SOURce and TRIGger:TRANsient:SOURce
FAULt	<i>Applies only to pin 1.</i> Setting FAULt means that pin 1 functions as an isolated fault output. The fault signal is true when any output is in a protected state (from OC, OV, OT, INH, PF, or OS). Note also that Pin 2 serves as the isolated common for pin 1. When pin 1 is set to the FAULt function, the instrument ignores any commands to program pin 2. Queries of pin 2 will return FAULt. If pin 1 is changed from FAULt to another function, pin 2 is set to DINPut.
INHibit	<i>Applies only to pin 3.</i> When pin 3 is configured as an inhibit input, a true signal at the pin will disable all output channels.
ONCOuple	<i>Applies only to pins 4-7.</i> When configured as an On control, the pin will synchronize the output On state between mainframes. Only one pin can be configured as an On control. The pin will function as both an input and an output. The polarity of the pin is fixed and cannot be programmed.
OFFCOuple	<i>Applies only to pins 4-7.</i> When configured as an Off control, the pin will synchronize the output Off state between mainframes. Only one pin can be configured as an Off control. The pin will function as both an input and an output. The polarity of the pin is fixed and cannot be programmed.

Parameters

Name	Type	Range of Values	Default Value
<function>	CPD	DIO DINPut TOUTput TINPut FAULt INHibit ONCOuple OFFCOuple	Required parameter

Return Format

<CRD>

The query command returns DIO, DINP, TOUT, TINP, FAUL, INH, ONC, or OFFC.

Example

The following command sets pin 1 to Fault mode.

```
DIG:PIN1:FUNC FAUL
```

The following query returns the setting of pin 7.

```
DIG:PIN7:FUNC?
```

Typical Response: DIO

See Also

[\[SOURCE:\]DIGital:PIN<1-7>:POLarity](#)

[SOURce:]DIGital:PIN<1-7>:POLarity**Syntax**

```
[SOURce:]DIGital:PIN<1-7>:POLarity <polarity>
```

```
[SOURce:]DIGital:PIN<1-7>:POLarity?
```

Description

This command sets the polarity of the digital port pins. The pin polarities are saved in non-volatile memory.

POSitive Setting a polarity to POSitive means that a logical true signal is a voltage high at the pin. For trigger inputs and outputs, POSitive means a rising edge.

NEGative Setting the polarity NEGative means that a logical true signal is a voltage low at the pin. For trigger inputs and outputs, NEGative means a falling edge.

Parameters

Name	Type	Range of Values	Default Value
<function>	CPD	POSitive NEGative	Required parameter

Return Format

```
<CRD>
```

The query command returns POS or NEG.

Example

The following command sets pin 1 to negative polarity.

```
DIG:PIN1:POL NEGative
```

The following query returns the polarity of pin 7.

```
DIG:PIN7:POL?
```

Typical Response: POS

See Also

[\[SOURce:\]DIGital:PIN<1-7>:FUNCTION](#)

[SOURce:]LIST:COUNT**Syntax**

```
[SOURce:]LIST:COUNT <count>|INFinity [,(@<chanlist>)]
```

```
[SOURce:]LIST:COUNT? [MIN|MAX|INFinity, ]  
[(@<chanlist>)]
```

Description

This command sets the number of times that the list is executed before it is completed. The list count range is 1 through 256.

Parameters

Name	Type	Range of Values	Default Value
<count>	NRf+	1 - 256 MIN MAX INFinity	1
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

Use the INFinity parameter to execute a list continuously.

Use [ABORt:TRANsient](#) to stop the list at any time. When the list is aborted, the output returns to the settings that were in effect before the list started.

Return Format

```
<NR1>
```

The query command returns the list count for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the list count to 10 on channel 1.

```
LIST:COUNT 10,(@1)
```

The following query returns the list count of channel 1.

```
LIST:COUNT? (@1)
```

Typical Response: +10

[SOURce:]LIST:DWELl

Syntax

```
[SOURce:]LIST:DWELl <time>{,<time>} [ ,(@<chanlist>)]
[SOURce:]LIST:DWELl? [(@<chanlist>)]
```

Description

This command specifies the dwell time of each step in a dwell-paced list. A comma-delimited list of up to 512 steps may be programmed. Dwell time is the time that the output will remain at a specific step. Dwell times can be programmed from a minimum value through 33554 seconds. The minimum programmable dwell depends on the selected table size as specified by [\[SOURce:\]CURRent:MODE:DTABLE](#).

- 256-point tables** Minimum programmable dwell time = 0.03 seconds
- 4096-point tables** Minimum programmable dwell time = 0.35 seconds

The dwell time is checked against the minimum programmable value at the time this command is sent. The minimum allowable value that can be supported for a specified table size is checked at the time the [\[INITiate\]:\[IMMediate\]:\[TRANsient\]](#) command is sent.

Parameters

Name	Type	Range of Values	Default Value
<time>	NRf	minimum – 33554 The minimum value is dependent on the table size.	1 step with a value of 0.025
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

At the end of the dwell time, the list advances to the next step.

The order in which the dwell values are entered determines the sequence when the list executes.

To create a valid user-defined List, the IMP, ISC, VMP, VOC, and Dwell lists must either all be the same length, or have a length of 1, which is interpolated as having the same length as the maximum length list..

Return Format

<NR3>

The query command returns the programmed dwell time in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Examples

The following command programs a dwell list. The list contains 5 steps and is on channel 1.

```
LIST:DWEL 0.2,0.8,1.5,0.8,0.2,(@1)
```

The following query returns the dwell list settings of channel 1.

```
LIST:DWEL? (@1)
```

Typical Response:

```
+2.000000E-01,+8.000000E-01,+1.500000E+00,+8.000000E-01,+02.000000E-01
```

See Also

[\[SOURCE: |LIST:DWEL:POINTs?\]](#)

[\[SOURCE: |CURRENT:MODE:DTABLE\]](#)

[SOURce:]LIST:DWEL:POINts?**Syntax**

```
[SOURce:]LIST:DWEL:POINts? [(@<chanlist>)]
```

Description

This query returns the number of points (steps) in the dwell list, not the point values.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The number of points in the dwell list. Multiple responses are separated by commas.

Example

The following query returns the number of points in a dwell list for channel 1.

```
LIST:DWEL:POIN? (@1)
```

Typical Response: +1

See Also

[\[SOURce:\]LIST:DWEL](#)

[SOURce:]LIST:SAS:IMP[:LEVel]**Syntax**

```
[SOURce:]LIST:SAS:IMP[:LEVel] <current>{,<current>}
[ ,(@<chanlist>)]
```

```
[SOURce:] LIST:SAS:IMP[:LEVel]? [(@<chanlist>)]
```

Description

This command specifies the current at the maximum power point for each list step. The values are programmed in amperes. A comma-delimited list of up to 512 (0 - 511) steps may be programmed.

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf	must be less than or equal to Isc	1 step with a value of 6.528 A for E4361A; 3.264 A for E4362A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The order in which the Imp values are entered determines the sequence when the list executes.

To create a valid List, the Imp, Isc, Vmp, Voc, and Dwell lists must either all be the same length, or have a length of 1, which is interpolated as having the same length as the maximum length list.

Return Format

```
<NR1>
```

The query command returns the programmed Imp values. Multiple responses are separated by commas.

Examples

The following command programs an Imp list for channel 1. The list contains 5 steps.

```
step  ISC  IMP  VMP  VOC
0.    1.009 0.918 4.590 5.100
1.    0.506 0.461 4.611 5.120
2.    1.081 0.983 0.828 10.92
3.    0.508 0.462 4.625 5.139
4.    0.511 0.464 4.643 5.159
```

```
LIST:SAS:IMP 0.918, 0.416, 0.983, 0.462, 0.464,(@1)
```

The following query returns the Imp list settings of channel 1.

```
LIST:SAS:IMP? (@1)
```

Typical Response: +9.180000E-01,+4.160000E-01,+9.830000E-01,+4.620000E-01,+4.640000E-01

See Also

[\[SOURCE: LIST:SAS:IMP:POINTs?\]](#)

[SOURce:]LIST:SAS:IMP:POINts?**Syntax**

```
[SOURce:]LIST:SAS:IMP:POINts? [(@<chanlist>)]
```

Description

This query returns the number of points (steps) in the Imp list, not the point values.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The number of points in the Imp list. Multiple responses are separated by commas.

Examples

The following query returns the number of points in an Imp list for channel 1.

```
LIST:SAS:IMP:POIN? (@1)
```

Typical Response: +1

See Also

[\[SOURce:\]LIST:SAS:IMP\[:LEVel\]](#)

[SOURce:]LIST:SAS:ISC[:LEVel]

Syntax

```
[SOURce:]LIST:SAS:ISC[:LEVel] <current>{,<current>}
[ ,(@<chanlist>)]

[SOURce:] LIST:SAS:ISC[:LEVel]? [(@<chanlist>)]
```

Description

This command specifies the short-circuit current for each list step. The values are programmed in amperes. A comma-delimited list of up to 512 (0 - 511) steps may be programmed.

Parameters

Name	Type	Range of Values	Default Value
<current>	NRf	0 to the maximum rated current	1 step with a value of 8.7 A for E4361A; 5.1 A for E4362A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The order in which the Isc values are entered determines the sequence when the list executes.

To create a valid List, the Imp, Isc, Vmp, Voc, and Dwell lists must either all be the same length, or have a length of 1, which is interpolated as having the same length as the maximum length list.

Return Format

<NR1>

The query command returns the programmed Isc values. Multiple responses are separated by commas.

Examples

The following command programs an Imp list for channel 1. The list contains 5 steps.

```
step  ISC  IMP  VMP  VOC
0.    1.009 0.918 4.590 5.100
1.    0.506 0.461 4.611 5.120
2.    1.081 0.983 0.828 10.92
3.    0.508 0.462 4.625 5.139
4.    0.511 0.464 4.643 5.159
```

```
LIST:SAS:ISC 1.009, 0.506, 1.081, 0.508, 0.511,(@1)
```

3 Commands by Subsystem

The following query returns the Isc list settings of channel 1.

```
LIST:SAS:ISC? (@1)
```

Typical Response: +1.009000E+00,+5.060000E-01,+1.081000E+00,+5.080000E-01,+5.110000E-01

See Also

[\[SOURCE: LIST:SAS:ISC:POINTS?\]](#)

[SOURCE:]LIST:SAS:ISC:POINTS?**Syntax**

```
[SOURCE:]LIST:SAS:ISC:POINTS? [(@<chanlist>)]
```

Description

This query returns the number of points (steps) in the Isc list, not the point values.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The number of points in the Isc list. Multiple responses are separated by commas.

Examples

The following query returns the number of points in an Isc list for channel 1.

```
LIST:SAS:ISC:POIN? (@1)
```

Typical Response: +1

See Also

[\[SOURCE:\]LIST:SAS:ISC\[:LEVEL\]](#)

[SOURce:]LIST:SAS:VMP[:LEVel]**Syntax**

```
[SOURce:]LIST:SAS:VMP[:LEVel] <voltage>{,<voltage>}
[ ,(@<chanlist>)]
```

```
[SOURce:] LIST:SAS:VMP[:LEVel]? [(@<chanlist>)]
```

Description

This command specifies the voltage at the maximum power point for each list step. The values are programmed in volts. A comma-delimited list of up to 512 (0 - 511) steps may be programmed.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf	must be less than or equal to Voc	1 step with a value of 60 V for E4361A; 120 V for E4362A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The order in which the Vmp values are entered determines the sequence when the list executes.

To create a valid List, the Imp, Isc, Vmp, Voc, and Dwell lists must either all be the same length, or have a length of 1, which is interpolated as having the same length as the maximum length list.

Return Format

```
<NR1>
```

The query command returns the programmed Vmp values. Multiple responses are separated by commas.

Examples

The following command programs a Vmp list for channel 1. The list contains 5 steps.

```
step  ISC  IMP  VMP  VOC
0.    1.009 0.918 4.590 5.100
1.    0.506 0.461 4.611 5.120
2.    1.081 0.983 0.828 10.92
3.    0.508 0.462 4.625 5.139
4.    0.511 0.464 4.643 5.159
```

```
LIST:SAS:VMP 4.590, 4.611, 0.828, 4.625, 4.643,(@1)
```

The following query returns the Vmp list settings of channel 1.

```
LIST:SAS:VMP? (@1)
```

Typical Response: +4.590000E+00,+4.611000E+00,
+8.280000E-01,+4.625000E+00,+4.643000E+00

See Also

[\[SOURCE: LIST:SAS:VMP:POINTS?\]](#)

[SOURce:]LIST:SAS:VMP:POINts?**Syntax**

```
[SOURce:]LIST:SAS:VMP:POINts? [(@<chanlist>)]
```

Description

This query returns the number of points (steps) in the Vmp list, not the point values.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The number of points in the Vmp list. Multiple responses are separated by commas.

Examples

The following query returns the number of points in a Vmp list for channel 1.

```
LIST:SAS:VMP:POIN? (@1)
```

Typical Response: +1

See Also

[\[SOURce:\]LIST:SAS:VMP\[:LEVel\]](#)

[SOURce:]LIST:SAS:VOC[:LEVel]

Syntax

```
[SOURce:]LIST:SAS:VOC[:LEVel] <voltage>{,<voltage>}
[ ,(@<chanlist>)]

[SOURce:] LIST:SAS:VOC[:LEVel]? [(@<chanlist>)]
```

Description

This command specifies the open-circuit voltage for each list step. The values are programmed in volts. A comma-delimited list of up to 512 (0 - 511) steps may be programmed.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf	0 to maximum rated voltage	1 step with a value of 65 V for E4361A; 130 V for E4362A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The order in which the Voc values are entered determines the sequence when the list executes.

To create a valid List, the Imp, Isc, Vmp, Voc, and Dwell lists must either all be the same length, or have a length of 1, which is interpolated as having the same length as the maximum length list.

Return Format

<NR1>

The query command returns the programmed Voc values. Multiple responses are separated by commas.

Examples

The following command programs a Voc list for channel 1. The list contains 5 steps.

```
step  ISC  IMP  VMP  VOC
0.    1.009 0.918 4.590 5.100
1.    0.506 0.461 4.611 5.120
2.    1.081 0.983 0.828 10.92
3.    0.508 0.462 4.625 5.139
4.    0.511 0.464 4.643 5.159
```

```
LIST:SAS:VOC 5.100, 5.120, 10.92, 5.139, 5.159,(@1)
```

The following query returns the Voc list settings of channel 1.

```
LIST:SAS:VOC? (@1)
```

Typical Response: +5.100000E+00,+5.120000E+00,
+1.092000E+01,+5.139000E+00,+5.159000E+00

See Also

[\[SOURCE: LIST:SAS:VOC:POINTs?\]](#)

[SOURce:]LIST:SAS:VOC:POINts?**Syntax**

```
[SOURce:]LIST:SAS:VOC:POINts? [(@<chanlist>)]
```

Description

This query returns the number of points (steps) in the Voc list, not the point values.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The number of points in the Voc list. Multiple responses are separated by commas.

Examples

The following query returns the number of points in a Voc list for channel 1.

```
LIST:SAS:VOC:POIN? (@1)
```

Typical Response: +1

See Also

[\[SOURce:\]LIST:SAS:VOC\[:LEVel\]](#)

[SOURce:]LIST:STEP**Syntax**

```
[SOURce:]LIST:STEP <step> [ ,(@<chanlist>)]
```

```
[SOURce:]LIST:STEP? [(@<chanlist>)]
```

Description

This command specifies how the list responds to triggers.

ONCE Causes the output to remain at the present step until a trigger advances it to the next step.

AUTO Causes the output to automatically advance to each step, after the receipt of an initial starting trigger. The steps are paced by the dwell list. As each dwell time elapses, the next step is immediately output.

Parameters

Name	Type	Range of Values	Default Value
<step>	CPD	ONCE AUTO	AUTO
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<CRD>
```

The query command returns the list step setting. Multiple responses are separated by commas.

Examples

The following command sets the list step setting of channel 1 to ONCE, causing the list steps to be paced by trigger signals.

```
LIST:STEP ONCE ,(@1)
```

The following query returns the list step setting of channel 1.

```
LIST:STEP? (@1)
```

Typical Response: ONCE

See Also

[\[SOURce:\]LIST:TERMinate:LAST](#)

[SOURce:]LIST:STEP:ACTive?**Syntax**

```
[SOURce:]LIST:STEP:ACTive? [(@<chanlist>)]
```

Description

This query returns the presently active list step. Note that for lists with extremely short dwell times, the list may have progressed to the next step by the time the data has been returned..

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR1>
```

The presently active list step. Note that the step number is zero-based. Returns -1 if a list is not running.

Examples

The following query returns the active list step of channel 1.

```
LIST:STEP:ACT? (@1)
```

Typical Response: 4

See Also

[\[SOURce:\]LIST:STEP](#)

[SOURce:]LIST:TERMinate:LAST**Syntax**

```
[SOURce:]LIST:TERMinate:LAST <Bool> [ ,(@<chanlist>)]
[SOURce:]LIST:TERMinate:LAST? [(@<chanlist>)]
```

Description

This command determines the I-V characteristics when the list terminates. The state is either ON (1) or OFF (0).

When ON, the output voltage or current remains at the value of the last list step. The last list step becomes the IMMEDIATE value when the list completes.

When OFF, and also when the list is aborted, the I-V curve characteristics return to the IMMEDIATE values.

Parameters

Name	Type	Range of Values	Default Value
<Bool>	Boolean	OFF 0 ON 1	OFF
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NR1>

The query command returns 0 if the output returns to the settings that were in effect before the list started, and 1 if the output voltage or current remains at the value of the last list step.

Examples

The following command sets the list to terminate with the I-V curve remaining at the value of the last list step on channel 2.

```
LIST:TERM:LAST ON , (@2)
```

The following query returns the list terminate setting of channel 1.

```
LIST:TERM:LAST? (@1)
```

Typical Response: 0

See Also

[\[SOURce:\]LIST:STEP](#)

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]**Syntax**

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]
<voltage> [ , (@<chanlist>)]
```

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?
[MIN|MAX, ] [ (@<chanlist>)]
```

Description

This command sets the voltage level of the specified output channel. Units are in volts.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf+	0 - maximum MIN MAX The maximum value is dependent on the voltage rating of the output module.	0 V
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

This command applies in fixed MODE only.

Return Format

```
<NR3>
```

The query command returns the programmed voltage level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the output voltage to 50 V on channels 1 and 2.

```
VOLT 50 , (@1 , 2)
```

The following query returns the programmed voltage level on channels 1 and 2.

```
VOLT? (@1 , 2)
```

Typical Response: +5.000000E+01 , +5.000000E+01

See Also

[\[SOURce:\]CURRENT\[:LEVel\]\[:IMMediate\]](#)

[SOURce:]VOLTage:DTABle:SASimulator[:IMMediate]?**Syntax**

```
[SOURce:]VOLTage:DTABle:SASimulator[:IMMediate]?
[(@<chanlist>)]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings. The number of voltage points returned is based on the [\[SOURce:\]CURREnt:MODE:DTABle](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table settings of channel 2.

```
VOLT:DTAB:SAS? (@2)
```

Typical Response: 49.8, 49.9, 50.0, 50.1, 50.2, 50.3, ...

See Also

[\[SOURce:\]CURREnt:MODE:DTABle](#)

[SOURce:]VOLTage:DTABle:SASimulator[:IMMediate]:VMP?**Syntax**

```
[SOURce:]VOLTage:DTABle:SASimulator[:IMMediate]:VMP?
[(@<chanlist>)]
```

Description

This query returns the calculated Vmp (maximum power voltage) of the internal DAC table used for the SAS settings. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Vmp for channel 2 at a 256 point resolution with an SAS setting of::

```
Vmp = 55 V
Imp = 3.5 A
Voc = 60 V
Isc = 4 A
```

```
VOLT:DTAB:SAS:VMP? (@2)
```

Typical Response: +5.389988E+01

See Also

[\[SOURce:\]VOLTage::DTABle:SASimulator:VOC?](#)

[SOURce:]VOLTage:DTABLE:SASimulator[:IMMediate]:VOC?**Syntax**

```
[SOURce:]VOLTage:DTABLE:SASimulator[:IMMediate]:VOC?
[(@<chanlist>)]
```

Description

This query returns the calculated Voc (open-circuit voltage) of the internal DAC table used for the SAS settings. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Voc for channel 2 at a 256 point resolution with an SAS setting of:

Vmp = 55 V

Imp = 3.5 A

Voc = 60 V

Isc = 4 A

```
VOLT:DTAB:SAS:VOC? (@2)
```

Typical Response: +6.00000E+00

See Also

[\[SOURce:\]VOLTage:DTABLE:SASimulator:VMP?](#)

[SOURCE:]VOLTage:DTABLE:SASimulator:LIST?**Syntax**

```
[SOURCE:]VOLTage:DTABLE:SASimulator:LIST? <step>
[ , (@<chanlist>)]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings of the specified SAS list step. The number of voltage points returned is based on the [\[SOURCE:\]CURRENT:MODE:DTABLE](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table of the 4th step for channel 1 at a 256 point resolution.

```
VOLT:DTAB:SAS:LIST? 3, (@1)
```

Typical Response: 49.8,49.9,50.0,50.1,50.2,50.3, ...

See Also

[\[SOURCE:\]CURRENT:MODE:DTABLE](#)

[SOURCE:]VOLTage:DTABLE:SASimulator:LIST:VMP?**Syntax**

```
[SOURCE:]VOLTage:DTABLE:SASimulator:LIST:VMP? <step>
[ , (@<chanlist>)]
```

Description

This query returns the calculated Vmp (maximum power voltage) value of the internal DAC table used for the specified SAS list step. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the Vmp value of the 4th step for channel 1 at a 256 point resolution with an SAS setting of

Vmp = 40 V

Imp = 4.08 A

Voc = 60 V

Isc = 5.1 A

```
VOLT:DTAB:SAS:LIST:VMP? 3, (@1)
```

Typical Response: +4.298372E+01

See Also

[\[SOURCE:\]VOLTage::DTABLE:SASimulator:LIST:VOC?](#)

[SOURce:]VOLTage:DTABle:SASimulator:LIST:VOC?**Syntax**

```
[SOURce:]VOLTage:DTABle:SASimulator:LIST:VOC? <step>
[ , (@<chanlist>)]
```

Description

This query returns the calculated Voc (open-circuit voltage) value of the internal DAC table used for the specified SAS list step. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<step>	NR1	0-511 (The list step number is zero-based; the first step is step zero.)	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NRf>

Multiple channel responses are separated by commas.

Examples

The following query returns the Voc value of the 4th step for channel 1 at a 256 point resolution with an SAS setting of

Vmp = 40 V

Imp = 4.08 A

Voc = 60 V

Isc = 5.1 A

```
VOLT:DTAB:SAS:LIST:VOC? 3, (@1)
```

Typical Response: +6.000000E+01

See Also

[\[SOURce:\]VOLTage::DTABle:SASimulator:LIST:VMP?](#)

[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]?**Syntax**

```
[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]? <name>
[,@<chanlist>]
```

Description

This query returns the DAC table that was calculated based on the SAS parameter settings in the specified table. The number of voltage points returned is based on the [\[SOURce:\]CURRent:MODE:DTABle](#) setting (either 256 points or 4096 points).

NOTE

This command is only available with firmware revision A.01.00 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

The query returns the values of the DAC table. Multiple responses are separated by commas. If more than one channel has been queried, all the DAC values for the first channel are returned, followed immediately by the DAC values of the next channel.

Examples

The following query returns the DAC table values for the table named curve1.

```
VOLT:DTAB:TABL? curve1,@1
```

Typical Response: 49.8,49.9,50.0,50.1,50.2,50.3, ...

See Also

[\[SOURce:\]CURRent:MODE:DTABle](#)

[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]:VMP?**Syntax**

```
[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]:VMP?
<name> [,(@<chanlist>)]
```

Description

This query returns the calculated Vmp (maximum power voltage) value of the internal DAC table for the specified table. Note that the resolution of the DAC table can affect the accuracy of the calculation..

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Vmp for channel 1 at a 256 point resolution for the table named sample2:

```
Voltage Current
1 1.5
4 1.45
8 1.4
12 1.35
16 1.3
20 1.2
24 1
```

```
VOLT:DTAB:TABL:VMP? sample2,(@1)
```

Typical Response: +2.199414E+01

See Also

[\[SOURce:\]VOLTage::DTABle:TABLE:VOC?](#)

[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]:VOC?**Syntax**

```
[SOURce:]VOLTage:DTABle:TABLE[:IMMediate]:VOC?
<name> [,(@<chanlist>)]
```

Description

This query returns the calculated Voc (open-circuit voltage) value of the internal DAC table for the specified table. Note that the resolution of the DAC table can affect the accuracy of the calculation.

NOTE

This command is only available with firmware revision A.01.05 or later.

Parameters

Name	Type	Range of Values	Default Value
<name>	CPD	The table name.	Required parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NRf>
```

Multiple channel responses are separated by commas.

Examples

The following query returns the calculated Voc for channel 1 at a 256 point resolution for the table named sample2:

```
Voltage Current
1      1.5
4      1.45
8      1.4
12     1.35
16     1.3
20     1.2
24     1
```

```
VOLT:DTAB:TABL:VOC? sample2,(@1)
```

Typical Response: +4.400000E+01

See Also

[\[SOURce:\]VOLTage::DTABle:TABLE:VMP?](#)

[SOURce:]VOLTage:PROTection[:LEVel]**Syntax**

```
[SOURce:]VOLTage:PROTection[:LEVel] <voltage>
[ , (@<chanlist>)]

[SOURce:]VOLTage:PROTection[:LEVel]? [MIN|MAX, ]
[ (@<chanlist>)]
```

Description

This command sets the over-voltage protection (OVP) level of the output channel. The values are programmed in volts. If the output voltage exceeds the OVP level, the output is disabled and the Questionable Condition status register OV bit is set.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf+	0 - maximum MIN MAX The maximum value is dependent on the voltage rating of the output module.	MAX
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

An over-voltage condition can be cleared with the OUTput:PROTection:CLEar command after the condition that caused the OVP trip is removed.

Return Format

<NR3>

The query command returns the programmed protection level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the voltage protection level to 45 V on channel 1.

```
VOLT:PROT 45, (@1)
```

The following query returns the programmed voltage protection level on channel 1.

```
VOLT:PROT? (@1)
```

Typical Response: +4.500000E+01

See Also

[OUTPut:PROTection:CLEar](#)

[SOURce:]VOLTage:SAS:VMP**Syntax**

```
[SOURce:]VOLTage:DTABLE:SAS:VMP <voltage>
[ , (@<chanlist>)]

[SOURce:]VOLTage:DTABLE:SAS:VMP? [ (@<chanlist>)]
```

Description

This command sets the voltage at the peak power point of the curve when operating in SAS mode. The values are programmed in volts. If you are programming a follower channel that is paralleled with a master channel, you must program the follower channels with the same Voc and Vmp values.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf	must be less than or equal to Voc	0.8 X maximum rated value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

When a single curve parameter is sent (Isc, Imp, Voc, or Vmp), it is checked along with the three unchanged parameters to determine whether the parameters and resulting curve are within acceptable limits. A curve is then programmed using the new parameter. If a parameter value is outside the limits, an execution error occurs and no new curve is programmed. If all four parameters are programmed individually, there will be four curves produced as each parameter is programmed.

If you are programming an entirely new SAS curve, it is best to send all of the curve parameters on the same program line. When sending all curve parameters on the same program line, the instrument will determine whether all four parameters are within acceptable limits. The curve parameters can be sent in any order. Any parameter not programmed will take on the value of the previously programmed curve.

Return Format

```
<NR1>
```

The query command returns the peak power voltage value for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the peak power voltage to 90 V on channels 1 and 2.

```
VOLT:SAS:VMP 90,(@1,2)
```

The following command sends all four of the curve parameters on the same program line on channel 1.

```
VOLT:SAS:VOC 100,(@1);VMP 90,(@1);:CURR:SAS:ISC  
5,(@1);IMP 4.5,(@1)
```

The following query returns the peak power voltage on channels 1 and 2.

```
VOLT:SAS:VMP? (@1,2)
```

Typical Response: 90, 90

See Also

[\[SOURCE:VOLTage:SAS:VOC](#)

[SOURce:]VOLTage:SAS:VOC**Syntax**

```
[SOURce:]VOLTage:DTABLE:SAS:VOC <voltage>
[ , (@<chanlist>)]

[SOURce:]VOLTage:DTABLE:SAS:VOC? [ (@<chanlist>)]
```

Description

This command sets the open-circuit voltage when operating in SAS mode. The values are programmed in volts. If you are programming a follower channel that is paralleled with a master channel, you must program the follower channels with the same Voc and Vmp values.

Parameters

Name	Type	Range of Values	Default Value
<voltage>	NRf	0 to maximum rated value	maximum rated value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

When a single curve parameter is sent (Isc, Imp, Voc, or Vmp), it is checked along with the three unchanged parameters to determine whether the parameters and resulting curve are within acceptable limits. A curve is then programmed using the new parameter. If a parameter value is outside the limits, an execution error occurs and no new curve is programmed. If all four parameters are programmed individually, there will be four curves produced as each parameter is programmed.

If you are programming an entirely new SAS curve, it is best to send all of the curve parameters on the same program line. When sending all curve parameters on the same program line, the instrument will determine whether all four parameters are within acceptable limits. The curve parameters can be sent in any order. Any parameter not programmed will take on the value of the previously programmed curve.

Return Format

```
<NR1>
```

The query command returns the open-circuit voltage value for each channel specified. Multiple responses are separated by commas.

Examples

The following command sets the open-circuit voltage to 100 V on

channels 1 and 2.

```
VOLT:SAS:VOC 100,(@1,2)
```

The following command sends all four of the curve parameters on the same program line on channel 1.

```
VOLT:SAS:VOC 100,(@1);VMP 90,(@1);CURR:SAS:ISC  
5,(@1);IMP 4.5,(@1)
```

The following query returns the open-circuit voltage on channels 1 and 2.

```
VOLT:SAS:VOC? (@1,2)
```

Typical Response: 100, 100

See Also

[\[SOURCE:VOLTage::SAS:VMP](#)

[SOURce:]VOLTage:SLIMit:HIGH**Syntax**

```
[SOURce:]VOLTage:SLIMit:HIGH <highlimit>
[ , (@<chanlist>)]

[SOURce:]VOLTage:SLIMit:HIGH? [ (@<chanlist>)]
```

Description

This command sets the high limit for programming the output voltage. Units are in volts. This software limit provides protection against inadvertent over programming of the output voltage. If a programming command attempts to set the output above the limit, a “value out of range” error is generated and the command is ignored. The instrument does not go into protection mode.

Programming errors generated due to the high limit setting are not reported to the instrument's status subsystem. Your application program should include an error trapping routine to detect programming limit errors.

- FIXed** In Fixed mode, the software limit is applied to the [\[SOURce:\]VOLTage\[:LEVel\]\[:IMMEDIATE\]](#) command.
- SAS** In Immediate mode, the software limit is applied to the [\[SOURce:\]VOLTage:SAS:VOC](#) command.
In List mode, the limit is applied to the Isc list when the list system is activated by the [\[SOURce:\]CURRENT:SAS:MODE](#) command.
- TABLE** In Table mode, the combination of each current point and current offset is checked against the limit when the table is activated using [\[SOURce:\]CURRENT:TABLE:NAME](#), and when the current offset is changed with [\[SOURce:\]VOLTage:TABLE:OFFset](#) while the table is active. The table values are not checked when the table is being programmed.

Parameters

Name	Type	Range of Values	Default Value
<highlimit>	NRf+	0 - maximum MAXimum The maximum value is dependent on the voltage rating of the output module	N/A
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The limit is stored in non-volatile memory. It applies to all channels

and all stored states.

The *RST command has no affect on the limit; it does not get restored to any pre-defined value.

Return Format

<NR3>

The query command returns the software voltage limit. Multiple responses are separated by commas.

Examples

The following command sets a software voltage limit.

```
VOLT:SLIM:HIGH 50
```

The following query returns the software voltage limit.

```
VOLT:SLIM:HIGH?
```

Typical Response: MAX

See Also

[\[SOURCE:\]CURRENT:SLIMIT:HIGH](#)

[SOURce:]VOLTage:TABLE:OFFSet**Syntax**

```
[SOURce:]VOLTage:TABLE:OFFSet <offset>
[ , (@<chanlist>)]

[SOURce:]VOLTage:TABLE:OFFSet? [ (@<chanlist>)]
```

Description

This command adds a voltage offset when operating in Table mode. Units are in volts. The offset is added to all of the points in the table and each point is range checked. If the resulting table is valid, then it will be activated immediately. Offsets can be positive or negative..

Parameters

Name	Type	Range of Values	Default Value
<offset>	NRf	-Vmax to +Vmax The maximum value is dependent on the voltage rating of the output module.	0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

<NR1>

The query command returns the table offset.

Examples

The following command sets the table offset on channel 2.

```
VOLT:TABLE:OFFS 5, (@2)
```

The following query returns the table offset of channel 2.

```
VOLT:TABLE:OFFS? (@2)
```

Typical Response: 0

See Also

[\[SOURce:\]CURRENT:TABLE:OFFSet](#)

STATus Subsystem Introduction

Status register programming lets you determine the operating condition of the instrument at any time. The instrument has three groups of status registers; Operation, Questionable, and Standard Event. The Operation and Questionable status groups each consist of the Condition, Enable, and Event registers as well as NTR and PTR filters.

The Status subsystem is also programmed using Common commands. Common commands control additional status functions such as the Service Request Enable and the Status Byte registers.

Refer to the [Status System diagram](#).

SCPI Command	Description
STATus	
:OPERation	
[:EVENT]? (@chanlist)	Returns the value of the operation event register
:CONDition? (@chanlist)	Returns the value of the operation condition register
:ENABle <NRf>, (@chanlist)	Enables specific bits in the Event register
:NTRansition <NRf>, (@chanlist)	Sets the Negative transition filter
:PTRansition <NRf>, (@chanlist)	Sets the Positive transition filter
:PRESet	Presets all enable and transition registers to power-on
:QUEStionable	
[:EVENT]? (@chanlist)	Returns the value of the questionable event register
:CONDition? (@chanlist)	Returns the value of the questionable condition register
:ENABle <NRf>, (@chanlist)	Enables specific bits in the Event register
:NTRansition <NRf>, (@chanlist)	Sets the Negative transition filter
:PTRansition <NRf>, (@chanlist)	Sets the Positive transition filter

Operation Status Group

These registers record signals that occur during normal operation. The group consists of a Condition, PTR/NTR, Event, and Enable register. The outputs of the Operation Status register group are logically-ORed into the OPERation summary bit (7) of the Status Byte register.

Questionable Status Group

These registers record signals that indicate abnormal operation. The group consists of the same register types as the Operation Status group. The outputs of the Questionable Status group are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register.

Standard Event Status Group

These registers are programmed by Common commands. The Standard Event event register latches events relating to communication status. 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.

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*.

MSS and RQS Bits

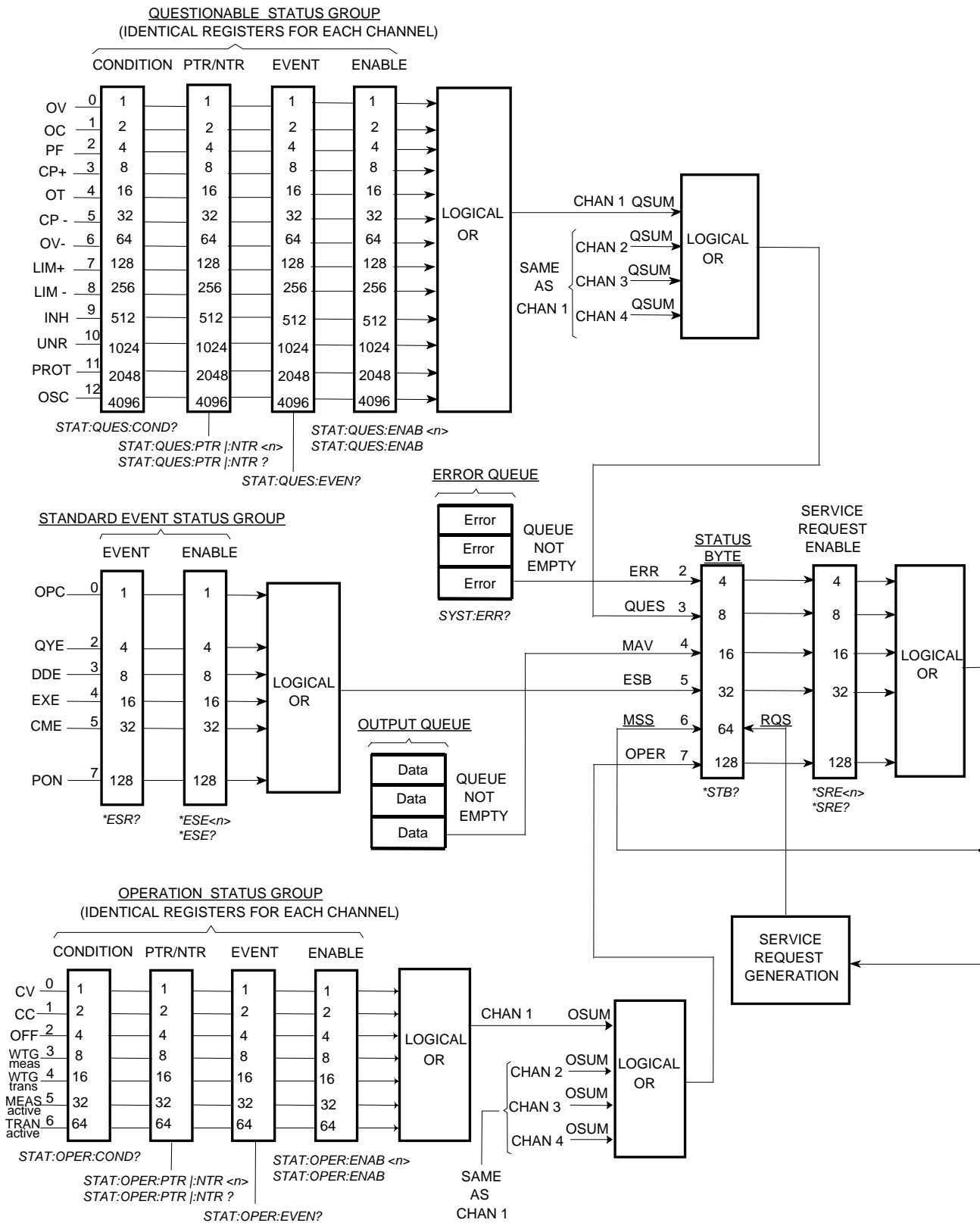
MSS is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the instrument has one or more reasons for requesting service. *STB? reads the MSS in bit position 6 of the response but does not clear any of the bits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

The RQS bit is a latched version of the MSS bit. Whenever the instrument 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.

MAV Bit and Output Queue

The Output Queue is a first-in, first-out (FIFO) data register that stores instrument-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.

Status System Diagram



STATus:OPERation[:EVENT]?

Syntax

STATus:OPERation[:EVENT]? [(@<chanlist>)]

Description

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

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit Number		Decimal Value	Definition
0, 1	Not Used	Not Used	0 is returned.
2	OFF	4	The output is programmed off.
3	WTG-meas	8	The output is waiting for a measurement trigger.
4	Not Used	Not Used	0 is returned.
5	WTG-tran	32	The output is waiting for a transient trigger. When Auto paced, the bit is set before and cleared after the first trigger. When Trigger paced, the bit is set and cleared on every trigger.
6	WTG-Dlog	64	The output is waiting for a datalog trigger
7	Busy-Dlog	128	The output is busy logging data to the datalog buffer.
8	CV	256	The output is in constant voltage.
9	Not Used	Not Used	0 is returned.
10	CC	1024	The output is in constant current.
11-15	Not Used	Not Used	0 is returned.

Once a bit is set, it remains set until cleared by reading the event register or the [*CLS](#) (clear status) command.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 2 (decimal value = 4) and bit 3 (decimal value = 8) are set, this command will return 12.

Example

The following command reads the Operation Status Event register.

```
STAT:OPER? (@1)
```

Typical Response: +12

See Also

[STATus:OPERation:ENABle](#)

[*CLS](#)

STATus:OPERation:CONDition?

Syntax

STATus:OPERation:CONDition? [(@<chanlist>)]

Description

This query returns the value of the Operation Status Condition register. The Condition register is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Operation Condition register does not clear it.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit Number		Decimal Value	Definition
0, 1	Not Used	Not Used	0 is returned.
2	OFF	4	The output is programmed off.
3	WTG-meas	8	The output is waiting for a measurement trigger.
4	Not Used	Not Used	0 is returned.
5	WTG-tran	32	The output is waiting for a transient trigger. When Auto paced, the bit is set before and cleared after the first trigger. When Trigger paced, the bit is set and cleared on every trigger.
6	WTG-Dlog	64	The output is waiting for a datalog trigger
7	Busy-Dlog	128	The output is busy logging data to the datalog buffer.
8	CV	256	The output is in constant voltage.
9	Not Used	Not Used	0 is returned.
10	CC	1024	The output is in constant current.
11-15	Not Used	Not Used	0 is returned.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 2 (decimal value = 4) and bit 3 (decimal value = 8) are set, this command will return 12.

Example

The following command reads the Operation Status Condition register.

```
STAT:OPER:COND? (@1)
```

Typical Response: +12

STATus:OPERation:ENABle

Syntax

STATus:OPERation:ENABle <value> [,(@<chanlist>)]

STATus:OPERation:ENABle? [(@<chanlist>)]

Description

This command and its query set and read the value of the Operation Status Enable register. The Enable 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. This bit (bit 7) is the logical OR of all the Operational Event register bits that are enabled by the Operation Status Enable register.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 3 (decimal value = 8), the corresponding decimal value would be 12 (4 + 8).

Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit Number	Decimal Value	Definition
0, 1 Not Used	Not Used	0 is returned.
2 OFF	4	The output is programmed off.
3 WTG-meas	8	The output is waiting for a measurement trigger.
4 Not Used	Not Used	0 is returned.
5 WTG-tran	32	The output is waiting for a transient trigger. When Auto paced, the bit is set before and cleared after the first trigger. When Trigger paced, the bit is set and cleared on every trigger.

3 Commands by Subsystem

Bit Number		Decimal Value	Definition
6	WTG-Dlog	64	The output is waiting for a datalog trigger
7	Busy-Dlog	128	The output is busy logging data to the datalog buffer.
8	CV	256	The output is in constant voltage.
9	Not Used	Not Used	0 is returned.
10	CC	1024	The output is in constant current.
11-15	Not Used	Not Used	0 is returned.

The [STATus:PRESet](#) command will clear all bits in the enable register.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 2 (decimal value = 4) and bit 3 (decimal value = 8) are set, this command will return 12.

Example

The following command enables bits 2 and 3 (decimal value = 12) in the enable register.

```
STAT:OPER:ENAB 12, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:OPER:ENAB? (@1)
```

Typical Response: +12

See Also

[STATus:OPERation\[:EVENT\]?](#)

[STATus:PRESet](#)

STATus:OPERation:NTRansition

Syntax

```
STATus:OPERation:NTRansition <value> [ ,(@<chanlist>)]
STATus:OPERation:NTRansition? [(@<chanlist>)]
```

Description

These command sets and reads the value of the Operation NTR (Negative-Transition) register. This register serves as a polarity filter between the Operation Condition and Operation Event registers. 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.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 3 (decimal value = 8), the corresponding decimal value would be 12 (4 + 8).

Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit Number	Decimal Value	Definition
0, 1 Not Used	Not Used	0 is returned.
2 OFF	4	The output is programmed off.
3 WTG-meas	8	The output is waiting for a measurement trigger.
4 Not Used	Not Used	0 is returned.
5 WTG-tran	32	The output is waiting for a transient trigger. When Auto paced, the bit is set before and cleared after the first trigger. When Trigger paced, the bit is set and cleared on every trigger.

3 Commands by Subsystem

Bit Number		Decimal Value	Definition
6	WTG-Dlog	64	The output is waiting for a datalog trigger
7	Busy-Dlog	128	The output is busy logging data to the datalog buffer.
8	CV	256	The output is in constant voltage.
9	Not Used	Not Used	0 is returned.
10	CC	1024	The output is in constant current.
11-15	Not Used	Not Used	0 is returned.

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.

The [STATus:PRESet](#) command will set all bits in the NTR register to 0.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register.

Examples

The following command enables bits 2 and 3 (decimal value = 12) in the NTR register.

```
STAT:OPER:NTR 12, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:OPER:NTR? (@1)
```

Typical Response: +12

See Also

[STATus:OPERation:PTRansition](#)

[STATus:PRESet](#)

STATus:OPERation:PTRansition

Syntax

```
STATus:OPERation:PTRansition <value> [ ,(@<chanlist>)]
STATus:OPERation:PTRansition? [(@<chanlist>)]
```

Description

These command sets and reads the value of the Operation PTR (Positive-Transition) register. This register serves as a polarity filter between the Operation Condition and Operation Event registers. When a bit in 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.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 31
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 3 (decimal value = 8), the corresponding decimal value would be 12 (4 + 8).

Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit Number	Decimal Value	Definition
0, 1 Not Used	Not Used	0 is returned.
2 OFF	4	The output is programmed off.
3 WTG-meas	8	The output is waiting for a measurement trigger.
4 Not Used	Not Used	0 is returned.
5 WTG-tran	32	The output is waiting for a transient trigger. When Auto paced, the bit is set before and cleared after the first trigger. When Trigger paced, the bit is set and cleared on every trigger.

3 Commands by Subsystem

Bit Number		Decimal Value	Definition
6	WTG-Dlog	64	The output is waiting for a datalog trigger
7	Busy-Dlog	128	The output is busy logging data to the datalog buffer.
8	CV	256	The output is in constant voltage.
9	Not Used	Not Used	0 is returned.
10	CC	1024	The output is in constant current.
11-15	Not Used	Not Used	0 is returned.

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.

The [STATus:PRESet](#) command will set all bits in the PTR register to 1.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register.

Examples

The following command enables bits 2 and 3 (decimal value = 12) in the PTR register.

```
STAT:OPER:PTR 12, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:OPER:PTR? (@1)
```

Typical Response: +12

See Also

[STATus:OPERation:NTRansition](#)

[STATus:PRESet](#)

STATus:PRESet**Syntax**

```
STATus:PRESet
```

Description

This command sets all defined bits in the status system's PTR registers and clears all bits in the NTR and Enable registers.

Operation Register	Questionable Register	Preset setting
STAT:OPER:ENAB	STAT:QUES:ENAB	0 - all bits disabled
STAT:OPER:NTR	STAT:QUES:NTR	0 - all bits disabled
STAT:OPER:PTR		31 all defined bits enabled
	STAT:QUES:PTR	3647 all defined bits enabled

Example

The following command presets the Operation and Questionable status registers.

```
STAT:PRES
```

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

```
STATus:QUEStionable[:EVENT]? [(@<chanlist>)]
```

Description

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

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit Number	Decimal Value	Definition
0 OV	1	The output is disabled by the over-voltage protection.
1 OC	2	The output is disabled by the over-current protection.
2 PF	4	The output is disabled by the power-fail - which may be caused by a low-line or brownout condition on the AC line.
3 Not Used	Not Used	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5-7 Not Used	Not Used	0 is returned.
8 OS	256	The over-switching protection has tripped. Over-switching protection shuts down the output if the switching frequency exceeds the 50 kHz limit
9 INH	512	The output is inhibited by an external signal.
10 UNR	1024	The output is unregulated.
11 PROT	2048	The output has been disabled because it is coupled to a protection condition that occurred on another channel.
12-15 Not Used	Not Used	0 is returned.

Once a bit is set, it remains set until cleared by reading the event register or the [*CLS](#) (clear status) command.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 10 is set, this command will return 1024.

Example

The following command reads the Questionable Status Event register.

```
STAT:QUES?
```

Typical Response: +1024

See Also

[STATus:QUEStionable:ENABle?](#)

[*CLS](#)

STATus:QUEStionable:CONDition?**Syntax**

```
STATus:QUEStionable:CONDition? [(@<chanlist>)]
```

Description

This query returns the value of the Questionable Status Condition register. The Condition register is a read-only register, which holds the live (unlatched) questionable status of the instrument. Reading the Questionable Condition register does not clear it.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit Number	Decimal Value	Definition
0 OV	1	The output is disabled by the over-voltage protection.
1 OC	2	The output is disabled by the over-current protection.
2 PF	4	The output is disabled by the power-fail - which may be caused by a low-line or brownout condition on the AC line.
3 Not Used	Not Used	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5-7 Not Used	Not Used	0 is returned.
8 OS	256	The over-switching protection has tripped. Over-switching protection shuts down the output if the switching frequency exceeds the 50 kHz limit
9 INH	512	The output is inhibited by an external signal.
10 UNR	1024	The output is unregulated.
11 PROT	2048	The output has been disabled because it is coupled to a protection condition that occurred on another channel.
12-15 Not Used	Not Used	0 is returned.

Return Format

```
<NR1>
```


The value returned is the binary-weighted sum of all bits set in the register. For example, if bit 10 is set, this command will return 1024.

Example

The following command reads the Questionable Status Condition register.

```
STAT:QUES:COND? (@1)
```

Typical Response: +1024

STATus:QUEStionable:ENABle

Syntax

```
STATus:QUEStionable:ENABle <value> [ ,(@<chanlist>)]
```

```
STATus:QUEStionable:ENABle? [(@<chanlist>)]
```

Description

This command and its query set and read the value of the Questionable Status Enable register. The Enable register is a mask for enabling specific bits from the Questionable Event register to set the questionable summary bit (QUES) 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.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 4 (decimal value = 16), the corresponding decimal value would be 20 (4 + 16).

Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit Number	Decimal Value	Definition
0 OV	1	The output is disabled by the over-voltage protection.
1 OC	2	The output is disabled by the over-current protection.
2 PF	4	The output is disabled by the power-fail - which may be caused by a low-line or brownout condition on the AC line.
3 Not Used	Not Used	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5-7 Not Used	Not Used	0 is returned.
8 OS	256	The over-switching protection has tripped. Over-switching protection shuts down the output if the switching frequency exceeds the 50 kHz limit

Bit Number	Decimal Value	Definition
9 INH	512	The output is inhibited by an external signal.
10 UNR	1024	The output is unregulated.
11 PROT	2048	The output has been disabled because it is coupled to a protection condition that occurred on another channel.
12-15 Not Used	Not Used	0 is returned.

The [STATus:PRESet](#) command will clear all bits in the enable register.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register.

Examples

The following command enables bits 0 and 1 (decimal value = 3) in the enable register.

```
STAT:QUES:ENAB 3, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:QUES:ENAB? (@1)
```

Typical Response: +3

See Also

[STATus:QUESTionable\[:EVENT\]?](#)

[STATus:PRESet](#)

STATus:QUEStionable:NTRansition

Syntax

```
STATus:QUEStionable:NTRansition <value>
[ , (@<chanlist>)]

STATus:QUEStionable:NTRansition? [ (@<chanlist>)]
```

Description

These command sets and reads the value of the Questionable NTR (Negative-Transition) register. This register serves as a polarity filter between the Questionable Condition and Questionable Event registers. 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.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 4 (decimal value = 16), the corresponding decimal value would be 20 (4 + 16).

Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit Number	Decimal Value	Definition
0 OV	1	The output is disabled by the over-voltage protection.
1 OC	2	The output is disabled by the over-current protection.
2 PF	4	The output is disabled by the power-fail - which may be caused by a low-line or brownout condition on the AC line.
3 Not Used	Not Used	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5-7 Not Used	Not Used	0 is returned.

Bit Number	Decimal Value	Definition
8 OS	256	The over-switching protection has tripped. Over-switching protection shuts down the output if the switching frequency exceeds the 50 kHz limit
9 INH	512	The output is inhibited by an external signal.
10 UNR	1024	The output is unregulated.
11 PROT	2048	The output has been disabled because it is coupled to a protection condition that occurred on another channel.
12-15 Not Used	Not Used	0 is returned.

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.

The [STATus:PRESet](#) command will set all bits in the NTR register to 0.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register.

Examples

The following command enables bits 0 and 1 (decimal value = 3) in the NTR register.

```
STAT:QUES:NTR 3, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:QUES:NTR? (@1)
```

Typical Response: +3

See Also

[STATus:QUEStionable:PTRansition](#)

[STATus:PRESet](#)

STATus:QUEStionable:PTRansition

Syntax

```
STATus:QUEStionable:PTRansition <value>
[ , (@<chanlist>)]
```

```
STATus:QUEStionable:PTRansition? [ (@<chanlist>)]
```

Description

These command sets and reads the value of the Questionable PTR (Positive-Transition) register. This register serves as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in 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.

NOTE

For more information on the SCPI Status System, see the [Status System diagram](#)

Parameters

Name	Type	Range of Values	Default Value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (see table below).	Preset = 3647
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

For example, to enable bit 2 (decimal value = 4) and bit 4 (decimal value = 16), the corresponding decimal value would be 20 (4 + 16).

Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit Number	Decimal Value	Definition
0 OV	1	The output is disabled by the over-voltage protection.
1 OC	2	The output is disabled by the over-current protection.
2 PF	4	The output is disabled by the power-fail - which may be caused by a low-line or brownout condition on the AC line.
3 Not Used	Not Used	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5-7 Not Used	Not Used	0 is returned.

Bit Number	Decimal Value	Definition
8 OS	256	The over-switching protection has tripped. Over-switching protection shuts down the output if the switching frequency exceeds the 50 kHz limit
9 INH	512	The output is inhibited by an external signal.
10 UNR	1024	The output is unregulated.
11 PROT	2048	The output has been disabled because it is coupled to a protection condition that occurred on another channel.
12-15 Not Used	Not Used	0 is returned.

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.

The [STATus:PRESet](#) command will set all bits in the PTR register to 1.

Return Format

<NR1>

The value returned is the binary-weighted sum of all bits set in the register.

Examples

The following command enables bits 0 and 1 (decimal value = 3) in the PTR register.

```
STAT:QUES:PTR 3, (@1)
```

The following query returns which bits are enabled in the register.

```
STAT:QUES:PTR? (@1)
```

Typical Response: +3

See Also

[STATus:QUEStionable:NTRansition](#)

[STATus:PRESet](#)

SYSTEM Commands Introduction

System commands control system functions that are not directly related to output control, measurement, or status functions.

Common commands also control system functions such as self-test, saving and recalling states, and others.

SCPI Command	Description
SYSTem	
:CHANnel	
[:COUNt]?	Returns the number of output channels in a mainframe
:MODEl? (@chanlist)	Returns the model number of the selected channel
:OPTion? (@chanlist)	Returns the option installed in the selected channel
:SERial? (@chanlist)	Returns the serial number of the selected channel
:COMMunicate	
:RLState LOCAL REMote RWLock	Specifies the Remote/Local state of the instrument
:ERRor?	Returns the error number and error string
:GROup	(Group commands do NOT apply to N678xA SMU)
:CATalog?	Returns the groups that have been defined
:DEFine (@chanlist)	Group multiple channels to create a single output
:DELete <channel>	Removes the specified channel from a group
:ALL	Ungroups all channels
:PARallel AUTO DIRect [,@chanlist]	Specifies how output modules are connected in parallel
:MMEMory	
:INITialize	Initializes the instrument's memory to the factory-default settings
:SANitize	Implements the clearing and sanitizing standard DoD 5220.22-M
:PASSword	
:FPANel	
:RESet	Resets the front panel lock password to zero
:REBoot	Returns the unit to its power-on state
:VERSion?	Returns the SCPI version number

SYSTem:CHANnel[:COUNT]?

Syntax

```
SYSTem:CHANnel[:COUNT]?
```

Description

This query returns the number of output channels in a mainframe.

Return Format

```
<NR1>
```

The number of output channels.

Example

The following query returns the number of channels.

```
SYST:CHAN?
```

Typical Response: +2

SYSTem:CHANnel:MODEl?**Syntax**

```
SYSTem:CHANnel:MODEl? [ (@<chanlist> ) ]
```

Description

This query returns the model numbers of the selected output channel.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<AARD>
```

The model number of the output channel. Multiple responses are separated by commas.

Example

The following query returns the model number in channels 1 and 2.

```
SYST:CHAN:MOD? (@1,2)
```

Typical Response: E4361A,E4362A

SYSTem:CHANnel:OPTion?**Syntax**

```
SYSTem:CHANnel:OPTion? [(@<chanlist>)]
```

Description

This query returns a list of options installed in each channel given in the channel list.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<SRD>
```

The query command returns an ASCII string enclosed in double quotes. If no option is installed, an empty quoted string (") is returned. Multiple responses are separated by commas.

Example

The following query returns the options in channels 1 and 2.

```
SYST:CHAN:OPT? (@1,2)
```

Response with options installed in channel 1, none in channel 2:

```
"Opt xxx", ""
```

SYSTEM:CHANnel:SERial?**Syntax**

```
SYSTem:CHANnel:SERial? [(@<chanlist>)]
```

Description

This query returns the serial number of the selected output channel.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<AARD>
```

The serial number of the output channel. Multiple responses are separated by semicolons.

Example

The following query returns the serial numbers in channels 1 and 2.

```
SYST:CHAN:SER? (@1,2)
```

Typical Response: MY00123456;MY00234567

SYSTem:COMMunicate:RLState

Syntax

```
SYSTem:COMMunicate:RLState <state>
```

```
SYSTem:COMMunicate:RLState?
```

Description

This command configures the remote/local state of the instrument according to the following settings:

LOCal	The instrument is set to front panel and remote interface control.
REMOte	The instrument is set to front panel and remote interface control.
RWLock	The front panel keys are disabled. The instrument can only be controlled via the remote interface. This programmable setting is completely independent from the front panel lock/unlock function that is available from the front panel menu.

NOTE

The remote/local state is unaffected by *RST or any SCPI commands other than SYSTem:COMMunicate:RLState.

Parameters

Name	Type	Range of Values	Default Value
<state>	CPD	LOCAl REMote RWLock	LOCAl (set at power-on)

Remarks

The remote/local instrument state can also be set by other interface commands over the GPIB and some other I/O interfaces.

When multiple remote programming interfaces are active, the interface with the most recently changed remote/local state determines the instrument's remote/local state.

Return Format

```
<CRD>
```

The query command returns LOC, REM, or RWL.

Examples

The following command sets the remote/local state of the instrument.

```
SYST:COMM:RLST REMote
```

The following query returns the remote/local state of the instrument.

```
SYST:COMM:RLST?
```

Typical Response: LOC

SYSTem:ERRor?

Syntax

SYSTem:ERRor?

Description

This query returns the next error number and its corresponding message string from the error queue.

For a complete listing of instrument and system error messages, see [SCPI Error Messages](#).

Remarks

The front-panel **ERR** annunciator turns on when one or more errors are currently stored in the error queue.

Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored.

Errors are cleared as you read them. When you have read all errors from the error queue, the **ERR** annunciator turns off and the errors are cleared.

If more errors have accumulated than the queue can hold, the last error stored in the queue (the most recent error) is replaced with -350, "Error queue overflow". No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the instrument responds with +0, "No error".

The error queue is cleared by the [*CLS](#) (Clear Status) command and when power is cycled. The errors are also cleared when you read the error queue.

Return Format

<NR1>,<SRD>

The error string may contain up to 80 characters and consists of an error number and an error string enclosed in double quotes. For example: -113,"Undefined header".

Example

The following query reads and clears one error.

```
SYST:ERR?
```

Typical Response: -101,"Invalid character"

SYSTem:GROup:CATalog?

Syntax

```
SYSTem:GROup:CATalog?
```

Description

This query returns information about channels that are grouped. The defined groups are enclosed in quotes.

Return Format

```
<SRD>
```

The defined groups in quotes. Multiple responses are separated by commas.

Example

The following query returns the channels that are grouped.

```
SYST:GRO:CAT?
```

Response with 2 channels; none grouped: "1" , "2"

Response with channels 1 + 2 grouped: "1,2"

SYSTem:GROup:DEFine

Syntax

```
SYSTem:GROup:DEFine [ (@<chanlist> ) ]
```

Description

This command defines a list of output channels as a paralleled group. This effectively creates a single output with higher current and power capability. You can group two channels per mainframe.

All channels in a group must have identical model numbers and have identical options installed.

All grouped channels must have their output terminals and, if remote sensing, their sense terminals connected in parallel.

After the channels are wired in parallel and defined as a group, they can be addressed using any of the channel-specific SCPI commands by sending the channel number of the **LOWEST** channel in the group.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

For the group changes to take effect, you must reboot the unit. Either cycle AC power or send the [SYSTem:REBoot](#) command.

Group channel definitions are stored in non-volatile memory and are unaffected by [*RST](#) or [*RCL](#).

Group channel *settings* (voltage, current, etc.) are set and saved by [*RST](#) or [*RCL](#).

This command also deletes any previously saved states.

Example

The following command defines channels 1 and 2 as a group.

```
SYST:GRO:DEF (@1,2)
```


SYSTem:GROup:DELeTe

Syntax

SYSTem:GROup:DELeTe <channel>

Description

This command removes the indicated channel from a group. It leaves the other channels in the group intact.

When ungrouping a channel, you must also remove the parallel connections between the output and sense terminals of that channel.

Parameters

Name	Type	Range of Values	Default Value
<channel>	NR1	1 2	Required parameter

Remarks

For the group changes to take effect, you must reboot the unit. Either cycle AC power or send the [SYSTem:REBoot](#) command.

This command also deletes any previously saved states.

Example

The following command deletes channel 2 from a group.

```
SYST:GRO:DEL 2
```

SYSTem:GROup:DELeTe:ALL

Syntax

```
SYSTem:GROup:DELeTe:ALL
```

Description

This command deletes all groups on a mainframe. It returns any channels that have been grouped back to an ungrouped state.

When ungrouping channels, you must also remove all paralleled sense and output terminal connections between channels.

Remarks

For the group changes to take effect, you must reboot the unit. Either cycle AC power or send the [SYSTem:REBoot](#) command.

This command also deletes any previously saved states.

Example

The following command deletes all groups.

```
SYST:GRO:DEL:ALL
```

SYSTem:GROup:PARallel

Syntax

```
SYSTem:GROup:PARallel <mode> [ ,(@<chanlist>)]
```

```
SYSTem:GROup:PARallel? [(@<chanlist>)]
```

Description

This command tells the instrument how the output modules are physically connected in parallel.

DIRect The outputs are connected in direct parallel, with no additional connections on the control connector.

AUTO The outputs are connected in direct parallel, with no additional connections on the control connector.

Parameters

Name	Type	Range of Values	Default Value
<mode>	CPD	AUTO DIRect	Required Parameter
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

For the group changes to take effect, you must reboot the unit. Either cycle AC power or send the [SYSTem:REBoot](#) command.

Return Format

```
<CRD>
```

The query command returns AUTO or DIRect.

Examples

The following command defines channels 1 and 2 as auto-paralleled

```
SYST:GRO:PAR (@1,2)
```

The following command returns the paralleled setting of the output channels.

```
SYST:GRO:PAR? (@1,2)
```

Typical Response: AUTO,AUTO

SYSTem:MMEMory:INITialize

Syntax

```
SYSTem:MMEMory:INITialize
```

Description

This command restores the mass memory file system after it has been destroyed using the SYSTem:MMEMory:SANitize command. The mass memory file system is used for storing user-defined tables.

Example

The following command restores the factory-default memory settings

```
SYST:MMEM:INIT
```

See Also

[SYSTem:MMEMory:SANitize](#)

SYSTem:MMEMory:SANitize

Syntax

```
SYSTem:MMEMory:SANitize
```

Description

This command implements the clearing and sanitizing standard DoD 5220.22-M for flash memory devices. The sanitize process for mass memory consists of the following four steps:

1. writes a 0 to all locations
2. writes a 1 to all locations
3. writes a pseudo-random value to all locations
4. perform a full chip erase as per the manufacturer's data sheet.

When this command completes, the file system is destroyed, the instrument is disabled, and must be turned off. A selftest error is generated if the instrument is turned on and mass memory has not been restored using SYST:MMEM:INIT. Note that the instrument will be functional in Fixed and SAS modes, but not in Table mode.

Example

The following command sanitizes the flash memory of the instrument.

```
SYST:MMEM:SAN
```

See Also

[SYSTem:MMEMory:INITialize](#)

SYSTem:PASSword:FPANel:RESet

Syntax

```
SYSTem:PASSword:FPANel:RESet
```

Description

This command resets the front panel lockout password to the factory-shipped setting, which is zero (0). This command does not reset the administrator/calibration password.

NOTE

The administrator/calibration password can also be reset to 0 by setting an internal switch on the unit. Refer to the Service Manual under “Calibration Switches” for more information. This switch does NOT reset the front panel lockout password.

Example

The following command resets the front panel password.

```
SYST:PASS:FPAN:RES
```

See Also

[*RST](#)

SYSTem:REBoot

Syntax

SYSTem:REBoot

Description

This command causes the instrument to reboot to its power-on state.

Example

The following command reboots the instrument.

```
SYST:REB
```

See Also

[*RST](#)

SYSTem:VERSion?

Syntax

SYSTem:VERSion?

Description

This command returns the version of the SCPI (Standard Commands for Programmable Instruments) standard with which the instrument is in compliance.

Return Format

<SRD>

The command returns a string in the form "YYYY.V", where YYYY represents the year of the version and V represents a version for that year.

Example

The following query returns the SCPI version.

```
SYST:VERS?
```

Typical Response: "1995.0"

TRIGger Subsystem Introduction

Trigger commands control the remote triggering of the instrument. They specify the trigger source for the transient and the measurement system and also generate software triggers.

The Common command [*TRG](#) is also used to trigger the instrument.

SCPI Command	Description
TRIGger	
[:TRANsient]	
[:IMMEDIATE] [(@chanlist)	Triggers the output immediately
:SOURce BUS DWEL<n> PIN<n> TRAN<n>[,(@chanlist)]	Sets the output trigger source
:ACQuire	
[:IMMEDIATE] [(@chanlist)	Triggers the measurement immediately
:DELay <NRf> [,@chanlist)	Specifies a time delay for the measurement trigger
:SOURce BUS PIN<n> TRAN<n> [,@chanlist)]	Sets the measurement trigger source
:DLOG	
[:IMMEDIATE] [(@chanlist)	Triggers the datalog immediately
:SOURce BUS DLOG<n> PIN<n> TRAN<n>[,(@chanlist)]	Sets the datalog trigger source
:PIN<3-7>	
:SOURce BUS ACQ<n> DLOG<n> DWEL<n> TRAN<n>	Selects the trigger source for the digital I/O pin

TRIGger[:TRANsient][:IMMediate]**Syntax**

```
TRIGger[:TRANsient][:IMMediate] [(@<chanlist>)]
```

Description

When the transient trigger system has been initiated, this command generates an immediate output trigger regardless of the selected trigger source

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

You must initiate the transient trigger system before you can send any triggers.

When sent, the transient trigger will initiate an output change in LIST mode as specified by the [SOURce:]LIST commands.

When the transient trigger sequence has completed, the WTG-tran bit in the Status Operation Condition register is cleared.

Example

The following command generates an output trigger on channels 1 and 2.

```
TRIG:TRAN (@1,2)
```

See Also

[INITiate\[:IMMediate\]:TRANsient](#)

[*TRG](#)

TRIGger[:TRANsient]:SOURce

Syntax

```
TRIGger[:TRANsient]:SOURce <source> [ ,(@<chanlist>)]
TRIGger[:TRANsient]:SOURce? [(@<chanlist>)]
```

Description

This command selects the trigger source for the output trigger system. The following trigger sources can be selected:

- BUS** Selects GPIB device trigger, *TRG, or <GET> (Group Execute Trigger).
- DWELI<n>** Selects the output channel's dwell timer. <n> specifies the output channel.
- PIN<n>** Selects an output port pin configured as a trigger input. <n> specifies the pin number.
- TRANsient<n>** Selects the output channel's output trigger. <n> specifies the output channel.

Parameters

Name	Type	Range of Values	Default Value
<source>	CPD	BUS DWELI <1-2> PIN<3-7> TRANsient<1-2>	BUS
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

PIN<n> must be configured as trigger input before it can be used as a trigger source. See [\[SOURce:JDIGital:PIN<n>:FUNctIon\]](#) and [\[SOURce:JDIGital:PIN<n>:POLarity\]](#).

You must configure the LIST mode to use the Dwell list as a trigger source. See: [\[SOURce:JLIST:STEP\]](#).

Return Format

<CRD>

The query command returns BUS, DWEL<n>, PIN<n>, or TRAN<n>.

Examples

The following command selects pin 1 as the output trigger source for channel 2.

```
TRIG:TRAN:SOUR PIN1 , (@2)
```

3 Commands by Subsystem

The following query returns the output trigger source of channel 1.

```
TRIG:TRAN:SOUR? (@1)
```

Typical Response: BUS

See Also

[\[SOURCE: DIGital:PIN<1-7>:FUNCTION](#)

[\[SOURCE: DIGital:PIN<1-7>:POLarity](#)

[\[SOURCE: LIST:STEP](#)

TRIGger:ACQuire[:IMMediate]

Syntax

```
TRIGger:ACQuire[:IMMediate] [(@<chanlist>)]
```

Description

When the acquisition trigger system has been initiated, this command generates an immediate measurement trigger regardless of the selected trigger source.

A measurement trigger causes the instrument to measure the output voltage and current and store the results in a buffer

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

You must initiate the acquisition trigger system before you can send any triggers.

The measured quantity, either voltage or current, must be specified by the SENSE:FUNCTION command.

When the triggered measurement has completed, the WTG-meas bit in the Status Operation Condition register is cleared.

Example

The following command generates a measurement trigger on channels 1 and 2.

```
TRIG:ACQ (@1,2)
```

See Also

[INITiate\[:IMMediate\]:ACQuire](#)

[*TRG](#)

TRIGger:ACQuire:DELAy

Syntax

```
TRIGger:ACQuire:DELAy <delay>[ , (@<chanlist>)]
```

```
TRIGger:ACQuire:DELAy? [ (@<chanlist>)]
```

Description

This command specifies a time delay for the acquisition trigger after the trigger is received. The delay is programmed in seconds, from 0 to 0.255.

Parameters

Name	Type	Range of Values	Default Value
<delay>	NRf	0 - 0.255	0
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Return Format

```
<NR3>
```

The query command returns the programmed trigger delay. Multiple responses are separated by commas.

Examples

The following command programs a trigger delay of 100ms on channel 1

```
TRIG:ACQ:DEL 0.100, (@1)
```

The following query returns the trigger delay on channel 2..

```
TRIG:ACQ:DEL? (@2)
```

Typical Response: 0

See Also

[INITiate\[:IMMediate\]:ACQuire](#)

[*TRG](#)

TRIGger:ACQuire:SOURce

Syntax

```
TRIGger:ACQuire:SOURce <source> [ ,(@<chanlist>)]
```

```
TRIGger:ACQuire:SOURce? [(@<chanlist>)]
```

Description

This command selects the trigger source for the measurement subsystem. The following trigger sources can be selected:

- BUS** Selects a remote interface trigger command.
- PIN<n>** Selects an output port pin configured as a trigger input. <n> specifies the pin number.
- TRANSient<n>** Selects the transient system of an output channel as the trigger source. <n> specifies the channel.

Parameters

Name	Type	Range of Values	Default Value
<source>	CPD	BUS PIN<3-7> TRANSient<1-2>	BUS
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

PIN<n> must be configured as trigger input before it can be used as a trigger source. See [\[SOURce:|DIGital:PIN<n>:FUNCTION\]](#) and [\[SOURce:|DIGital:PIN<n>:POLarity\]](#).

Return Format

<CRD>

The query command returns BUS, PIN<n>, or TRAN<n>.

Examples

The following command selects pin 1 as the measurement trigger source for channel 2.

```
TRIG:ACQ:SOUR PIN1 , (@2)
```

The following query returns the measurement trigger source of channel 1.

```
TRIG:ACQ:SOUR? (@1)
```

Typical Response: BUS

See Also

[\[SOURCE: |DIGital:PIN<1-7>:FUNction](#)

[\[SOURCE: |DIGital:PIN<1-7>:POLarity](#)

TRIGger:DLOG[:IMMediate]**Syntax**

```
TRIGger:DLOG[:IMMediate] [(@<chanlist>)]
```

Description

When the data logger is initiated, this command generates an immediate data log trigger regardless of the selected trigger source.

Parameters

Name	Type	Range of Values	Default Value
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

You must initiate the data log trigger system before you can send any triggers.

When sent, the data log trigger immediately starts the data log.

Example

The following command starts a data log on channels 1 and 2.

```
TRIG:DLOG (@1,2)
```

See Also

[INITiate\[:IMMediate\]:DLOG](#)

[*TRG](#)

TRIGger:DLOG:SOURce

Syntax

```
TRIGger:DLOG:SOURce <source> [ ,(@<chanlist>)]
```

```
TRIGger:DLOG:SOURce? [(@<chanlist>)]
```

Description

This command selects the trigger source for the data logger. The following trigger sources can be selected:

- BUS** Selects GPIB device trigger, *TRG, or <GET> (Group Execute Trigger).
- DLOG<n>** Selects the output channel's data log timer. <n> specifies the output channel.
- PIN<n>** Selects an output port connector pin. <n> specifies the pin number.
- TRANSient<n>** Selects the transient system of an output channel as the trigger source. <n> specifies the output channel.

Parameters

Name	Type	Range of Values	Default Value
<source>	CPD	BUS DLOG<1-2> PIN<3-7> TRANSient<1-2>	BUS
<chanlist>	Numeric	One or more channels. (@2) - channel 2 (@1,2) - channels 1 and 2	Optional parameter

Remarks

Pins 3–7 must be configured as trigger outputs before they can be used as a trigger source. See [\[SOURCE:\]DIGital:PIN<n>:FUNCTION](#) and [\[SOURCE:\]DIGital:PIN<n>:POLarity](#).

Return Format

```
<CRD>
```

The query command returns BUS, DLOG<n>, PIN<n>, or TRAN<n>.

Examples

The following command selects pin 3 as the data log trigger source for channel 2.

```
TRIG:DLOG:SOUR PIN3, (@2)
```

The following query returns the data log trigger source of channel 1.

```
TRIG:DLOG:SOUR? (@1)
```

Typical Response: BUS

See Also

[\[SOURCE: |DIGital:PIN<1-7>:FUNction](#)

[\[SOURCE: |DIGital:PIN<1-7>:POLarity](#)

TRIGger:PIN<3-7>:SOURce**Syntax**

```
TRIGger:PIN<3-7>:SOURce <source>
```

```
TRIGger:PIN<3-7>:SOURce?
```

Description

This command selects the trigger source for the specified digital I/O pin when the pin is configured for the trigger output function using [DIGital:PIN<1-7>:FUNCTION TOUT](#). Only pins 3-7 can be configured as trigger outputs. The following trigger sources can be selected for these pins:

BUS	Selects GPIB device trigger, *TRG, or <GET> (Group Execute Trigger).
ACQuire<n>	Selects the output channel's measurement trigger. <n> specifies the output channel.
DLOG<n>	Selects the output channel's data log timer. <n> specifies the output channel.
DWELI<n>	Selects the output channel's dwell timer. <n> specifies the output channel.
TRANSient<n>	Selects the output channel's output trigger. <n> specifies the output channel.

Parameters

Name	Type	Range of Values	Default Value
<source>	CPD	BUS ACQuire<1-2> DLOG<1-2> DWELI<1-2> TRANSient<1-2>	NONE

Remarks

You must configure the LIST mode to use the Dwell list as a trigger source. See: [\[SOURce\]:\[LIST\]:STEP](#).

The output channel's measurement and transient triggers must be initiated to be used as trigger sources. See [INITiate\[:IMMediate\]:ACQuire](#) and [INITiate\[:IMMediate\]:TRANSient](#).

Return Format

```
<CRD>
```

The query command returns BUS, ACQ<n>, DLOG<n>, DWEL<n>, or TRAN<n>.

Examples

The following command selects DWEL1 as pin 1's trigger source.

```
TRIG:PIN1:SOUR DWEL1
```

The following query returns the output trigger source of pin 7.

```
TRIG:PIN7:SOUR?
```

Typical Response: BUS

See Also

[\[SOURCE: DIGital:PIN<1-7>:FUNCTION](#)

[\[SOURCE: DIGital:PIN<1-7>:POLarity](#)

[\[SOURCE: LIST:STEP](#)



4

Command Quick Reference

SCPI Command Summary	224
Non-Volatile Settings	229
Reset Settings (*RST)	230

SCPI Command Summary

NOTE

Some [optional] commands have been included for clarity. All settings commands have a corresponding query. Not all commands apply to all models.

SCPI Command	Description
ABORt	
[:TRANsient] [(@chanlist)]	Resets the transient trigger system to the Idle state
:ACQuire [(@chanlist)]	Resets the measurement trigger system to the Idle state
:DLOG [(@chanlist)]	Stops the currently running datalog
CALibrate	
:CURRent	
[:DATA] <NRf>	Enters the calibration value
:LEVel P1 P2, (@channel)	Calibrates the output current programming
:PROTection (@channel)	Calibrates the over-current protection
:DATE <SPD>, (@channel)	Sets the calibration date
:PASSword <NRf>	Sets the numeric calibration password
:SAVE	Saves the new cal constants in non-volatile memory
:STATE <Bool> [,<NRf>]	Enables/disables calibration mode
:VOLTag	
[:DATA] <NRf>	Enters the calibration value
:LEVel P1 P2, (@channel)	Calibrates the output voltage programming
:PROTection (@channel)	Calibrates the over-voltage protection
DISPlay	
[:WINDow]	
:TEXT <SPD>	Sends the text to be displayed on the front panel
:VIEW CHAN1 CHAN2 ALL TEXT	Selects the channels displayed in meter view
:ENAbLe <Bool>	Enables/disables the front panel display
FETCh	
[:SCALar]	
:CURRent [:DC]? [(@chanlist)]	Returns the average output current
:VOLTag [:DC]? [(@chanlist)]	Returns the average output voltage
DLOG [(@chanlist)]	Returns measurements from the datalog buffer
FORMat	
[:DATA] ASCII REAL [,<NRf>]	Selects transferred data format
:BORder NORMAl SWAPped [,<NRf>]	Specifies how binary data is transferred
INITiate	
[:IMMediate]	
[:TRANsient] [(@chanlist)]	Enables output triggers
:ACQuire [(@chanlist)]	Enables measurement triggers
:DLOG [(@chanlist)]	Enables the datalogger
:CONTinuous	
[:TRANsient] <Bool> [,<NRf>]	Enables/disables continuous transient triggers
MEASure	
[:SCALar]	
:CURRent [:DC]? [(@chanlist)]	Takes a measurement; returns the average output current
:VOLTag [:DC]? [(@chanlist)]	Takes a measurement; returns the average output voltage

SCPI Command	Description
MEMory	
:COpy	
:TABLE <CPD>	Copies the selected table to non-volatile memory
:DElete	
[:NAME] <CPD>	Deletes the specified table in volatile and non-volatile memory
:ALL	Deletes all tables in volatile and non-volatile memory
:TABLE	
:CATalog?	Returns all table names in volatile and non-volatile memory
:CURRent	
[:MAGnitude] <NRf>{,<NRf>}	Programs a list of current points for a new table
:POINts?	Returns the number of current points in the active table
:SElect [<CPD>]	Creates a new table in volatile memory
:VOLTage	
[:MAGnitude] <NRf>{,<NRf>}	Programs a list of voltage points for a new table
:POINts?	Returns the number of voltage points in the active table
OUTPut	
[:STATe] <Bool> [,@chanlist]	Enables/disables the specified output channel(s)
:COUple	
[:STATe] <Bool>	Enables/disables channel coupling for output synchronization
:CHANnel [<NR1> {,<NR1>}]	Selects which channels are coupled
:INHibit	
:MODE LATCHing OFF	Sets the remote inhibit mode
:PON	
:STATe RST RCL0	Programs the power-on state
:PROtection	
:CLEar [(@chanlist)]	Resets latched protection
:COUple <Bool>	Enables/disables channel coupling for protection faults
SENSe	
:DLOG	
:CLOCK TRIGger TIMer [,@chanlist]	Specifies the measurement capture trigger source
:TINTerval <NRf> [,@chanlist]	Sets the time interval for capturing measurements
[SOURce:]	
CURRent	
[:LEVel]	
[:IMMediate][:AMPLitude] <NRf+> [,@chanlist]	Sets the output current in Fixed mode
:DTABLE	
:SASimulator	
[:IMMediate]? [(@chanlist)]	Returns the calculated DAC table used for the SAS settings
[:IMMediate]:IMP? [(@chanlist)]	Returns the calculated DAC table Imp value for the SAS settings
[:IMMediate]:ISC? [(@chanlist)]	Returns the calculated DAC table Isc value for the SAS settings
:LIST? <NR1> [,@chanlist]	Returns the calculated DAC table used for the specified step
:LIST:IMP? <NR1> [,@chanlist]	Returns the calculated DAC table Imp value for the specified step
:LIST:ISC? <NR1> [,@chanlist]	Returns the calculated DAC table Isc value for the specified step
:TABLE	
[:IMMediate]? <CPD> [,@chanlist]	Returns the calculated DAC table used for the specified table
[:IMMediate]:IMP? <CPD> [,@chanlist]	Returns the calculated DAC table Imp value for the specified table
[:IMMediate]:ISC? <CPD> [,@chanlist]	Returns the calculated DAC table Isc value for the specified table
:MODE FIXed SAS TABLE [,@chanlist]	Sets the operating mode of the instrument
:DTAbLe 256 4096 [,@chanlist]	Specifies the size of the table in points
:PROGramming INTernal EXTernal [,@chanlist]	Specifies the programming source in auto-parallel operation

4 Command Quick Reference

SCPI Command	Description
:PROTection [:LEVel] <NRf+> [,@chanlist]	Sets the over-current protection level
:SAS :IMP <NRf> [,@chanlist] :ISC <NRf> [,@chanlist] :MODE IMMEDIATE LIST [,@chanlist]	Sets the current at the peak power point of the curve Sets the short-circuit current Selects the source of the curve parameters
:SLIMit :HIGH <NRf+> MAXimum [,@chanlist]	Sets the high soft limit for programming the output current
:TABLE :NAME <CPD> [,@<chanlist>] :OFFSet <NRf> [,@<chanlist>]	Activates a user-defined table in Table mode Adds a current offset when operating in Table mode
DIGital	
:INPut :DATA?	Reads the state of the digital port pins
:OUTPut :DATA <NRf>	Sets the digital port
:PIN<1-7> :FUNctIon DIO DINPut TOUTput TINPut FAULT INHibit ONCouple OFFCouple :POLarity POSitive NEGative	Sets the selected pin's function Sets the selected pin's polarity
LIST	
:COUNt <NRf+> INFINity [,@chanlist] :DWELI <NRf> {,<NRf>} [,@chanlist] :POINts? [,@chanlist]	Sets the list repeat count Sets the list of dwell times Returns the number of dwell list points
:SAS :IMP [:LEVel] <NRf>{,<NRf>} [,@chanlist] :POINts? (@chanlist) :ISC [:LEVel] <NRf>{,<NRf>} [,@chanlist] :POINts? (@chanlist) :VMP [:LEVel] <NRf>{,<NRf>} [,@chanlist] :POINts? (@chanlist) :VOC [:LEVel] <NRf>{,<NRf>} [,@chanlist] :POINts? (@chanlist)	Sets the list of peak power current points Returns the number of peak power current points Sets the list of short-circuit current points Returns the number of short-circuit points Sets the list of peak power voltage points Returns the number of peak power voltage points Sets the list of open-circuit voltage points Returns the number of open-circuit points
:STEP ONCE AUTO [,@chanlist] :ACTive? [,@chanlist]	Specifies how the list responds to triggers Returns the present list step
:TERMinate :LAST <Bool> [,@chanlist]	Sets the list termination mode
VOLTage [:LEVel] [:IMMEDIATE][:AMPLitude] <NRf+> [,@chanlist]	Sets the output voltage in Fixed mode
:DTABLE :SASimulator [:IMMEDIATE]? [,@chanlist] [:IMMEDIATE]:VMP? [,@chanlist] [:IMMEDIATE]:VOC? [,@chanlist] :LIST? <NR1> [,@chanlist] :LIST:VMP? <NR1> [,@chanlist] :LIST:VOC? <NR1> [,@chanlist]	Returns the calculated DAC table used for the SAS settings Returns the calculated DAC table Vmp value for the SAS settings Returns the calculated DAC table Voc value for the SAS settings Returns the calculated DAC table used for the specified step Returns the calculated DAC table Vmp value for the specified step Returns the calculated DAC table Voc value for the specified step

SCPI Command	Description
:TABLE	
[:IMMEDIATE]? <CPD> [,@chanlist]	Returns the calculated DAC table used for the specified table
[:IMMEDIATE]:VMP? <CPD> [,@chanlist]	Returns the calculated DAC table Vmp value for the specified table
[:IMMEDIATE]:VOC? <CPD> [,@chanlist]	Returns the calculated DAC table Voc value for the specified table
:PROTECTION	
[:LEVEL] <NRf>, (@chanlist)	Sets the over-voltage protection level
:SAS	
:VMP <NRf>, (@chanlist)	Sets the voltage at the peak power point of the curve
:VOC <NRf>, (@chanlist)	Sets the open-circuit voltage
:SLIMIT	
:HIGH <NRf+> MAXIMUM [,@chanlist]	Sets the high soft limit for programming the output voltage
:TABLE	
:OFFSET <NRf> [,@chanlist]	Adds a voltage offset when operating in Table mode
STATUS	
:OPERATION	
[:EVENT]? [(@chanlist)]	Returns the value of the operation event register
:CONDITION? [(@chanlist)]	Returns the value of the operation condition register
:ENABLE <NRf> [,@chanlist]	Enables specific bits in the Event register
:NTRANSITION <NRf> [,@chanlist]	Sets the Negative transition filter
:PTRANSITION <NRf> [,@chanlist]	Sets the Positive transition filter
:PRESET	Presets all enable and transition registers to power-on
:QUESTIONABLE	
[:EVENT]? [(@chanlist)]	Returns the value of the questionable event register
:CONDITION? [(@chanlist)]	Returns the value of the questionable condition register
:ENABLE <NRf> [,@chanlist]	Enables specific bits in the Event register
:NTRANSITION <NRf> [,@chanlist]	Sets the Negative transition filter
:PTRANSITION <NRf> [,@chanlist]	Sets the Positive transition filter
SYSTEM	
:CHANNEL	
[:COUNT]?	Returns the number of output channels in a mainframe
:MODEL? [(@chanlist)]	Returns the model number of the selected channel
:OPTION? [(@chanlist)]	Returns the option installed in the selected channel
:SERIAL? [(@chanlist)]	Returns the serial number of the selected channel
:COMMUNICATE	
:RLSTATE LOCAL REMOTE RWLOCK	Specifies the Remote/Local state of the instrument
:ERROR?	Returns the error number and error string
:GROUP	
:CATALOG?	Returns the groups that have been defined
:DEFINE [(@chanlist)]	Group multiple channels together to create a single output
:DELETE <channel>	Removes the specified channel from a group
:ALL	Ungroups all channels
:PARALLEL AUTO DIRECT [,@chanlist]	Specifies how output modules are connected in parallel
:MEMORY	
:INITIALIZE	Initializes the instrument's memory to the factory-default settings
:SANITIZE	Implements the clearing and sanitizing standard DoD 5220.22-M
:PASSWORD	
:FPANEL	
:RESET	Resets the front panel lock password to zero
:REBOOT	Returns the unit to its power-on state
:VERSION?	Returns the SCPI version number

4 Command Quick Reference

SCPI Command	Description
TRIGger	
[:TRANsient]	
[:IMMEDIATE] [(@chanlist)]	Triggers the output immediately
:SOURce BUS DWEL<n> PIN<n> TRAN<n>[.(@chanlist)]	Sets the output trigger source
:ACQuire	
[:IMMEDIATE] [(@chanlist)]	Triggers the measurement immediately
:DELay <NRf> [.(@chanlist)]	Specifies a time delay for the measurement trigger
:SOURce BUS PIN<n> TRAN<n> [.(@chanlist)]	Sets the measurement trigger source
:DLOG	
[:IMMEDIATE] [(@chanlist)]	Triggers the datalog immediately
:SOURce BUS DLOG<n> PIN<n> TRAN<n>[.(@chanlist)]	Sets the datalog trigger source
:PIN<3-7>	
:SOURce BUS ACQ<n> DLOG<n> DWEL<n> TRAN<n>	Selects the trigger source for the digital I/O pin
*CLS	Clear status
*ESE <NRf>	Standard event status enable
*ESR?	Return event status register
*IDN?	Return instrument identification
*LRN	Returns a sequence of SCPI commands
*OPC	Enable "operation complete" bit in ESR
*OPT?	Return option number
*RCL <NRf>	Recalls a saved instrument state
*RDT?	Return output channel descriptions
*RST	Reset
*SAV <NRf>	Saves an instrument state
*SRE <NRf>	Set service request enable register
*STB?	Return status byte
*TRG	Trigger
*TST?	Performs self-test, then returns result
*WAI	Pauses additional command processing until all device commands are done

Non-Volatile Settings

The following tables show the factory-shipped non-volatile settings of the instrument. Information in non-volatile memory is **NOT** lost when power is turned off. These settings are all customer-configurable.

The factory-shipped LAN settings can be restored by selecting the front panel Reset control located in the System\I/O\LAN\Config\Reset menu.

All non-volatile settings including LAN, can be restored by selecting the front panel Reset control located in the System\Admin\Nvram menu

Factory-shipped non-volatile LAN settings

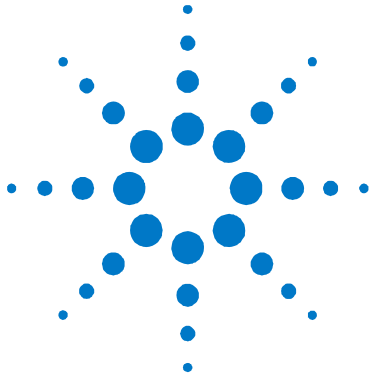
DHCP	Enabled	Host name	A-E4360-xxxxx
AutoIP	Enabled	Use DNS naming service	Enabled
IP Address	169.254.67.0	Use NetBIOS naming service	Enabled
Subnet Mask	255.255.0.0	Domain name	Blank
Default Gateway	0.0.0.0	TCP keepalive	Enabled
Obtain DNS server from DHCP	Enabled	TCP keepalive seconds	1800
DNS server	Blank		

Other factory-shipped non-volatile settings

Admin/Calibration password	0 (zero)	On/Off key affects all channels	Disabled
Calibration date	May 1, 2008	Output Inhibit mode	Off
Channel grouping	No groups	Saved states	*RST command
Digital port function (all pins)	Digital In	Screen contrast	50%
Digital port polarity (all pins)	Positive	Screen saver	Enabled
Front panel lockout	Disabled	Screen saver delay	60 minutes
Front panel meter view	1-channel	USB interface	Enabled
GPIB Address	5	Wake on I/O	Enabled
Key clicks	Enabled	Web server	Enabled
LAN interface	Enabled		

Reset Settings (*RST)

These settings are set by the *RST (Reset) command			
CAL:STAT	OFF	[SOUR:]LIST:SAS:VOC	65 E4361A; 130 E4362A
[SOUR:]CURR	0	[SOUR:]LIST:STEP	AUTO
[SOUR:]CURR:MODE	FIX	[SOUR:]LIST:TERM:LAST	OFF
[SOUR:]CURR:PROT:LEV	1.1 X MAX	MEM:TABL:CURR	0
[SOUR:]CURR:SAS:IMP	0.8 X MAX	MEM:TABL:VOLT	0
[SOUR:]CURR:SAS:ISC	MAX	OUTP	OFF
[SOUR:]CURR:SAS:MODE	IMM	OUTP:COUP	OFF
[SOUR:]CURR:TABL:OFFS	0	OUTP:PROT:COUP	OFF
[SOUR:]DIG:OUTP:DATA	0	TRIG:ACQ:DEL	0
DISP:ENAB	ON	TRIG:ACQ:SOUR	BUS
DISP:VIEW	METER1	TRIG:TRAN:SOUR	BUS
INIT:CONT:TRAN	OFF	TRIG:PIN:SOUR	NONE
[SOUR:]LIST:COUN	1	[SOUR:]VOLT	0
[SOUR:]LIST:DWEL	0.025	[SOUR:]VOLT:PROT	MAX
[SOUR:]LIST:SAS:IMP	6.528 E4361A; 3.264 E4362A	[SOUR:]VOLT:SAS:VMP	0.8 X MAX
[SOUR:]LIST:SAS:ISC	8.7 E4361A; 5.1 E4362A	[SOUR:]VOLT:SAS:VOC	MAX
[SOUR:]LIST:SAS:VMP	60 E4361A; 120 E4362A	[SOUR:]VOLT:TABL:OFFS	0



5 Programming Examples

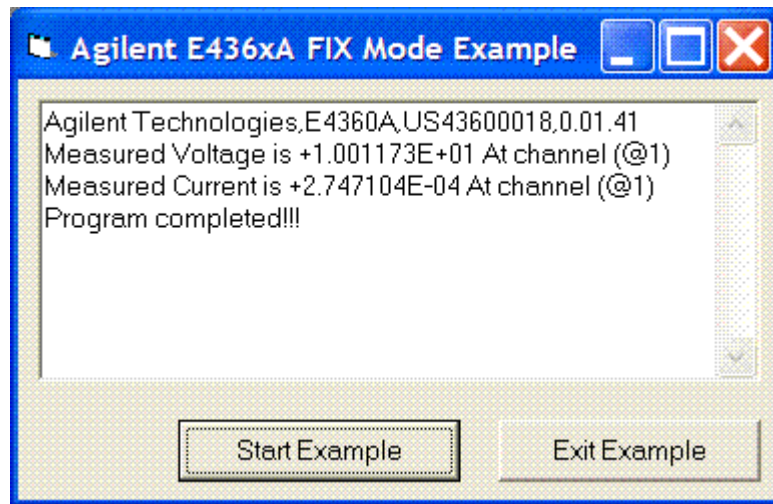
Output Programming Example (Fixed Mode only)	232
List Programming Example	235

For your convenience, this chapter contains several Microsoft Visual Basic 6.0 example programs to help you develop programs for your own application. The example programs are for illustration only, and are provided with the assumption that you are familiar with the programming language being demonstrated and the tools used to create and debug procedures.

Output Programming Example (Fixed Mode only)

This is a simple program that sets a voltage level, current limit, and the over-voltage level in Fixed mode. This example also measures the output voltage and current.

When the program runs, it displays a dialog box that will contain the measured values. Buttons on the dialog let you Start the example or Exit the example.



You have a royalty-free right to use, modify, reproduce and distribute this example program (and/or any modified version) in any way you find useful, provided you agree that Agilent Technologies has no warranty, obligations, or liability for any example programs.

This program is written in Microsoft Visual Basic 6.0 using the VISA IO library. The VISA library must be downloaded from the Agilent IO Libraries Suite CD-ROM to use this program. For information about using VISA in another Visual Basic project, refer to “Programming Your Instruments” in the USB/LAN/GPIB Interfaces Connectivity Guide, also included on the Agilent IO Libraries Suite CD-ROM.

```
Option Explicit
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Private Sub StartExample_Click()
'
' Agilent E436xA Solar Array Simulator VB6 Example Program
'
' This program demonstrates programming the instrument in FIXed mode
'
' variable declarations
'
Dim DeviceAddress As String
```



```

Dim ChannelList As String
Dim VoltSetting As Double
Dim CurrSetting As Double
Dim overVoltSetting As Double
    On Error GoTo ErrorHandler
    'These variable are necessary to initialize the VISA COM.
Dim ioMgr As AgilentRMLib.SRMClS
Dim Instrument As VisaComLib.FormattedIO488
    ' The following command line provides the program with the VISA name of the
    ' interface that it will communicate with. It is currently set to use GPIB.
DeviceAddress = "GPIB0::5::INSTR"
    ' Use the following line for LAN communication using an IP address
'DeviceAddress="TCPIP0::141.25.36.214"
    ' Use the following line for LAN communication using a hostname
'DeviceAddress = "TCPIP0::a-e4360a-00018"
    ' Use the following line for USB communication
'DeviceAddress = "USB0::2391::4359::US43600018"
    ' Initialize the VISA COM communication
Set ioMgr = New AgilentRMLib.SRMClS
Set Instrument = New VisaComLib.FormattedIO488
Set Instrument.IO = ioMgr.Open(DeviceAddress)
    ' send power-on reset to the instrument
Instrument.WriteString "*RST"
    ' Query the instrument for the IDN string
Instrument.WriteString "*IDN?"
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
    ' Desired settings for this program
overVoltSetting = 15    ' volts
VoltSetting = 10       ' volts
CurrSetting = 2        ' amps
ChannelList = "@1"     ' channel 1 only
With Instrument
    ' Set the voltage
    .WriteString "VOLT" & Str$(VoltSetting) & "," & ChannelList
    ' Set the over voltage level
    .WriteString "VOLT:PROT:LEV " & Str$(overVoltSetting) & "," & ChannelList
    ' Set current limit
    .WriteString "CURR " & Str$(CurrSetting) & "," & ChannelList
    ' Turn the output on
    .WriteString "OUTP ON," & ChannelList

```

5 Programming Examples

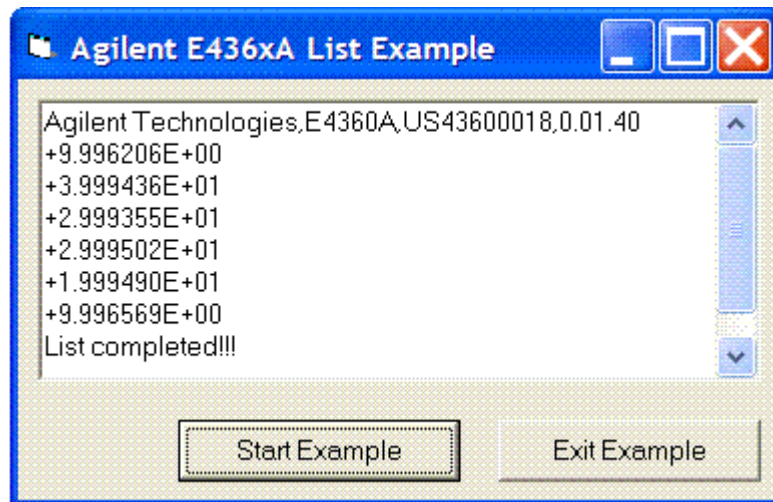
```
' Wait for previous commands to complete.
.WriteString "*OPC?"
.ReadString
' Measure the voltage
.WriteString "MEAS:VOLT? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & "Measured Voltage is " & .ReadString &
" At channel " & ChannelList & vbCrLf
' Fetch output current measurement taken at the previous measurement
.WriteString "FETC:CURR? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & "Measured Current is " & .ReadString &
" At channel " & ChannelList & vbCrLf
' Turn the output off
.WriteString "OUTP OFF," & ChannelList
'
StatusTextBox.Text = StatusTextBox.Text & "Program completed!!!" & vbCrLf
End With
Exit Sub
ErrorHandler:
    MsgBox "ErrorHandler:" & vbCrLf & Err.Description
    ReadInstrumentError Instrument
    Exit Sub
End Sub
Private Sub ReadInstrumentError(Instrument As VisaComLib.FormattedIO488)
' Read instrument error queue until its empty.
Dim errCode As Long
Dim errMsg As String
    StatusTextBox.Text = StatusTextBox.Text & vbCrLf
    errCode = 999
    While errCode <> 0
        Instrument.WriteString "SYST:ERR?"
        errMsg = Instrument.ReadString
        errCode = Val(errMsg)
        StatusTextBox.Text = StatusTextBox.Text & "ErrorQuery: " & errCode & ", " &
errMsg & vbCrLf
    Wend
End Sub
Private Sub ExitExample_Click()
    Unload Me
End Sub
```

List Programming Example

This program demonstrates SAS mode using a list of model parameters with independent dwell time for each model. It also demonstrates the use of a custom initial model instead of the instrument default model. The following model values are used in the list:

	Voc	Isc	Vmp	Imp	Dwell
Model 1	40	2	35	1.5	0.5
Model 2	30	2	25	1	1
Model 3	30	2.2	25	1.5	1
Model 4	20	1	18	0.5	0.5

When the program runs, it displays a dialog box that will contain the measured values. Buttons on the dialog let you Start the example or Exit the example.



You have a royalty-free right to use, modify, reproduce and distribute this example program (and/or any modified version) in any way you find useful, provided you agree that Agilent Technologies has no warranty, obligations, or liability for any example programs.

This program is written in Microsoft Visual Basic 6.0 using the VISA IO library. The VISA library must be downloaded from the Agilent IO Libraries Suite CD-ROM to use this program. For information about using VISA in another Visual Basic project, refer to “Programming Your Instruments” in the USB/LAN/GPIB Interfaces Connectivity Guide, also included on the Agilent IO Libraries Suite CD-ROM.

5 Programming Examples

Option Explicit

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub StartExample_Click()

,

' Agilent E436xA Solar Array Simulator VB6 Example Program

,

' This program demonstrates the use of a list to change the model parameters in SAS mode.

,

' variable declarations

,

Dim IDN As String

Dim DeviceAddress As String

Dim ErrString As String

Dim CmdString As String

Dim ChannelList As String

Dim RespBuf As String

Dim StatusCode As Integer

,

' bit position of WTG_tran

,

Const TRAN_WTG_MASK As Integer = 32

Const MIN_SAS_OPC_TIME As Integer = 250

,

' defined the lists

,

Const VocList = "40, 30, 30, 20" ' volts

Const IscList = "2, 2, 2.3, 1" ' amperes

Const VmpList = "35, 25, 25, 18" ' volts

Const ImpList = "1.5, 1, 1.5, 0.5" ' amperes

Const DwellList = "0.5, 1, 1, 0.5" ' seconds

On Error GoTo ErrorHandler

'These variable are necessary to initialize the VISA COM.

Dim ioMgr As AgilentRMLib.SRMClS

Dim Instrument As VisaComLib.FormattedIO488

' The following command line provides the program with the VISA name of the

' interface that it will communicate with. It is currently set to use GPIB.

DeviceAddress = "GPIB0::5::INSTR"

' Use the following line for LAN communication using an IP address

' DeviceAddress="TCPIP0::141.25.36.214::INSTR"

' Use the following line for LAN communication using a Hostname

```

' DeviceAddress="TCPIP0::a-e4360a-00018::INSTR"
' Use the following line for USB communication
' DeviceAddress = "USB0::2391::4359::US43600018"
' Initialize the VISA COM communication
Set ioMgr = New AgilentRMLib.SRMClS
Set Instrument = New VisaComLib.FormattedIO488
Set Instrument.IO = ioMgr.Open(DeviceAddress)
' send power-on reset to the instrument
Instrument.WriteString "*RST"
'
Instrument.WriteString "*IDN?"
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
'
ChannelList = "@1" ' channel 1 only
' setup the lists
Instrument.WriteString "LIST:SAS:VOC " & VocList & "," & ChannelList
Instrument.WriteString "LIST:SAS:ISC " & IscList & "," & ChannelList
Instrument.WriteString "LIST:SAS:VMP " & VmpList & "," & ChannelList
Instrument.WriteString "LIST:SAS:IMP " & ImpList & "," & ChannelList
Instrument.WriteString "LIST:DWEL " & DwellList & "," & ChannelList
' other list parameters
Instrument.WriteString "LIST:STEP AUTO," & ChannelList
Instrument.WriteString "LIST:COUN 1," & ChannelList
' setup initial model parameters
CmdString = "VOLT:SAS:VOC 10," & ChannelList
CmdString = CmdString & ";VMP 9," & ChannelList
CmdString = CmdString & ";CURR:SAS:ISC 1," & ChannelList
CmdString = CmdString & ";IMP 0.9," & ChannelList
Instrument.WriteString CmdString
' SAS mode parameters
Instrument.WriteString "CURR:MODE:DTAB 4096," & ChannelList
Instrument.WriteString "CURR:MODE SAS," & ChannelList
Instrument.WriteString "CURR:SAS:MODE LIST," & ChannelList
' set up a trigger source (not necessary if using software trigger)
Instrument.WriteString "TRIG:TRAN:SOUR PIN3" & ChannelList
' enable the output
Instrument.WriteString "OUTP:STAT ON," & ChannelList
' check output voltage
Sleep MIN_SAS_OPC_TIME
Instrument.WriteString "MEAS:VOLT? " & ChannelList

```

5 Programming Examples

```
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
' INITiate the output sub-system
Instrument.WriteString "INIT:TRAN " & ChannelList
' check if the system is ready for trigger (to start)
StatusCode = 0
While StatusCode <> TRAN_WTG_MASK
    Instrument.WriteString "STAT:OPER:COND? " & ChannelList
    RespBuf = Instrument.ReadString
    StatusCode = Val(RespBuf) And TRAN_WTG_MASK
Wend
'
' Trigger the list to start
Instrument.WriteString "TRIG:TRAN " & ChannelList
'
' measure output voltage to verify each list step
'
Sleep MIN_SAS_OPC_TIME
Instrument.WriteString "MEAS:VOLT? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
Sleep 500
Instrument.WriteString "MEAS:VOLT? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
Sleep 1000
Instrument.WriteString "MEAS:VOLT? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
Sleep 1000
Instrument.WriteString "MEAS:VOLT? " & ChannelList
StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
Sleep 500
Instrument.WriteString "MEAS:VOLT? " & ChannelList
```

```

StatusTextBox.Text = StatusTextBox.Text & Instrument.ReadString & vbCrLf
StatusTextBox.Refresh
'
' disable the output
Instrument.WriteString "OUTP:STAT OFF," & ChannelList
'
StatusTextBox.Text = StatusTextBox.Text & "List completed!!!" & vbCrLf
StatusTextBox.Refresh
'
Exit Sub
ErrorHandler:
    MsgBox "ErrorHandler:" & vbCrLf & Err.Description
    ReadInstrumentError Instrument
    Exit Sub
End Sub
Private Sub ReadInstrumentError(Instrument As VisaComLib.FormattedIO488)
' Read instrument error queue until its empty.
Dim errCode As Long
Dim errMsg As String
    StatusTextBox.Text = StatusTextBox.Text & vbCrLf
    errCode = 999
    While errCode <> 0
        Instrument.WriteString "SYST:ERR?"
        errMsg = Instrument.ReadString
        errCode = Val(errMsg)
        StatusTextBox.Text = StatusTextBox.Text & "ErrorQuery: " & errCode & ", " & errMsg & vbCrLf
    Wend
End Sub
Private Sub ExitExample_Click()
    Unload Me
End Sub

```




6 SCPI Error Messages

Introduction	242
Error List	242

This appendix gives the error numbers and descriptions that are returned by the Agilent E4360.

Introduction

The front-panel ERR annunciator turns on when one or more errors are currently stored in the error queue.

Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored.

Errors are cleared as you read them. When you have read all errors from the error queue, the ERR annunciator turns off and the errors are cleared.

If more errors have occurred than can fit in the buffer, the last error stored in the queue (the most recent error) is replaced with -350,"Error queue overflow". No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the instrument responds with +0,"No error".

The error queue is cleared by the *CLS (Clear Status) command and when power is cycled. The errors are also cleared when you read the error queue. The error queue is not cleared by a Factory Reset (*RST command) or an Instrument Preset (SYSTem:PRESet command).

Front-Panel Operation:

If the ERR annunciator is on, press the Error key to view the errors. Use the navigation keys to scroll through the error list. All errors are cleared when you exit the menu.

Remote Interface Operation:

SYSTem:ERRor? Reads and clears one error from the queue

Errors have the following format (the error string may contain up to 80 characters):

-113,"Undefined header"

Error List

Error	Device-dependent Errors (these errors set Standard Event Status register bit #3)
0	No error This is the response to the ERR? query when there are no errors.
100	Too many channels You have specified more channels than are installed in the mainframe.
101	Calibration state is off Calibration is not enabled. The instrument will not accept calibration commands.
102	Calibration password is incorrect The calibration password is incorrect.
103	Calibration is inhibited by switch setting Calibration mode is locked out by the calibration switch.
104	Bad sequence of calibration commands Calibration commands have not been entered in the proper sequence.

Device-dependent Errors (continued)	
105	Unexpected output current The measured output current is outside the acceptable range.
106	Zero measurement out of range error The "zero" measurement value is outside the acceptable range.
107	Programming cal constants out of range The programmed calibration constant is outside the acceptable range.
108	Measurement cal constants out of range The measurement calibration constant is outside the acceptable range.
109	Over voltage cal constants out of range The over voltage calibration constant is outside the acceptable range.
110	Wrong V+I The instrument was unable to set the correct voltage or current value.
111	Aux vloc cal constants out of range Calibration constants on the internal auxiliary local ADC are outside the acceptable range.
112	Aux vrem cal constants out of range Calibration constants on the internal auxiliary remote ADC are outside the acceptable range.
113	Aux imon cal constants out of range Calibration constants on the internal auxiliary imon ADC are outside the acceptable range.
114	Wrong status An incorrect status function has been reported.
116	Locked out by internal switch setting This function has been locked out by an internal switch in the mainframe.
117	Calibration error A calibration error has occurred. Do not save calibration constants. Try re-calibrating the unit.
118	Must be in Fixed mode to calibrate The instrument must be set to Fixed mode operation before it can be calibrated.
200	Hardware error channel <channel> A hardware error has occurred on the specified channel.
201	Invalid configuration, empty slots There is an empty slot between modules. This configuration is not allowed.
202	Selftest Fail A selftest failure has occurred. See selftest failure list for details.
204	NVRAM checksum error A checksum error has occurred in the instrument's nonvolatile random access memory.
205	NVRAM full The nonvolatile random access memory of the instrument is full.
206	File not found The internal calibration file or the internal channel attribute file was not found in NVRAM.
207	Cal file version error The calibration file was written or read using old firmware. Firmware must be updated.
208	Running backup firmware The instrument is presently running the backup (previous) version of the firmware.
209	Modules do not match group NV config One or more modules that have their output coupled do not match the instrument state settings.
210	Frame NVRAM error A non-volatile RAM error has occurred in the mainframe

Device-dependent Errors (continued)	
302	Option not installed The option that is programmed by this command is not installed.
303	There is not a valid acquisition to fetch from There is no valid data in the measurement buffer.
304	Volt and curr in incompatible transient modes Voltage and current cannot be in Step and List mode at the same time.
305	A triggered value is on a different range A triggered value is on a different range than the one that is presently set.
306	Too many list points Too many list points have been specified.
307	List lengths are not equivalent One or more lists are not the same length.
308	This setting cannot be changed while transient trigger is initiated Setting cannot be changed while the instrument is waiting for or executing a trigger sequence.
309	Cannot initiate, voltage and current in fixed mode Cannot initiate transient generator. Either the voltage or current function is set to Fixed mode.
311	This group of channel has non-identical features One or more modules in this group does not match the ratings or options of the other modules.
312	The command is disabled when the channel is grouped This command is disabled when the output channel is grouped.
313	Datalog time interval out of range The datalog time interval is set to a value that is outside the allowable range.
315	Settings conflict error A data element could not be programmed because of the present instrument state.
316	Mass storage error An error has occurred with the mass storage. For example, the mass storage could be full.
318	Datalog error An error has occurred with the datalogger.
320	VMP must be less than VOC The value of the maximum-point voltage must be less than the value of the open-circuit voltage.
321	IMP must be less than or equal to ISC The value of the maximum-point current must be less than or equal to the value of the short-circuit current.
322	VMP and/or IMP too small The values of either the maximum-point voltage or the maximum-point current are too small.
323	Can't init unless output in SAS mode with List submode selected The instrument cannot be initialized unless it is set to SAS mode with List mode selected.
324	Cannot change resolution unless in FIXEd mode The instrument must be set to Fixed mode operation before the resolution can be changed.
325	Too many table points Too many table points have been specified. The maximum is 4000.
326	Table not found A table with that name has not been found.
327	Cannot de-select table in table mode The table cannot be de-selected because the instrument is operating in Table mode. Set the instrument to Fixed mode to de-select the table.

Device-dependent Errors (continued)	
328	Calculated open circuit voltage too large The calculated open-circuit voltage exceeds the output voltage rating of the instrument.
329	Exceeds soft limit The programmed value exceeds the soft limit setting.
330	Table error An error has occurred because of one of the table value parameters.
331	Exceeds low-line limit The programmed current exceeds the reduced current that is available when the instrument is operating at low AC line (100 VAC or 120 VAC nominal).
400	Power must be cycled for new settings to take effect Cycle AC power for the new settings to take effect.
401	I/O settings have been reset to factory defaults. Power must be cycled for settings to take effect. Cycle AC power for the new I/O settings to take effect.
402	NVRAM settings have been reset to factory defaults. Power must be cycled for settings to take effect Cycle AC power for the new NVRAM settings to take effect.
414	Settings conflict warning There is a settings conflict error. For example, the Vmp setting may be greater than the Voc setting. This is not allowed.
Command Errors (these errors set Standard Event Status register bit #5)	
-100	Command error Generic syntax error.
-101	Invalid character An invalid character was found in the command string.
-102	Syntax error Invalid syntax was found in the command string. Check for blank spaces.
-103	Invalid separator An invalid separator was found in the command string. Check for proper use of , ; :
-104	Data type error A different data type than the one allowed was found in the command string.
-105	GET not allowed A group execute trigger is not allowed in a command string.
-108	Parameter not allowed More parameters were received than were expected.
-109	Missing parameter Fewer parameters were received than were expected.
-110	Command header error An error was detected in the header.
-111	Header separator error A character that was not a valid header separator was found in the command string.
-112	Program mnemonic too long The header contains more than 12 characters.
-113	Undefined header A command was received that was not valid for this instrument.
-114	Header suffix out of range The value of the numeric suffix is not valid.

Command Errors (continued)	
-120	Numeric data error Generic numeric data error.
-121	Invalid character in number An invalid character for the data type was found in the command string.
-123	Exponent too large The magnitude of the exponent was larger than 32000.
-124	Too many digits The mantissa of a numeric parameter contained more than 255 digits, excluding leading zeros.
-128	Numeric data not allowed A numeric parameter was received but a character string was expected.
-130	Suffix error Generic suffix error
-131	Invalid suffix A suffix was incorrectly specified for a numeric parameter.
-134	Suffix too long The suffix contains more than 12 characters.
-138	Suffix not allowed A suffix is not supported for this command.
-140	Character data error Generic character data error
-141	Invalid character data Either the character data element contains an invalid character, or the element is not valid.
-144	Character data too long The character data element contains more than 12 characters.
-148	Character data not allowed A discrete parameter was received, but a string or numeric parameter was expected.
-150	String data error Generic string data error
-151	Invalid string data An invalid character string was received. Check that the string is enclosed in quotation marks.
-158	String data not allowed A character string was received, but is not allowed for this command.
-160	Block data error Generic block data error
-161	Invalid block data The number of data bytes sent does not match the number of bytes specified in the header.
-168	Block data not allowed Data was sent in arbitrary block format but is not allowed for this command.
-170	Expression error Generic expression error
-171	Invalid expression data The expression data element was invalid.
-178	Expression data not allowed Expression data element was sent but is not allowed for this command.

Execution Errors (these errors set Standard Event Status register bit #4)	
-200	Execution error Generic syntax error
-220	Parameter error A data element related error occurred.
-221	Settings conflict A data element could not be executed because of the present instrument state.
-222	Data out of range A data element could not be executed because the value was outside the valid range.
-223	Too much data A data element was received that contains more data than the instrument can handle.
-224	Illegal parameter value An exact value was expected but not received.
-225	Out of memory The device has insufficient memory to perform the requested operation.
-226	Lists not same length One or more lists are not the same length.
-230	Data corrupt or stale Possible invalid data. A new reading was started but not completed.
-231	Data questionable The measurement accuracy is suspect.
-232	Invalid format The data format or structure is inappropriate.
-233	Invalid version The version of the data format is incorrect to the instrument.
-240	Hardware error The command could not be executed because of a hardware problem with the instrument.
-241	Hardware missing The command could not be executed because of missing hardware, such as an option.
-260	Expression error An expression program data element related error occurred.
-261	Math error in expression An expression program data element could not be executed due to a math error.

Query Errors (these errors set Standard Event Status register bit #2)	
-400	Query Error Generic error query
-410	Query INTERRUPTED A condition causing an interrupted query error occurred.
-420	Query UNTERMINATED A condition causing an unterminated query error occurred.
-430	Query DEADLOCKED A condition causing a deadlocked query error occurred.
-440	Query UNTERMINATED after indefinite response A query was received in the same program message after a query indicating an indefinite response was executed.

Selftest Errors (these errors set Standard Event Status register bit #3)	
202	Thermistor Failed <chan> The internal temperature thermistor has failed. <chan> is the channel location of the failure.
202	Mass Storage Error The user table flash memory is not initialized. This can occur if the table flash memory has been cleared or sanitized. Use SYST:MMEM:INIT to restore the table flash memory.



7

Agilent E4350 Compatibility

Programming Differences	250
---	-----

The Agilent E4360 Modular Solar Array Simulator is programmatically compatible with the Agilent E4350B/ E4351B Solar Array Simulator. This means that, with the following exceptions, you can program the Agilent E4360 SAS using the same commands that are used to program the Agilent E4350B/E4351B SAS.

NOTE

The programing differences described here are to help you convert existing programs written for the Agilent E4350B/E4351B instruments to run on the Agilent E4360 Modular SAS.

Programming Differences

The following table documents the differences between the way SCPI programming commands work on the Agilent E4360 Modular SAS and the way they work on the Agilent E4350B/E4351B SAS instruments.

Item	E4350B/E4351B	E4360A Modular SAS
Channel parameter	No channel parameter required	Specify channel 1 or 2. For compatibility with the E4350B/E4351B, you can group channels 1 and 2 so that no channel parameter is required when addressing the instrument. Refer to the User'sGuide for details.
Unique status indicators	RI, CAL, WTG	PF, PROT, OFF, WTG-meas, WTG-tran, WTG-dlog, BUSY-dlog
Max. short-circuit current	8 A or 4 A (model dependent)	8.5 A or 5 A (model dependent)
Fetch measurements	Not available	Fetch previously measured (triggered)voltage or current values
Measurement (data) logging	Not available	64K buffer size
Curve lists	Not available	Generates pre-defined I-V curves Can contain up to 512 sets of points
Soft voltage and current limit	Not available	Prevent inadvertent programming over the specified limit
Calibration	No over-current calibration	Over-current calibration
Storage states	5 in Fixed mode only	2 mainframe states
Display enable	Not available	Enable/disable front panel display
Digital IO functions	4 pins, INH, FLT, and IO functions only	8 pins, INH, FLT, IO, TrigIn, TrigOut, OnCouple, OFFCouple
Trigger sources	BUS	BUS, PIN<n>, TRAN<n>, DWEL<n>, DLOG<n>, ACQ<n>
Over-current protection	OC - independently set for SAS and Table modes OCP – tied to current setting in Fixed mode	OC – independently set for all modes
Command processing time	20 ms	1 ms

Index

A	
abort commands.....	18
ABOR ACQ.....	20
ABOR DLOG.....	21
ABOR TRAN.....	19
C	
calibrate	
password.....	27
calibrate commands	
CAL CURR.....	23
CAL CURR LEV.....	24
CAL CURR PROT.....	25
CAL DATE.....	26
CAL PASS.....	27
CAL SAVE.....	28
CAL STAT.....	29
CAL VOLT.....	31
CAL VOLT LEV.....	32
CAL VOLT PROT.....	33
channel	
parameter.....	13
range.....	13
command	
combining.....	11
completion.....	16
coupling.....	12
synchronization.....	16
common commands.....	17
*CLS.....	35
*ESE.....	36
*ESR.....	38
*IDN?.....	40
*OPC.....	41
*OPT?.....	42
*RCL.....	43
*RDT?.....	44
*RST.....	45
*SAV.....	46
*SRE.....	47
*STB.....	49
*TRG.....	51
*TST.....	52
*WAI.....	53
combining.....	11
compatibility	
differences.....	250
models.....	249
contact numbers.....	7
conventions used in this guide.....	14
coupled commands.....	12
D	
device clear.....	16
display commands	
DISP ENAB.....	57
DISP TEXT.....	55
DISP VIEW.....	56
E	
error codes.....	242
displaying.....	242
F	
fetch commands	
FETC CURR?.....	59
FETC DLOG?.....	61
FETC VOLT?.....	60
format commands	
FORM [DATA].....	64
FORM BORD.....	65
H	
headers.....	13
history.....	2
I	
initiate commands	
INIT ACQ.....	68
INIT CONT.....	70
INIT DLOG.....	69
INIT TRAN.....	67
instrument drivers.....	6
L	
list	
programming example.....	235

Index

M	
MAV	169
measure commands	
MEAS CURR?	72
MEAS VOLT?	73
memory commands	
MEM COPY TABL	75
MEM DEL ALL	76
MEM DEL NAME	77
MEM TABL CAT?	78
MEM TABL CURR	79
MEM TABL CURR POIN?	80
MEM TABL SEL	81
MEM TABL VOLT	82
MEM TABL VOLT POIN?	83
message separator	14
message terminator	
end or identify	14
newline	14
MSS	169
multipliers	15
N	
non-volatile RAM settings	229
numeric data formats	15
O	
operation status group	168
header	11
output	
programming example	232
queue	169
output commands	
OUTP	85
OUTP COUP	86
OUTP COUP CHAN	87
OUTP INH MODE	89
OUTP PON STAT	90
OUTP PROT CLE	91
OUTP PROT COUP	92
P	
print date	2
Q	
query	
description	11
indicator	13
questionable status group	168
R	
remote interface	
factory settings	229
reset states	230
root specifier	14
RQS	169
S	
SCPI	
command completion	16
common commands	34
device clear	16
header path	10
message structure	12
message unit	12
multiple commands	10
program message	12
response message	12
subsystem commands	224
sense commands	
SENS DLOG CLOC	94
SENS DLOG TINT	95
source commands	
[SOUR] CURR DTAB SAS IMP?	100
[SOUR] CURR DTAB SAS ISC?	101
[SOUR] CURR DTAB SAS LIST IMP?	103
[SOUR] CURR DTAB SAS LIST ISC?	104
[SOUR] CURR DTAB SAS LIST?	102
[SOUR] CURR DTAB SAS?	99
[SOUR] CURR DTAB TABL IMP?	106
[SOUR] CURR DTAB TABL ISC?	107
[SOUR] CURR DTAB TABL?	105
[SOUR] CURR MODE	108
[SOUR] CURR MODE DTAB	110
[SOUR] CURR MODE PROG	112
[SOUR] CURR PROT	114
[SOUR] CURR SAS IMP	116
[SOUR] CURR SAS ISC	118
[SOUR] CURR SAS MODE	120
[SOUR] CURR SLIM HIGH	121
[SOUR] CURR TABL NAME	123
[SOUR] CURR TABL OFFS	124
[SOUR] CURR?	98
[SOUR] DIG INP DATA?	125
[SOUR] DIG OUTP DATA	126

[SOUR] DIG PIN<1-7> FUNC	127
[SOUR] DIG PIN<1-7> POL	129
[SOUR] LIST COUN	130
[SOUR] LIST DWEL	131
[SOUR] LIST DWEL POIN?	133
[SOUR] LIST SAS IMP	134
[SOUR] LIST SAS IMP POIN?	136
[SOUR] LIST SAS ISC	137
[SOUR] LIST SAS ISC POIN?	139
[SOUR] LIST SAS VMP	140
[SOUR] LIST SAS VMP POIN?	142
[SOUR] LIST SAS VOC	143
[SOUR] LIST SAS VOC POIN?	145
[SOUR] LIST STEP	146
[SOUR] LIST STEP ACT?	147
[SOUR] LIST TERM LAST	148
[SOUR] VOLT [IMM]	149
[SOUR] VOLT DTAB SAS LIST VMP?	154
[SOUR] VOLT DTAB SAS LIST VOC?	155
[SOUR] VOLT DTAB SAS LIST?	153
[SOUR] VOLT DTAB SAS VMP?	151
[SOUR] VOLT DTAB SAS VOC?	152
[SOUR] VOLT DTAB SAS?	150
[SOUR] VOLT DTAB TABL VMP?	157
[SOUR] VOLT DTAB TABL VOC?	158
[SOUR] VOLT DTAB TABL?	156
[SOUR] VOLT PROT	159
[SOUR] VOLT SAS VMP	161
[SOUR] VOLT SAS VOC	163
[SOUR] VOLT SLIM HIGH	165
[SOUR] VOLT TABL OFFS	167
standard event status group	168
status byte	169
status commands	
STAT OPER COND?	173
STAT OPER ENAB	175
STAT OPER NTR	177
STAT OPER PTR	179
STAT OPER?	171
STAT PRES	181
STAT QUES COND?	184
STAT QUES ENAB	186
STAT QUES NTR	188
STAT QUES PTR	190
STAT QUES?	182
subsystem commands	17
suffixes	15
system commands	
SYST CHAN	193
SYST CHAN MOD?	194
SYST CHAN OPT?	195
SYST CHAN SER?	196
SYST COMM RLST	197
SYST ERR?	198
SYST GRO CAT?	199
SYST GRO DEF	200
SYST GRO DEL	201
SYST GRO DEL ALL	202
SYST GRO PAR	203
SYST MMEM INIT	204
SYST MMEM SAN	205
SYST PASS FPAN RES	206
SYST REB	207
SYST VERS?	208
T	
trigger commands	
TRIG ACQ	213
TRIG ACQ DEL	214
TRIG ACQ SOUR	215
TRIG DLOG	217
TRIG DLOG SOUR	218
TRIG PIN<3-7> SOUR	220
TRIG TRAN	210
TRIG TRAN SOUR	211
types of SCPI commands	10
W	
Web server	6
Web URL's	6, 7