Autoranging DC Power Supplies

E36150 Series

This manual contains SCPI programming information which can be used to program the E36150 Series autoranging DC power supplies.

KEYSIGHT

PROGRAMMING GUIDE

Notices

Copyright Notice

Manual Part Number

E36151-90008

Edition

Edition 2, November 2023

Published by

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

Warranty

Technology Licenses

U.S. Government Rights

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

Third Party Licenses

Portions of this software are licensed by third parties including open source terms and conditions. To the extent such licenses require that Keysight make source code available, we will do so at no cost to you. For more information, please contact Keysight support at https://www.keysight.com/find/assist.

Waste Electrical and Electronic Equipment (WEEE)

This product complies with the WEEE Directive) marking requirement. The affixed product label (see below) indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category: With reference to the equipment types in the WEEE directive Annex 1, this product is classified as “Monitoring and Control instrumentation” product. Do not dispose in domestic household waste.

To return unwanted products, contact your local Keysight office, or see about.keysight.com/en/companyinfo/environment/takeback.shtml for more information.

Declarations of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://regulations.corporate.keysight.com/DoC/search.htm and click on “Declarations of Conformity.” You can then search by product number to find the latest Declaration of Conformity.

Safety Information

CAUTION

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

WARNING

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

1 Remote Operation

This chapter describes the remote operation for the E36150 Series autoranging DC power supply.

Introduction to the SCPI Language

Standard Commands for Programmable Instruments (SCPI) is an ASCII-based instrument command language designed for test and measurement instruments. Refer to Simplified Programming Overview for basic techniques for programming the instrument over the remote interface.

SCPI commands are based on a hierarchical structure, also known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems. A portion of the SOURce subsystem is shown below to illustrate the tree system.

```
[SOURce:]
        CURRent <current> | MIN | MAX
        CURRent? MIN | MAX
        CURRent:
                TRIGgered <current> | MIN | MAX
                TRIGgered? MIN | MAX
        VOLTage <voltage> | MIN | MAX
        VOLTage? MIN|MAX
        VOLTage:
                TRIGgered <voltage> | MIN | MAX
                TRIGgered? MIN | MAX
```

SOURce is the root keyword of the command, CURRent and VOLTage are second-level keywords, and TRIGgered is the third-level keyword. A colon (:) separates a command keyword from a lower-level keyword.

Command Format Used in this Manual

The format used to show commands in this manual is shown below:

```
CURRent <current> | MINimum | MAXimum
```

The command syntax shows most commands (and some parameters) as a mixture of upper-case and lower-case letters. The upper-case letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, send the long form.

For example, in the above syntax statement, CURR and CURRENT are both acceptable forms. You can use upper-case or lower-case letters. Therefore, CURRENT, curr, and Curr are all acceptable. Other forms, such as CUR and CURREN, will generate an error.

A vertical bar (|) separates multiple parameter choices for a given command string.

Angle brackets (<>) indicate that you must specify a value for the enclosed parameter. For example, the above syntax statement shows the current parameter enclosed in angle brackets. The brackets are not sent with the command string. You must specify a value for the parameter (such as CURR 0.1).

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

Some parameters are enclosed in square brackets ([]). The brackets indicate that the parameter is optional and can be omitted. The brackets are not sent with the command string. If you do not specify a value for an optional parameter, the instrument chooses a default value.

A colon (:) separates a command keyword from a lower-level keyword. You must insert a blank space to separate a parameter from a command keyword. If a command requires more than one parameter, you must separate adjacent parameters using a comma as shown below:

```
APPLy MAX,MAX
```

Command Separators

A colon (:) separates a command keyword from a lower-level keyword as shown below:

```
SOURce:CURRent:TRIGgered
```

A semicolon (;) is used to separate two commands within the same subsystem, and can also minimize typing. For example, sending the following command string:

```
SOUR:VOLT MIN;CURR MAX
```

is the same as sending the following two commands:

```
SOUR:VOLT MIN
```

```
SOUR:CURR MAX
```

Use a colon and a semicolon to link commands from different subsystems. For example, in the following command string, an error is generated if you do not use the colon and semicolon:

```
DISP:TEXT:CLE;:SOUR:CURR MIN
```

Using the MIN and MAX parameters

You can substitute MINimum or MAXimum in place of a parameter for many commands. For example, consider the following command:

CURRent <current> | MIN | MAX

Instead of selecting a specific current, you can substitute MINimum to set the current to its minimum value or MAXimum to set the current to its maximum value.

Querying Parameter Settings

You can query the value of most parameters by adding a question mark (?) to the command. For example, the following command sets the output current to 5 A:

CURR 5

You can query the value by executing:

CURR?

You can also query the maximum or minimum value allowed with the present function as follows:

CURR? MAX; CURR? MIN

CAUTION **If you send two query commands without reading the response from the first, and then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query command without reading the response. When you cannot avoid this situation, send a device clear before sending the second query command.**

SCPI Command Terminators

A command string sent to the instrument must terminate with a <new line> character. The IEEE-488 EOI (end-or-identify) message is interpreted as a <new line> character and can be used to terminate a command string in place of a <new line> character. A <carriage return> followed by a <new line> is also accepted. Command string termination will always reset the current SCPI command path to the root level. The <new line> character has the ASCII decimal code of 10.

IEEE-488.2 Common Commands

The IEEE-488.2 standard defines a set of common commands that perform functions like reset, self-test, and status operations. Common commands always begin with an asterisk (*), are four to five characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a blank space. Use a semicolon (;) to separate multiple commands as shown below:

*RST; *CLS; *ESE 32; *OPC?

SCPI Parameter Types

The SCPI language defines several different data formats to be used in program messages and response messages.

Channel parameters

Channel parameter <chanlist> is required to address one or more channels. It has the following syntax:

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

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

(@<start_channel>:<end_channel>)

For example, (@1) specifies channel 1. The channel list, shown as <chanlist> throughout this document, must be preceded with the @ symbol and must be enclosed in parentheses (). 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.**

Numeric parameters

Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters like MINimum, MAXimum, and DEFault are also accepted.

You can also send engineering unit suffixes (V, A, or SEC) with numeric parameters. If only specific numeric values are accepted, the power supply will automatically round the input numeric parameters. The following command uses a numeric parameter:

```
CURR <current> | MIN | MAX
```

Discrete parameters

Discrete parameters are used to program settings that have a limited number of values such as BUS and IMM. Query responses will always return the short form in all upper-case letters. The following command uses discrete parameters:

```
TRIG:SOUR BUS | IMM
```

Boolean parameters

Boolean parameters represent a single binary condition that is either true or false. For a false condition, the instrument will accept OFF or 0. For a true condition, the instrument will accept ON or 1. When you query a boolean setting, the instrument will always return 0 or 1. The following command uses a boolean parameter:

```
DISP OFF | ON
```

String parameters

String parameters can contain virtually any set of ASCII characters. A string must begin and end with matching quotes; either with a single quote or with a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between. The following command uses a string parameter:

DISP:TEXT <quoted string>

Halting an Output in Progress

You can send a device clear at any time to stop an output in progress over the GPIB interface. The status registers, the error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions.

- The instrument's input and output buffers are cleared.
- The instrument is prepared to accept a new command string.
- The following command sends a device clear over the GPIB interface using Keysight BASIC.

CLEAR 705 IEEE-488 Device Clear

Programming Ranges

The following table shows the voltage and current values that can be programmed for each model. The DEFault voltage is always 0 V.

NOTE The allowable settings for both current and voltage are 0 to maximum value. When 0 is set, it will automatically set the unit to the minimum value.

Output		E36154A	E36155A
Current	MAXimum	82.4 A	41.2 A
	MINimum	8 mA	4 mA
	DEFault (*RST)	8 A	4 A
Voltage	MAXimum	30.9 V	61.8 V
	MINimum	0 V	0 V
	DEFault (*RST)	0 V	0 V

SCPI Error Messages

The instrument returns error messages in accordance with the SCPI standard.

- Up to 20 errors can be stored in the instrument's error queue, and the ERROR annunciator turns on when one or more errors are in the error queue.
- Error retrieval is first-in-first-out (FIFO), and errors are cleared as you read them. When you have read all errors from the error queue, the **ERR** annunciator turns off.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with -350,"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".
- Send SYSTem:ERRor? to read the most recent error. Each error is in the format: -104,"Data type error".
- To read the error queue from the front panel, press **Utilities** > **Error**. If there are more than 10 errors on the display, press **Next** to scroll to the next page.
- The error queue is cleared by power cycles and *CLS., but not *RST.

Execution error codes

The instrument's error codes are listed below:

Code	Text
0000	No error This is the response to the ERR? query when there are no errors.
-100	Command error A generic syntax error.
-101	Invalid character An invalid character was found in the command string. You may have inserted a character such as #, $, or % in the command keyword or within a parameter. Example: OUTP:TRAC #ON
-102	Syntax error Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon in the command header, or before a comma. Example: VOLT:LEV ,1
-103	Invalid separator An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space – or you may have used a blank space instead of a comma. Example: TRIG:SOUR,BUS or APPL CH1 1.0 1.0
-104	Data type error The wrong parameter type was found in the command string. You may have specified a number where a string was expected, or vice versa.
-105	GET not allowed A Group Execute Trigger (GET) is not allowed within a command string.

Code	Text
-108	Parameter not allowed More parameters were received than expected for the command. You may have entered an extra parameter, or you added a parameter to a command that does not accept a parameter. Example: OUTP? 10
-109	Missing parameter Fewer parameters were received than expected for the command. You omitted one or more parameters that are required for this command. Example: APPL
-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 A command header was received which contained more than the maximum 12 characters allowed.
-113	Undefined header A command was received that is not valid for this power supply. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters. Example: TRIGG:DEL 3
-114	Header suffix out of range The numeric suffix attached to a command header is not one of the allowable values. Example: STAT:QUES:INST:ISUM4?
-118	Query not allowed
-120	Numeric data error An invalid number was specified for a numeric parameter. Example: VOLT 1.0E+320000
-121	Invalid character in number An invalid character was found in the number specified for a parameter value. Example: *ESE #B01010102
-123	Exponent too large A numeric parameter was found whose exponent was larger than 32,000.
-124	Too many digits A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.
-128	Numeric data not allowed A numeric parameter was received but a character string was expected. Example: DISP:TEXT 123
-130	Suffix error A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix or the numeric parameter does not accept a suffix. Example: TRIG:DEL 0.5 SECS
-131	Invalid suffix A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix. Example: TRIG:DEL 0.5 SECS

Code	Text
-134	Suffix too long A suffix for a numeric parameter contained too many characters.
-138	Suffix not allowed A suffix was received following a numeric parameter which does not accept a suffix. Example: STAT:QUES:ENAB 18 SEC (SEC is not a valid suffix).
-140	Character data error A generic character data error.
-141	Invalid character data Either the character data element contained an invalid character or the particular element received was not valid for the header.
-144	Character data too long The character data element contained too many characters.
-148	Character data not allowed A discrete parameter was received but a character string or a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. Example: DISP:TEXT ON
-150	String data error A generic string data error.
-151	Invalid string data An invalid character string was received. Check to see if you have enclosed the character string in single or double quotes. Example: DISP:TEXT 'ON
-158	String data not allowed A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type. Example: TRIG:DEL 'zero'
-160	Block data error A 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 A generic expression error.
-171	Invalid expression The expression data element was invalid.
-178	Expression data not allowed Expression data element was sent but is not allowed for this command.
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error

Code	Text
-200	Execution error A generic syntax error.
-201	Invalid while in local
-202	Settings lost due to rtl
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-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 numeric parameter value is outside the valid range for the command. Example: TRIG:DEL -3
-223	Too much data A character string was received but could not be executed because the string length was more than 40 characters. This error can be generated by the CALibration:STRing command.
-224	Illegal parameter value A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice. Example: DISP:STAT XYZ (XYZ is not a valid choice).
-225	Out of memory The device has insufficient memory to perform the requested operation.
-230	Data corrupt or stale Possible invalid data. A new reading was started but not completed.
-231	Data questionable The measurement accuracy is suspect.
-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.
-250	Mass storage error
-251	Missing mass storage
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full

Code	Text
-256	File name not found
-257	File name error
-258	Media protected
-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.
-270	Macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label
-274	Macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory error
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-321	Out of memory
-330	Self-test failed
-350	Queue overflow
-363	Input buffer overrun
-400	Query error A generic error query.

Code	Text
-410	Query INTERRUPTED A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power has been off, or after a *RST (reset) command has been executed.
-420	Query UNTERMINATED The power supply was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed an APPLy command (which does not generate data) and then attempted an ENTER statement to read data from the remote interface.
-430	Query DEADLOCKED A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.
-440	Query UNTERMINATED after indefinite response A query was received in the same program message after a query indicating an indefinite response was executed.
304	Volt and curr in incompatible transient modes Voltage and current cannot be in Step and List mode at the same time.
307	List lengths are not equivalent
308	This command is not allow while list is running
513	LAN invalid IP address
514	LAN duplicate IP address
515	LAN failed to renew DHCP lease
516	LAN failed to configure
517	LAN failed to initialize
518	LAN VXI-11 fault
543	Configuration mismatched
550	3.3V power lost
551	5.0V power lost
552	12V power lost
561	Analog board (CH1) does not respond
564	Analog board (CH1) over temperature
567	Analog board (CH1) command timed out
600	Analog board (CH1) failed to enter boot loader
603	Analog board (CH1) failed to comm (SEM Fail)
606	Analog board (CH1) failed to comm (UNSPECIFIED)
610	Fan test failed
611	EEPROM load failed
612	EEPROM checksum failed
613	EEPROM save failed
614	Invalid serial number

Code	Text
615	Invalid MAC address
616	Front panel does not respond
722	Multiple channels selection is not allowed during calibration Calibration can only be done one channel at a time.
729	Not allow to enable output Possible cause due to OVP/OTP/OCP.
735	Cannot change while trigger is initiated
736	Trig init disallow as channel is coupled with another channel that has trig initiated
742	Illegal operation. List system has started Stop the list before doing this operation.
744	There is not a valid acquisition to fetch from
745	Initiate with no sense functions enabled
750	USB not connected This may be due to data logger requiring a USB thumb drive to be connected.
751	USB host access failed This may be due to DUT fails to access the USB thumb drive.
752	Insufficient space in USB drive This may be due to the memory size required by data logger is larger than the available free space.
753	Data logger is running Data logger setting can't be changed while it's running.
754	Data logger do not have valid data Fetch is not allowed as data logger doesn't have valid data.
755	Measurement is not allowed while data logger running
756	Sample period cannot be more than logging duration
757	Datalog settings exceeded available memory. Settings are adjusted.
770	Too many measurement point
900	Firmware update failed

Calibration error codes

The following errors indicate failures that may occur during a calibration.

Code	Text
577	CH1 Invalid Cal data
581	Invalid state. Cal secured
582	Invalid secure code
583	Secure code too long
584	Failed to calibrate voltage DAC
585	Failed to calibrate voltage ADC
586	Failed to calibrate OVP
587	Failed to calibrate current DAC
588	Failed to calibrate current ADC
590	Invalid Calibration sequence
590	Failed to calibrate low range current
592	This action is not allowed as calibration has not completed
593	Failed to calibrate OCP

Self-test error codes

The following errors indicate failures that may occur during a self-test.

Code	Text
530	(CH1) Analog bias output 15V test failed
533	(CH1) Chan 1 Precision DAC test failed
536	(CH1) Chan 1 Auxiliary DAC test failed
539	(CH1) Incorrect model number
542	(Front panel) Incorrect model number

Licensing error codes

The following errors indicate failures that may occur during licensing.

Code	Text
617	Invalid License
618	Invalid license format
619	Invalid model number
620	Option not supported
621	License delete failed
622	Serial number change not allowed

Reset and Non-volatile Settings

The following tables show the reset and non-volatile settings. These parameters are set to the indicated default values at power-on or after *RST.

Reset (*RST) settings

The instrument's reset settings are listed below:

SCPI command	Default value
APPLy	Same as CURRent and VOLTage commands
ARB: COUNt	1
ARB:CURRent|VOLTage:EXPonential:END	MIN
ARB:CURRent|VOLTage:EXPonential:STARt	MIN
ARB:CURRent|VOLTage:EXPonential:STARt:TIMe	0
ARB:CURRent|VOLTage:EXPonential:TCONstant	1
ARB:CURRent|VOLTage:EXPonential:TIMe	1
ARB:CURRent|VOLTage:PULSe:END:TIMe	0
ARB:CURRent|VOLTage:PULSe:STARt	MIN
ARB:CURRent|VOLTage:PULSe:STARt:TIMe	0 s
ARB:CURRent|VOLTage:PULSe:TOP	MIN
ARB:CURRent|VOLTage:PULSe:TOP:TIMe	1
ARB:CURRent|VOLTage:RAMP:END	MIN
ARB:CURRent|VOLTage:RAMP:END:TIMe	0
ARB:CURRent|VOLTage:RAMP:RTIMe	1
ARB:CURRent|VOLTage:RAMP:STARt	MIN
ARB:CURRent|VOLTage:RAMP:STARt:TIMe	0
ARB:CURRent|VOLTage:SINusoid:AMPLitude	0
ARB:CURRent|VOLTage:SINusoid:FREQuency	1
ARB:CURRent|VOLTage:SINusoid:OFFSet	0
ARB:CURRent|VOLTage:STAircase:END	MIN
ARB:CURRent|VOLTage:STAircase:END:TIMe	0
ARB:CURRent|VOLTage:STAircase:NSTeps	10
ARB:CURRent|VOLTage:STAircase:STARt	MIN
ARB:CURRent|VOLTage:STAircase:STAR:TIMe	0
ARB:CURRent|VOLTage:STAircase:TIMe	1
ARB:CURRent|VOLTage:STEP:END	MIN
ARB:CURRent|VOLTage:STEP:STARt	MIN
ARB:CURRent|VOLTage:STEP:STARt:TIMe	0

SCPI command	Default value
ARB:CURRent|VOLTage:TRAPezoid:END:TIMe	0
ARB:CURRent|VOLTage:TRAPezoid:FTIMe	1
ARB:CURRent|VOLTage:TRAPezoid:RTIMe	1
ARB:CURRent|VOLTage:TRAPezoid:STARt	MIN
ARB:CURRent|VOLTage:TRAPezoid:STARt:TIMe	0
ARB:CURRent|VOLTage:TRAPezoid:TOP	MIN
ARB:CURRent|VOLTage:TRAPezoid:TOP:TIMe	1
ARB:CURRent|VOLTage:UDEFined:LEVel	MIN
ARB:FUNCtion:SHAPe	UDEF
ARB:FUNCtion:TYPE	VOLT
ARB:TERMinate:LAST	OFF
ARB:UDEFined:BOSTep	OFF
ARB:UDEFined:EOSTep	OFF
ARB:UDEFined:DWELl	0.01
CURRent	Output dependent value For E36154A (30 V) model - MAX = 82.4 A - MIN = 8 mA - DEF = 8 A For E36155A (60 V) model - MAX = 41.2 A - MIN = 4 mA - DEF = 4 A
CURRent:MODE	FIX
CURRent:PROTection:STATe	OFF
CURRent:PROTection:DELay	0.05
CURRent:PROTection:DELay:STARt	SCH
CURRent:RANGe	HIGH
CURRent:STEP	Minimum current calibration resolution
CURRent:TRIGgered	MIN (0). If no triggered level is programmed, the CURR level is returned.
DIGital:OUTPut:DATA	0
DIGital:TOUTput:BUS	FALSE
DISPlay	ON
DISPlay:TEXT	“” (empty)
LIST:COUNt	1
LIST:CURRent	MIN

SCPI command	Default value
LIST:DWELl	0.01
LIST:STEP	AUTO
LIST:TERMinate:LAST	OFF
LIST:TOUTput:BOSTep	OFF
LIST:TOUTput:EOSTep	OFF
LIST:VOLTage	MIN
INITiate:CONTinuous:TRANsient	OFF
OUTPut:STATe	OFF
OUTPut:STATe:DELay:FALL	0
OUTPut:STATe:DELay:RISE	0
OUTPut:STATe: PMODe	VOLT
SENSe:DLOG:FUNCtion:CURRent	All channels off
SENSe:DLOG:FUNCtion:MINMax	OFF
SENSe:DLOG:FUNCtion:VOLTage	All channels on
SENSe:DLOG:OFFSet	0
SENSe:DLOG:PERiod	0.1
SENSe:DLOG:TIME	30
SENSe:DLOG:TINTerval	0.1
SENSe:FUNCtion:CURRent	ON
SENSe:FUNCtion:VOLTage	ON
SENSe:SWEep:OFFSet:POINt	0
SENSe:SWEep:POINts	30
SENSe:SWEep:TINTerval	0.01
SENSe:SWEep:SCOPe:OFFSet:POINt	0
SENSe:SWEep:SCOPe:POINts	4096
SENSe:SWEep:SCOPe:TINTerval	0.00002
SYSTem:BEEPer:STATe	ON
TRIGger:ACQuire:SOURce	BUS
TRIGger:ACQuire:CURRent	0
TRIGger:ACQuire:CURRent:SLOPe	POS
TRIGger:ACQuire:VOLTage	0
TRIGger:ACQuire:VOLTage:SLOPe	POS
TRIGger:DLOG:SOURce	IMM
TRIGger:DLOG:CURRent	0
TRIGger:DLOG:CURRent:SLOPe	POS
TRIGger:DLOG:VOLTage	0

SCPI command	Default value
TRIGger:DLOG:VOLTage:SLOPe	POS
TRIGger[:TRANsient][SEQuence]:DELay	0
TRIGger[:TRANsient][SEQuence]:SOURce	BUS
VOLTage	MIN (0)
VOLTage:MODE	FIX
VOLTage:PROTection	Output dependent value (MAX)
VOLTage:SENSe	Internal (0)
VOLTage:STEP	Minimum voltage calibration resolution
VOLTage:TRIGgered	MIN (0). If no triggered level is programmed, the VOLT level is returned.

Non-volatile settings

The instrument's non-volatile settings are listed below:

SCPI command	Default value
OUTPut:STATe:COUPle:CHANnel (channel grouping)	NONE
OUTPut:STATe: INHibit:MODE (output Inhibit mode)	OFF
OUTPut:PON:STATE	*RST
DIGital:PIN<1-3>:FUNCtion (digital port function)	Digital In
DIGital:PIN<1-3>:POLarity (digital port polarity)	Positive
GPIB Address	5
SYSTem:BEEPer:STATe	ON
DISPlay:LCD:BRIGhtness	100
SYSTem:DATE	-
SYSTem:TIME	-

Non-volatile LAN settings

The instrument's non-volatile LAN settings are listed below:

SCPI command	Default value
Get IP Address	Automatic
IP Address	192.168.10.1
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.1
Obtain DNS server from DHCP	Enabled
DNS server	Blank
Host name	K-x-xxxxx
Dynamic DNS naming service	Enabled
Domain name	Blank
Web password	Keysight

SCPI Status Registers

This section provides a detailed description of the individual registers and register groups. The status diagram at the end of this topic shows how the status registers and groups are interconnected.

Status Registers

The Operation and Questionable status groups use four different type of registers to track qualify, flag, and enable instrument events.

- A Condition register continuously monitors the state of the instrument. The bits in the condition register are updated in real time and the bits are not latched or buffered
- An PTR/NTR register qualifies the signal that passes to the event register. When a PTR bit is set, signals with positive edge transition pass to the event register. When an NTR bit is set, signals with a negative edge transition pass to the event register. When both bits are set, all signal pass. When neither bits are set, no signals pass.
- An Event register latches the various events from the condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read-only register
- An Enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register.

To program individual bits in any register group, you must send a value that corresponds to the binary-weighted value of all the bits that you wish to enable. For example, to enable bit 2 (decimal value = 4) and bit 4 (decimal value = 16), the corresponding decimal value would be 20 (4 + 16).Similarly, any register queries return the binary-weighted value of the bits that have been set. For example, with bit 3 (value 8) and bit 5 (value 32) being set, the query returns +40.

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. Refer to Status Registers for a description of each register. The following table describes the bit assignments.

Bit	Bit Name	Decimal Value	Definition
0	CV	1	Output is in constant voltage.
1	CC	2	Output is in constant current.
2	CP	4	Output is in constant power mode.
3	CCN	8	Output is limited by negative constant current.
4	not used	not used	0 is returned.
5	not used	not used	0 is returned.
6	WTG MEAS	64	Measurement system is waiting for a trigger.
7	WTG Tran	128	Transient system is waiting for a trigger.
8	WTG Dlog	256	Data logging system is waiting for trigger.
9	MEAS Active	512	Transient system is initiated.
10	TRAN Active	1024	Transient system is initiated.
11	DLOG Active	2048	Data logging system is initiated.
12-15	not used	not used	0 is returned.

Questionable Status Group

These register groups record signals that indicate abnormal operation. The group consists of a Condition, PTR/NTR, Event, and Enable register. The outputs of the Questionable Status group are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register. Refer to Status Registers for a description of each register. The following table describes the bit assignments.

Bit	Bit Name	Decimal Value	Definition
0	OV	1	Output is disabled by the over-voltage protection.
1	OC	2	Output is disabled by the over-current protection.
2	not used	not used	0 is returned
3	not used	not used	0 is returned
4	OT	16	Over-temperature protection has tripped.
5	UNR	32	Output is unregulated.
6	INH	64	Output is inhibited by an external INHibit signal.
7	IPKP	128	Output is protected by positive peak current.
8	IPKN	256	Output is protected by neagtive peak current.
9-15	not used	not used	0 is returned

Standard Event Status Group

These registers are programmed by Common commands. The group consists of an Event and Enable register.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. Refer to Status Registers for a description of each register. The following table describes the bit assignments.

Bit	Bit Name	Decimal Value	Definition
0	Operation Complete	1	All commands before 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, a new command line was received before a previous query has been read, or both input and output buffers are full.
3	Device-Specific Error	8	A device-specific error, including a selftest error, calibration error or other device-specific error occurred. See Error Messages.
4	Execution Error	16	An execution error occurred. See Error Messages.
5	Command	32	A command syntax error occurred. See Error Messages.
6	not used	not used	0 is returned
7	Power On	128	Power has been cycled since the last time the event register was read or cleared.

Status Byte Register

This register summarizes the information from all other status groups as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The following table describes the bit assignments.

Bit	Bit Name	Decimal Value	Definition
0	not used	not used	0 is returned
1	not used	not used	0 is returned
2	Error Queue	4	One or more errors in the Error Queue. Use SYSTem:ERRor? to read and delete errors.
3	Questionable Status Summary	8	One or more bits are set in the Questionable Data Register. Bits must be enabled, see STATus:QUEStionable:ENABle.
4	Message Available	16	Data is available in the instrument's output buffer.
5	Event Status Summary	32	One or more bits are set in the Standard Event Register. Bits must be enabled, see *ESE.
6	Master Status Summary	64	One or more bits are set in the Status Byte Register and may generate a Service Request. Bits must be enabled, see *SRE.
7	Operation Status Summary	128	One or more bits are set in the Operation Status Register. Bits must be enabled, see STATus:OPERation:ENABle.

MSS and RQS Bits

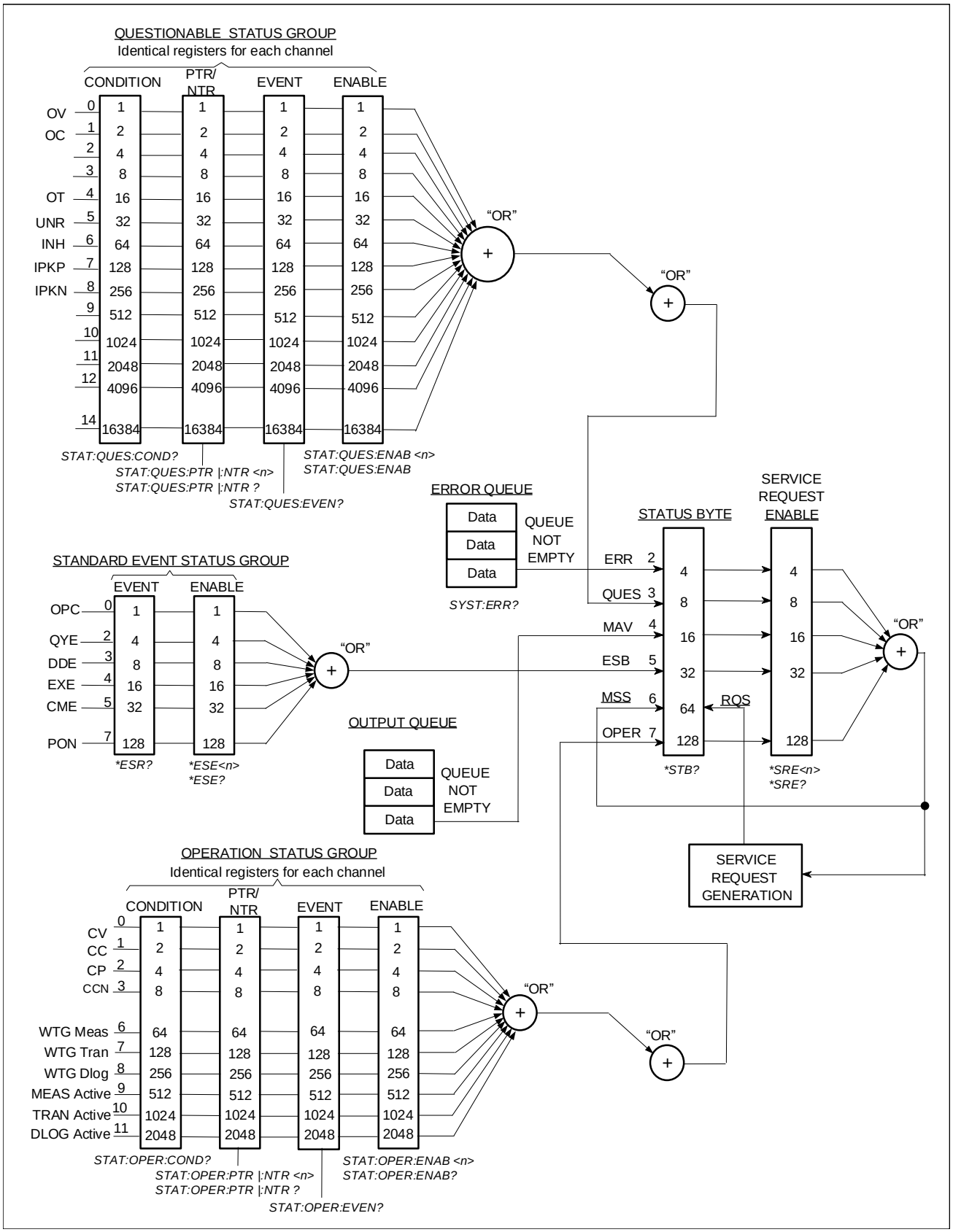
MSS is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the instrument has one or more reasons for requesting service. *STB? reads the MSS in bit position 6 of the response but does not clear any of the bits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

Error and Output Queues

The Error Queue is a first-in, first-out (FIFO) data register that stores numerical and textual description of an error or event. Error messages are stored until they are read with SYSTem:ERRor? If the queue overflows, the last error/event in the queue is replaced with error -350,"Queue overflow".

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 messages, it sets the MAV bit (4) of the Status Byte register.


QUESTIONABLE STATUS GROUP
Identical registers for each channel
CONDITION
PTR/ NTR
EVENT
ENABLE
OV
OC
OT
UNR
INH
IPKP
IPKN
"OR"
"OR"
STAT:QUES:COND?
STAT:QUES:PTR |:NTR <n>
STAT:QUES:PTR |:NTR ?
STAT:QUES:EVEN?
STAT:QUES:ENAB <n>
STAT:QUES:ENAB
ERROR QUEUE
Data
Data
Data
QUEUE NOT EMPTY
SYST:ERR?
STATUS BYTE
SERVICE REQUEST ENABLE
STANDARD EVENT STATUS GROUP
EVENT
ENABLE
OPC
QYE
DDE
EXE
CME
PON
"OR"
*ESR?
*ESE<n>
*ESE?
ERR
QUES
MAV
ESB
MSS
OPER
RQS
"OR"
*STB?
*SRE<n>
*SRE?
OUTPUT QUEUE
Data
Data
Data
QUEUE NOT EMPTY
SERVICE REQUEST GENERATION
OPERATION STATUS GROUP
Identical registers for each channel
CONDITION
PTR/ NTR
EVENT
ENABLE
CV
CC
CP
CCN
WTG Meas
WTG Tran
WTG Dlog
MEAS Active
TRAN Active
DLOG Active
"OR"
"OR"
STAT:OPER:COND?
STAT:OPER:PTR |:NTR <n>
STAT:OPER:PTR |:NTR ?
STAT:OPER:EVEN?
STAT:OPER:ENAB <n>
STAT:OPER:ENAB?

2 SCPI Programming

This chapter describes the subsystem commands available to the E36150 Series autoranging DC power supply.

ABORt:ACQuire [(@<chanlist>)]

The command cancels any pending triggered measurements and returns the trigger system to idle. This command also resets the WTG-meas bit in the Operation Condition Status register.

Parameter	Typical return
(none)	(none)
Cancels the triggered measurement: ABOR:ACQ (@1)	

Remarks

- ABORt:ACQuire is also executed at power-on and upon execution of the *RST command.

ABORt:DLOG

The command stops the current data logging session similar to pressing the List Run/Stop key on the instrument.

Parameter	Typical return
(none)	(none)
Stops the current data logging session: ABOR:DLOG	

ABORt:ELOG

NOTE In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.

The command stops the external data logging.

Parameter	Typical return
(none)	(none)
Stops the external data logging: ABOR:ELOG	

ABORt:TRANsient [(@<chanlist>)]

The command cancels any pending triggered actions and returns the trigger system to idle. This command also resets the WTG-tran bit in the Operation Condition Status register.

Parameter	Typical return
(none)	(none)
Cancels the triggered action: ABOR:TRAN (@1)	

Remarks

- ABORt:TRANsient is also executed at power-on and upon execution of the *RST command.
- Abort:TRANsient does not turn off continuous triggers if INITiate:CONTinuous:TRANsient ON has been programmed. In this case, the trigger system will automatically re-initiate.

APPLy [<voltage> | DEFault | MINimum | MAXimum [,<current> | DEFault | MINimum | MAXimum]]
APPLy?

The command is combination of [SOURce:]VOLTage, and [SOURce:]CURRent commands. The values of the voltage and current will change as soon as the command is executed.

You can substitute “MINimum”, “MAXimum”, or “DEFault” in place of a specific value for the voltage and current parameters. MIN selects the lowest voltage and current values allowed. MAX selects the highest voltage and current values allowed. The default voltage values are 0 volts for all outputs. The default current values are 8 A for the 30 V model and 4 A for the 60 V model. The default voltage and current values are exactly the same as the *RST values. If you specify only one value for the parameter, the power supply regards it as voltage setting value.

The query returns the power supply's present voltage and current values for the output as a quoted string, as shown in the sample string below (the quotation marks are returned as part of the string).

"5.00000,1.00000"

Referring to the above string, the first number 5.00000 is the voltage limit value and the second number 1.00000 is the current limit value for the output.

Parameter	Typical return
<voltage> | DEF | MIN | MAX, <current> | DEF | MIN | MAX (Values are range and product-model dependent. See Programming Range) *RST<DEF in Range of values>	<voltage>, <current>
Sets the maximum voltage and current : APPL MAX, MAX	

ARB Subsystem

ARB commands program the arbitrary waveform generator. They program pre-defined, user-defined, and sequenced waveforms.

NOTE The Arb commands only applies to E36150 Series with Option E36150ADVU except for User-Defined command. While the Constant-Dwell command is only available with Option E36150ATMU.

Constant-Dwell

Exponential

Pulse

Ramp

Sinusoid

Staircase

Step

Trapezoid

User-Defined

ARB Sequence

[SOURce:]ARB:COUNt <*count*> | MINimum | MAXimum | INFinity[, (@<chanlist>)]
[SOURce:]ARB:COUNt? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the number of times that the arbitrary waveform is repeated. The repeat count range is 1 through 9999. With Option E36150ADVU, the repeat count is 1 through > 16 million.

Parameter	Typical Return
1 – 9999 | MIN | MAX | INFinity	<Arb count>
Option E36150ADVU 1 – 16,777,216 | MIN | MAX | INFinity	
*RST 1	
Sets the Arb count to 10: ARB:COUN 10, (@1)	

Remarks

- Use the INFinity parameter to repeat the arbitrary waveform continuously. If MAX, or a value greater than 9999 or 16,777,216 is programmed, the Arb will also repeat continuously.
- Use ABORt:TRANsient to stop the Arb at any time. When the Arb is aborted, the output returns to the settings that were in effect before the Arb started.
- If a count of 9.9E37 is returned, it means that the Arb is set to repeat continuously.
- This command replaces the previous LIST:COUNt command and should be used in new applications. LIST:COUNt is still available for backward compatibility.

[SOURce:]ARB[:CURRent|:VOLTage]:CONVert

The command converts the specified arbitrary waveform to a list of points in the user-defined current or voltage Arb types. This makes it possible to modify/edit one of the "standard" arbitrary waveforms. The command performs the same function as the "Edit Points" softkey on the front panel Arb main menu.

To use this command you must first specify an Arb shape and type using the ARB:FUNCtion:SHAPe and the ARB:FUNCtion:TYPE commands.

Parameter	Typical Return
(none)	(none)
The following commands convert the ARB previously specified: ARB:CURR:CONV ARB:VOLT:CONV	

Remarks

- If the Arb function is set to UDEFined, the command is ignored.

[SOURce:]ARB:FUNCtion:SHAPe STEP | RAMP | STAircase | SINusoid | PULSe | TRAPezoid | EXPonential | UDEFined | CDWell | SEQuence[, (@<chanlist>)]
[SOURce:]ARB:FUNCtion:SHAPe? [(@<chanlist>)]

The command sets the function of the arbitrary waveform generator as follows:

Shape	Descriptions
STEP	Specifies a step (Option E36150ADVU)
RAMP	Specifies a ramp (Option E36150ADVU)
STAircase	Specifies a staircase (Option E36150ADVU)
SINusoid	Specifies a sine wave (Option E36150ADVU)
PULSe	Specifies a pulse (Option E36150ADVU)
TRAPezoid	Specifies a trapezoid (Option E36150ADVU)
EXPonential	Specifies a exponential waveform (Option E36150ADVU)
UDEFined	Specifies a user-defined waveform (List)
CDWell	Specifies a constant-dwell waveform (Option E36150ATMU)
SEQuence	Specifies a sequence of arbitrary waveforms (Option E36150ADVU)

Parameter	Typical Return
STEP | RAMP | STAircase | SINusoid | PULSe | TRAPezoid | EXPonential | UDEFined | CDWell | SEQuence	STEP, RAMP, STA, SIN, PULS, TRAP, EXP, UDEF, CDW, or SEQ
*RST UDEFined	
Specifies a step: ARB:FUNC:SHAP STEP	

Remarks

- Select the VOLTage:MODE Arb parameter to enable the instrument to generate arbitrary voltage waveforms when the transient system is initiated and triggered.
- Select the CURRent:MODE Arb parameter to enable the instrument to generate arbitrary current waveforms when the transient system is initiated and triggered.

[SOURce:]ARB:FUNCtion:TYPE CURRent | VOLTage[, (@<chanlist>)]
[SOURce:]ARB:FUNCtion:TYPE? [(@<chanlist>)]

The command selects either a current or voltage Arb. Only one type of Arb may be output at a time.

Parameter	Typical Return
CURRent | VOLTage	<type>
*RST VOLTage	
Specifies an Arb type: ARB:FUNC:TYPE CURR	

[SOURce:]ARB:TERMinate:LAST ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]ARB:TERMinate:LAST? [(@<chanlist>)]

The command determines the output value when the ARB terminates. The state is either ON (1) or OFF (0).

State	Description
ON	The output remains at the last Arb sequence value. The last voltage or current Arb value becomes the IMMediate value when the Arb completes.
OFF	The output returns to the settings in effect before the Arb started. This also applies when the Arb is aborted.

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1
*RST OFF	
Sets the ARB to terminate with the voltage or current set at the last Arb value: ARB:TERM:LAST ON, (@1)	

Remarks

- The query returns 0 if the output returns to the settings that were in effect before the Arb started, and 1 if the output remains at the last Arb value.
- This command replaces the previous LIST:TERMinate:LAST. LIST:TERMinate:LAST is still available for backward compatibility.

Constant-Dwell (CD) (Option E36150ATMU)

Back to ARB Commands

[SOURce:]ARB:CURRent:CDWell[:LEVel] <*value*>{<,*value*>},(@<chanlist>)
[SOURce:]ARB:CURRent:CDWell[:LEVel] <block>{<,*block*>}
[SOURce:]ARB:CURRent:CDWell[:LEVel]? (@<chanlist>)
[SOURce:]ARB:VOLTage:CDWell[:LEVel] <*value*>{<,*value*>},(@<chanlist>)
[SOURce:]ARB:VOLTage:CDWell[:LEVel] <block>{<,*block*>}
[SOURce:]ARB:VOLTage:CDWell[:LEVel]? (@<chanlist>)

The command specifies the level of each current or voltage point in a constant-dwell Arb.

You can only program one constant-dwell Arb type at a time, so setting the current CD Arb resets the voltage Arbs to 1 point at the minimum value. A comma-delimited list of up to 10,240 points may be programmed.

Parameter	Typical Return
minimum - maximum (Values are range and product-model dependent. See Programming Range)	<value> [,<value>] or <block> [,<block>]
*RST 1 point set to the minimum value.	
Program five current points: ARB:CURR:CDW 5,4,3,2,1, (@1) Program five voltage points: ARB:VOLT:CDW 20,21,22,23,24, (@1)	

Remarks

- For better performance, the list of points can be sent as a definite length binary block instead of an ASCII list.
- The return format depends on the settings of the FORMat:BORDer and FORMat[:DATA] commands. When the data format is set to ASCII, returned values are comma separated. Only one channel can be queried when the format is ASCII. When the data format is set to REAL, data is returned as single precision floating point numbers in definite length arbitrary block response format. Each block contains all the records for one of the channels given in the <chanlist> parameter. If multiple channels are given, then each definite length arbitrary block of data is separated by a comma.

[SOURce:]ARB:CURRent:CDWell:DWELl <*value*>,(@<chanlist>)
[SOURce:]ARB:CURRent:CDWell:DWELl? (@<chanlist>)
[SOURce:]ARB:VOLTage:CDWell:DWELl <*value*>,(@<chanlist>)
[SOURce:]ARB:VOLTage:CDWell:DWELl? (@<chanlist>)

The command specifies the dwell time in seconds for each current or voltage point in a constant-dwell Arb

You can only program one constant-dwell Arb type at a time, so setting the current CD Arb resets the voltage Arbs to 1 point at the minimum value. Programmed values can range from 0.0001 through 3600 seconds, with the resolution of 100 microsecond.

Parameter	Typical Return
0.0001 - 3600	<dwell value>
*RST 0.001	
Program a constant-dwell current time: ARB:CURR:CDW:DWEL 0.2, (@1) Program a constant-dwell voltage time: ARB:VOLT:CDW:DWEL 0.5, (@1)	

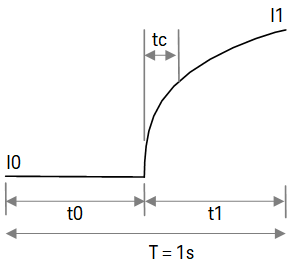
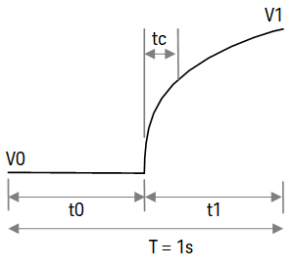
[SOURce:]ARB:CURRent:CDWell:POINts? (@<chanlist>)
[SOURce:]ARB:VOLTage:CDWell:POINts? (@<chanlist>)

The query returns the number of current or voltage points in a constant-dwell Arb in the form +n.nnnnnnE+nn.

Parameter	Typical Return
(none)	<number of points>
Program a constant-dwell time of 0.2 seconds: ARB:CURR:CDW:DWEL 0.2, (@1) ARB:VOLT:CDW:DWEL 0.2, (@1)	

Exponential (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Exponential		

[SOURce:]ARB:CURRent:EXPonential:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:EXPonential:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the ending current or voltage level after the exponential waveform ends. Referenced to I1 and V1 in the Exponential diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:END[:LEVel]? [MINimum | MAXimum,] <*step#*>, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:END[:LEVel]? [MINimum | MAXimum,] <*step#*> [, (@<chanlist>)]

The command specifies the ending current or voltage level after the exponential waveform ends for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<end value>
Step 0 - 99 *RST 0	(none)

Parameter	Typical Return
Programs an end level current: ARB:CURR:EXP:END 1, (@1) Programs an end level voltage: ARB:VOLT:EXP:END 20, (@1) Programs an end voltage for step 0: ARB:SEQ:STEP:VOLT:EXP:END 1, 0, (@1)	

[SOURce:]ARB:CURRent:EXPonential:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:EXPonential:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the initial current or voltage level when the exponential waveform starts. Referenced to I0 and V0 in the Exponential diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the initial current or voltage level after the exponential waveform ends for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>
Step 0 - 99 *RST 0	(none)
Programs a start level current: ARB:CURR:EXP:STAR 0, (@1) Programs a start level voltage: ARB:VOLT:EXP:STAR 0, (@1) Programs a start voltage for step 0: ARB:SEQ:STEP:VOLT:EXP:STAR 0, 0, (@1)	

[SOURce:]ARB:CURRent:EXPonential:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:EXPonential:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the exponential waveform occurs. Referenced to t0 in the Exponential diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:STARt:TIMe? [MINimum | MAXimum,] <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:STARt:TIMe? [MINimum | MAXimum,] <*step#*> [, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the exponential waveform occurs for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 0	<start time>
Step 0 - 99 *RST 0	(none)
Programs a start time for the current Arb: ARB:CURR:EXP:STAR:TIM 1, (@1) Programs a start time for the voltage Arb: ARB:VOLT:EXP:STAR:TIM 1, (@1) Programs a start time for step 0: ARB:SEQ:STEP:VOLT:EXP:STAR:TIM 1, 0, (@1)	

[SOURce:]ARB:CURRent:EXPonential:TCONstant <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:EXPonential:TCONstant? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:TCONstant <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:TCONstant? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time constant of the exponential curve. Referenced to tc in the Exponential diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:TCONstant <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:TCONstant ? [MINimum | MAXimum,] <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:TCONstant <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:TCONstant ? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time constant of the exponential curve for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 1	<time constant>

Parameter	Typical Return
Step 0 - 99	(none)
*RST 0	
Programs a time constant for the current Arb: ARB:CURR:EXP:TCON 5, (@1) Programs a time constant for the voltage Arb: ARB:VOLT:EXP:TCON 5, (@1) Programs a time constant for step 0: ARB:SEQ:STEP:VOLT:EXP:TCON 5, 0, (@1)	

[SOURce:]ARB:CURRent:EXPonential:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:EXPonential:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:EXPonential:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the total time for the current or voltage to go from the starting level to the ending level in seconds. Referenced to t1 in the Exponential diagrams.

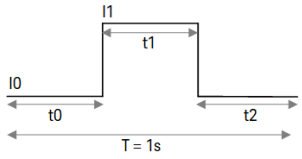
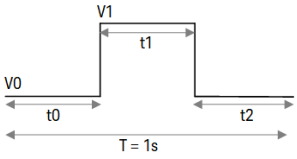
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:EXPonential:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:EXPonential:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the total time for the current or voltage to go from the starting level to the ending level in seconds for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX	<total time>
*RST 1	
Step 0 - 99	(none)
*RST 0	
Programs the total time for the current Arb: ARB:CURR:EXP:TIM 10, (@1) Programs the total time for the voltage Arb: ARB:VOLT:EXP:TIM 10, (@1) Programs the total time for step 0: ARB:SEQ:STEP:VOLT:EXP:TIM 10, 0, (@1)	

Pulse (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Pulse		

[SOURce:]ARB:CURRent:PULSe:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:PULSe:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time in seconds, after the pulse completes, that the starting current or voltage level persists. Referenced to t2 in the Pulse diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time in seconds, after the pulse completes, that the starting current or voltage level persists for Arb sequence steps.

Parameter	Typical Return
0 – 3600 | MIN | MAX *RST 0	<end time>
Step 0 - 99 *RST 0	(none)
Programs an end time for the current pulse: ARB:CURR:PULS:END:TIM 2, (@1) Programs an end time for the voltage pulse:ARB:VOLT:PULS:END:TIM 2, (@1) Programs an end time for step 0: ARB:SEQ:STEP:VOLT:PULS:END:TIM 2, 0, (@1)	

[SOURce:]ARB:CURRent:PULSe:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:PULSe:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the current or voltage before and after the pulse occurs. Referenced to I0 and V0 in the Pulse diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the current or voltage before and after the pulse occurs for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>
Step 0 - 99 *RST 0	(none)
Programs a start- and end-level current: ARB:CURR:PULS:STAR 1, (@1) Programs a start- and end-level voltage: ARB:VOLT:PULS:STAR 1, (@1) Programs a start- and end-level for step 0: ARB:SEQ:STEP:VOLT:PULS:STAR 1, 0, (@1)	

[SOURce:]ARB:CURRent:PULSe:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:PULSe:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the pulse occurs. Referenced to t0 in the Pulse diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:STARt:TIMe <time> | MINimum | MAXimum, <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:STARt:TIMe <time> | MINimum | MAXimum, <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the pulse occurs for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 0	<start time>
Step 0 - 99 *RST 0	(none)
Programs a start time for the current pulse: ARB:CURR:PULS:STAR:TIM 1, (@1) Programs a start time for the voltage pulse: ARB:VOLT:PULS:STAR:TIM 1, (@1) Programs a start time for step 0: ARB:SEQ:STEP:VOLT:PULS:STAR:TIM 1, 0, (@1)	

[SOURce:]ARB:CURRent:PULSe:TOP[:LEVel] <value> | MINimum | MAXimum, (@<chanlist>)
[SOURce:]ARB:CURRent:PULSe:TOP[:LEVel]? [MINimum | MAXimum,] (@<chanlist>)
[SOURce:]ARB:VOLTage:PULSe:TOP[:LEVel] <value> | MINimum | MAXimum, (@<chanlist>)
[SOURce:]ARB:VOLTage:PULSe:TOP[:LEVel]? [MINimum | MAXimum,] (@<chanlist>)

The command specifies the top current or voltage level of the pulse. Referenced to I1 and V1 in the Pulse diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:TOP[:LEVel] <value> | MINimum | MAXimum, <*step#*>, (@<chanlist>)
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:TOP[:LEVel]? [MINimum | MAXimum,] <*step#*>, (@<chanlist>)
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:TOP[:LEVel] <value> | MINimum | MAXimum, <*step#*>, (@<chanlist>)
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:TOP[:LEVel]? [MINimum | MAXimum,] <*step#*>, (@<chanlist>)

The command specifies the top current or voltage level of the pulse for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<top value>
Step 0 - 99 *RST 0	(none)
Programs the top level current: ARB:CURR:PULS:TOP 2, (@1) Programs the top level voltage: ARB:VOLT:PULS:TOP 20, (@1) Programs the top level for step 0: ARB:SEQ:STEP:VOLT:PULS:TOP 2, 0, (@1)	

[SOURce:]ARB:CURRent:PULSe:TOP:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:PULSe:TOP:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:TOP:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:PULSe:TOP:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time of the pulse in seconds. Referenced to t1 in the Pulse diagrams.

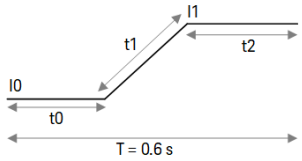
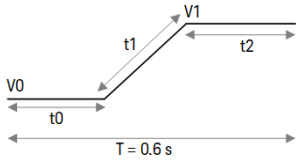
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:TOP:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:PULSe:TOP:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:TOP:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:PULSe:TOP:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time of the pulse in seconds for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 1	<end time>
Step 0 - 99 *RST 0	(none)
Programs a top time for the current pulse: ARB:CURR:PULS:TOP:TIM 2, (@1) Programs a top time for the voltage pulse: ARB:VOLT:PULS:TOP:TIM 2, (@1) Programs a top time for step 0: ARB:SEQ:STEP:VOLT:PULS:TOP:TIM 2, 0, (@1)	

Ramp (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Ramp		

[SOURce:]ARB:CURRent:RAMP:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:RAMP:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the ending current or voltage level after the ramp occurs. Referenced to I1 and V1 in the Ramp diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the ending current or voltage level after the ramp occurs for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<end value>
Step 0 - 99 *RST 0	(none)
Programs an end level current: ARB:CURR:RAMP:END 2, (@1) Programs an end level voltage: ARB:VOLT:RAMP:END 20, (@1) Programs an end level voltage for step 0: ARB:SEQ:STEP:VOLT:RAMP:END 20, 0, (@1)	

[SOURce:]ARB:CURRent:RAMP:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:RAMP:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time in seconds, after the ramp completes, that the ending current or voltage level persists. Referenced to t2 in the Ramp diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time in seconds, after the ramp completes, that the ending current or voltage level persists for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 0	<end time>
Step 0 - 99 *RST 0	(none)
Programs an end time for the current ramp: ARB:CURR:RAMP:END:TIM 2, (@1) Programs an end time for the voltage ramp:ARB:VOLT:RAMP:END:TIM 2, (@1) Programs an end time for step 0: ARB:SEQ:STEP:VOLT:RAMP:END:TIM 2, 0, (@1)	

[SOURce:]ARB:CURRent:RAMP:RTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:RAMP:RTIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:RTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:RTIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the rise time of the ramp in seconds. Referenced to t1 in the Ramp diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:RTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:RTIMe? [MINimum | MAXimum,] <*step#*>, [(@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:RTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:RTIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the rise time of the ramp in seconds for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX	<rise time>
*RST 1	
Step 0 - 99	(none)
*RST 0	
Programs a rise time for the current ramp: ARB:CURR:RAMP:RTIM 10, (@1) Programs a rise time for the voltage ramp: ARB:VOLT:RAMP:RTIM 10, (@1) Programs a rise time for step 0: ARB:SEQ:STEP:VOLT:RAMP:RTIM 10, 0, (@1)	

[SOURce:]ARB:CURRent:RAMP:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:RAMP:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the initial current or voltage level before the ramp occurs. Referenced to I0 and V0 in the Ramp diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the initial current or voltage level before the ramp occurs for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>
Step 0 - 99 *RST 0	(none)
Programs a start level current: ARB:CURR:RAMP:STAR 1, (@1) Programs a start level voltage: ARB:VOLT:RAMP:STAR 1, (@1) Programs a start level for step 0: ARB:SEQ:STEP:VOLT:RAMP:STAR 1, 0, (@1)	

[SOURce:]ARB:CURRent:RAMP:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:RAMP:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:RAMP:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the ramp occurs. Referenced to t0 in the Ramp diagrams.

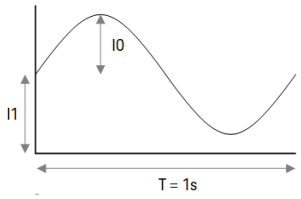
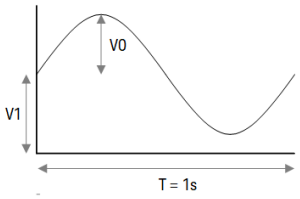
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:RAMP:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:RAMP:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received for Arb sequence steps.

Parameter	Typical Return
0 – 3600 | MIN | MAX *RST 0	<start time>
Step 0 - 99 *RST 0	(none)
Programs a start time for the current ramp: ARB:CURR:RAMP:STAR:TIM 1, (@1) Programs a start time for the voltage ramp: ARB:VOLT:RAMP:STAR:TIM 1, (@1) Programs a start time for step 0: ARB:SEQ:STEP:VOLT:RAMP:STAR:TIM 1, 0, (@1)	

Sinusoid (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Sinusoid		

[SOURce:]ARB:CURRent:SINusoid:AMPLitude <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:SINusoid:AMPLitude? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:AMPLitude <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:AMPLitude? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the amplitude or peak value of the current or voltage sine wave. Referenced to I0 and V0 in the Sinusoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:AMPLitude <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:AMPLitude? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:AMPLitude <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:AMPLitude? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the amplitude or peak value of the current or voltage sine wave for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST 0	<amplitude>
Step 0 - 99 *RST 0	(none)
Programs a current amplitude: ARB:CURR:SIN:AMPL 5, (@1) Programs a voltage amplitude: ARB:VOLT:SIN:AMPL 10, (@1) Programs a voltage amplitude for step 0: ARB:SEQ:STEP:VOLT:SIN:AMPL 5, 0, (@1)	

[SOURce:]ARB:CURRent:SINusoid:FREQuency <frequency> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:SINusoid:FREQuency? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:FREQuency <frequency> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:FREQuency? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the frequency of the current or voltage sine wave in Hertz.

[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:FREQuency <frequency> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:FREQuency? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:FREQuency <frequency> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:FREQuency? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the frequency of the current or voltage sine wave in Hertz for Arb sequence steps.

Parameter	Typical Return
2.7778E-6 - 100 | MIN | MAX *RST 1	<frequency>
Step 0 - 99 *RST 0	(none)
Programs a current frequency: ARB:CURR:SIN:FREQ 10, (@1) Programs a voltage frequency: ARB:VOLT:SIN:FREQ 10, (@1) Programs a voltage frequency for step 0: ARB:SEQ:STEP:VOLT:SIN:FREQ 10, 0, (@1)	

[SOURce:]ARB:CURRent:SINusoid:OFFSet <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:SINusoid:OFFSet? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:OFFSet <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:SINusoid:OFFSet? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the offset of the current or voltage sine wave from zero. Referenced to I1 and V1 in the Sinusoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:OFFSet <value> | MINimum | MAXimum, <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:SINusoid:OFFSet? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:OFFSet <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:SINusoid:OFFSet? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

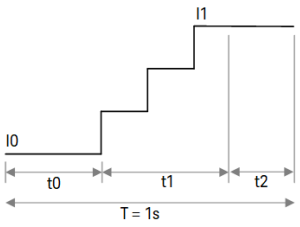
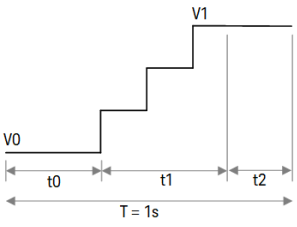
The command specifies the offset of the current or voltage sine wave from zero for Arb sequence steps.

NOTE **The output cannot generate negative amplitudes. Therefore, the offset value cannot be less than the peak value of the sine wave.**

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<offset value>
Step 0 - 99 *RST 0	(none)
Programs a current offset: ARB:CURR:SIN:OFFS 5, (@1) Programs a voltage offset: ARB:VOLT:SIN:OFFS 10, (@1) Programs a voltage offset for step 0: ARB:SEQ:STEP:VOLT:SIN:OFFS 5, 0, (@1)	

Staircase (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Staircase		

[SOURce:]ARB:CURRent:STAircase:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the ending current or voltage level after the staircase occurs. Referenced to I1 and V1 in the Staircase diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the ending current or voltage level after the staircase occurs for Arb sequence steps.

NOTE **The difference between the start level and the end level is divided equally among the number of steps.**

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<end value>
Step 0 - 99 *RST 0	(none)

Parameter	Typical Return
Programs an end level current: ARB:CURR:STA:END 2, (@1) Programs an end level voltage: ARB:VOLT:STA:END 20, (@1) Programs an end level voltage for step 0: ARB:SEQ:STEP:VOLT:STA:END 2, 0, (@1)	

[SOURce:]ARB:CURRent:STAircase:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time in seconds, after the staircase completes, that the ending current or voltage level persists. Referenced to t2 in the Staircase diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time in seconds, after the staircase completes, that the ending current or voltage level persists for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 0	<end time>
Step 0 - 99 *RST 0	(none)
Programs an end time after the current staircase:ARB:CURR:STA:END:TIM 2, (@1) Programs an end time after the voltage staircase:ARB:VOLT:STA:END:TIM 2, (@1) Programs an end time for step 0:ARB:SEQ:STEP:VOLT:STA:END:TIM 2, 0, (@1)	

[SOURce:]ARB:CURRent:STAircase:NSTeps <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:NSTeps? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:NSTeps <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:NSTeps? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the number of steps in the staircase.

[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:NSTeps <value> | MINimum | MAXimum, <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:NSTeps? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:NSTeps <value> | MINimum | MAXimum, <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:NSTeps? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the number of steps in the staircase for Arb sequence steps.

Parameter	Typical Return
1 – 500 | MIN | MAX *RST 10	<number of steps>
Step 0 - 99 *RST 0	(none)
Programs five steps for the current staircase: ARB:CURR:STA:NST 5, (@1) Programs five steps for the voltage staircase: ARB:VOLT:STA:NST 5, (@1) Programs five steps for sequence step 0: ARB:SEQ:STEP:VOLT:STA:NST 5, 0, (@1)	

[SOURce:]ARB:CURRent:STAircase:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the initial current or voltage before the staircase occurs. Referenced to I0 and V0 in the Staircase diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*> [, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the initial current or voltage before the staircase occurs for Arb sequence steps.

NOTE **The difference between the start level and the end level is divided equally among the number of steps.**

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>
Step 0 - 99 *RST 0	(none)
Programs a start level current: ARB:CURR:STA:STAR 1, (@1) Programs a start level voltage: ARB:VOLT:STA:STAR 10, (@1) Programs an start level for step 0:ARB:SEQ:STEP:VOLT:STA:STAR 1, 0, (@1)	

[SOURce:]ARB:CURRent:STAircase:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the staircase occurs. Referenced to t0 in the Staircase diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the staircase occurs for Arb sequence steps.

Parameter	Typical Return
0 -3600 | MIN | MAX *RST 0	<start time>
Step 0 - 99 *RST 0	(none)
Programs a start time for the current staircase:ARB:CURR:STA:STAR:TIM 1, (@1) Programs a start time for the voltage staircase:ARB:VOLT:STA:STAR:TIM 1, (@1) Programs a start time for step 0:ARB:SEG:STEP:VOLT:STA:STAR:TIM 1, 0, (@1)	

[SOURce:]ARB:CURRent:STAircase:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STAircase:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STAircase:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the total time to complete all of the staircase steps in seconds. Referenced to t1 in the Staircase diagrams. Each step is of equal time.

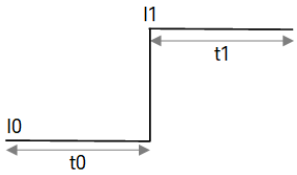
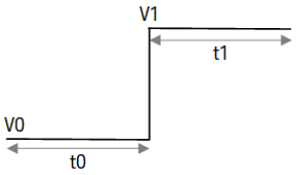
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STAircase:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STAircase:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the total time to complete all of the staircase steps in seconds for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 1	<staircase time>
Step 0 - 99 *RST 0	(none)
Programs a time for the current staircase: ARB:CURR:STA:TIM 10, (@1) Programs a time for the voltage staircase: ARB:VOLT:STA:TIM 10, (@1) Programs a time for step 0:ARB:SEG:STEP:VOLT:STA:TIM 10, 0, (@1)	

Step (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Step		

[SOURce:]ARB:CURRent:STEP:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STEP:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:END[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:END[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the step current or voltage level. Referenced to I1 and V1 in the Step diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:END[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:END[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the step current or voltage level for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<end value>
Step 0 - 99 *RST 0	(none)
Programs the step level current: ARB:CURR:STEP:END 2, (@1) Programs the step level voltage: ARB:VOLT:STEP:END 20, (@1) Programs a step level for step 0: ARB:SEQ:STEP:VOLT:STEP:END 2, 0, (@1)	

[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:END:TIME <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:END:TIME? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:END:TIME <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:END:TIME? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time in seconds, after the step completes, that the ending current or voltage level persists for Arb sequence steps.

Parameter	Typical Return
0 – 262.144 | MIN | MAX *RST 0	<end time>
Step 0 - 99 *RST 0	(none)
Programs a end time for step 0: ARB:SEQ:STEP:VOLT:STEP:END:TIM 1, 0, (@1)	

[SOURce:]ARB:CURRent:STEP:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STEP:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the initial current or voltage level. Referenced to I0 and V0 in the Step diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the initial current or voltage level for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>

Parameter	Typical Return
Step 0 - 99	(none)
*RST 0	
Programs a start level current: ARB:CURR:STEP:STAR 1, (@1) Programs a start level voltage: ARB:VOLT:STEP:STAR 1, (@1) Programs a start level for step 0: ARB:SEQ:STEP:VOLT:STEP:STAR 1, 0, (@1)	

[SOURce:]ARB:CURRent:STEP:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:STEP:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:STEP:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the step occurs. Referenced to t0 in the Step diagrams.

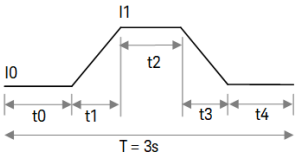
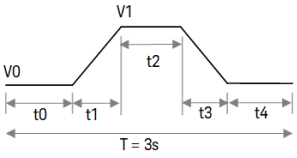
[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:STEP:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:STEP:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the step occurs for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX	<start time>
*RST 0	
Step 0 - 99	(none)
*RST 0	
Programs a start time for the current step: ARB:CURR:STEP:STAR:TIM 1, (@1) Programs a start time for the voltage step: ARB:VOLT:STEP:STAR:TIM 1, (@1) Programs a start time for step 0: ARB:SEQ:STEP:VOLT:STEP:STAR:TIM 1, 0, (@1)	

Trapezoid (Option E36150ADVU)

Back to ARB Commands

Arb Type	Current	Voltage
Trapezoid		

[SOURce:]ARB:CURRent:TRAPezoid:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:END:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:END:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time in seconds, after the trapezoid, that the starting current or voltage level persists. Referenced to t4 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:END:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:END:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time in seconds, after the trapezoid, that the starting current or voltage level persists for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 0	<end time>
Step 0 - 99 *RST 0	(none)
Programs an end time after the current trapezoid: ARB:CURR:TRAP:END:TIM 2, (@1) Programs an end time after the voltage trapezoid: ARB:VOLT:TRAP:END:TIM 2, (@1) Programs an end time for step 0:ARB:SEQ:STEP:VOLT:TRAP:END:TIM 2, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:FTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:FTIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:FTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:FTIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the fall time of the trapezoid in seconds. Referenced to t3 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:FTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:FTIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:FTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:FTIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the fall time of the trapezoid in seconds for Arb sequence steps.

Parameter	Typical Return
0 - 3600 | MIN | MAX *RST 1	<fall time>
Step 0 - 99 *RST 0	(none)
Programs a fall time for the current trapezoid: ARB:CURR:TRAP:FTIM 10, (@1) Programs a fall time for the voltage trapezoid: ARB:VOLT:TRAP:FTIM 10, (@1) Programs a fall time for step 0:ARB:SEQ:STEP:VOLT:TRAP:FTIM 10, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:RTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:RTIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:RTIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:RTIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the rise time of the trapezoid in seconds. Referenced to t1 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:RTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:RTIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:RTIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:RTIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the rise time of the trapezoid in seconds for Arb sequence steps.

Parameter	Typical Return
0 – 3600 | MIN | MAX *RST 1	<rise time>
Step 0 - 99 *RST 0	(none)
Programs a rise time for the current trapezoid: ARB:CURR:TRAP:RTIM 10, (@1) Programs a rise time for the voltage trapezoid: ARB:VOLT:TRAP:RTIM 10, (@1) Programs a rise time for step 0:ARB:SEQ:STEP:VOLT:TRAP:RTIM 10, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:STARt[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:STARt[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the initial and final current or voltage level of the trapezoid. Referenced to I0 and V0 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:STARt[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:STARt[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the initial and final current or voltage level of the trapezoid for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<start value>
Step 0 - 99 *RST 0	(none)
Programs a start- and end-level current: ARB:CURR:TRAP:STAR 1, (@1) Programs a start- and end- level voltage: ARB:VOLT:TRAP:STAR 1, (@1) Programs a start- and end-level for step 0: ARB:SEQ:STEP:VOLT:TRAP:STAR 1, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:STARt:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:STARt:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the rising ramp occurs. Referenced to t0 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:STARt:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:STARt:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the delay in seconds after the trigger is received, but before the rising ramp occurs for Arb Sequence steps.

Parameter	Typical Return
0 – 3600 | MIN | MAX *RST 0	<start time>
Step 0 - 99 *RST 0	(none)
Programs a start time for the current trapezoid: ARB:CURR:TRAP:STAR:TIM 1, (@1) Programs a start time for the voltage trapezoid: ARB:VOLT:TRAP:STAR:TIM 1, (@1) Programs a start time for step 0: ARB:SEQ:STEP:VOLT:TRAP:STAR:TIM 1, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:TOP[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:TOP[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:TOP[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:TOP[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the top current or voltage level of the trapezoid. Referenced to I1 and V1 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:TOP[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:TOP[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:TOP[:LEVel] <value> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:TOP[:LEVel]? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the top current or voltage level of the trapezoid for Arb sequence steps.

Parameter	Typical Return
minimum - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range). *RST MIN	<top value>
Step 0 - 99 *RST 0	(none)
Programs the top level current: ARB:CURR:TRAP:TOP 2, (@1) Programs the top level voltage: ARB:VOLT:TRAP:TOP 20, (@1) Programs a top level for step 0: ARB:SEQ:STEP:VOLT:TRAP:TOP 2, 0, (@1)	

[SOURce:]ARB:CURRent:TRAPezoid:TOP:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:CURRent:TRAPezoid:TOP:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:TOP:TIMe <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]ARB:VOLTage:TRAPezoid:TOP:TIMe? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the time of the top level of the trapezoid in seconds. Referenced to t2 in the Trapezoid diagrams.

[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:TOP:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:TRAPezoid:TOP:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:TOP:TIMe <time> | MINimum | MAXimum, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:TRAPezoid:TOP:TIMe? [MINimum | MAXimum,] <*step#*>[, (@<chanlist>)]

The command specifies the time of the top level of the trapezoid in seconds for Arb sequence steps.

Parameter	Typical Return
0 – 3600 | MIN | MAX *RST 1	<top time>
Step 0 - 99 *RST 0	(none)
Programs a top time for the current trapezoid: ARB:CURR:TRAP:TOP:TIM 2, (@1) Programs a top time for the voltage trapezoid: ARB:VOLT:TRAP:TOP:TIM 2, (@1) Programs a top time for step 0: ARB:SEQ:STEP:VOLT:TRAP:TOP:TIM 2, 0, (@1)	

User-Defined

Back to ARB Commands

NOTE For standard option, a comma-delimited list of up to 100 steps may be programmed. With Option E36150ADVU, you are able to programmed up to 512 steps.

[SOURce:]ARB:UDEFined:BOSTep[:DATA] ON | OFF | 1 | 0 {,0 or 1} [, (@<chanlist>)]
[SOURce:]ARB:UDEFined:BOSTep[:DATA]? [(@<chanlist>)]

The command specifies which user-defined Arb points or steps will generate a trigger-out signal at the beginning of the step (BOSTep). A comma-delimited list of up to 512 steps may be programmed. The state is either ON (1) or OFF (0). A trigger is only generated when the state is set to ON.

[SOURce:]ARB:SEQuence:STEP:UDEFined:BOSTep[:DATA] ON | OFF | 1 | 0 {,0 or 1}, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:UDEFined:BOSTep[:DATA]? <*step#*>[, (@<chanlist>)]

The command specifies which user-defined Arb points or steps will generate a trigger-out signal at the beginning of the step (BOSTep) for Arb sequence steps. The last value in the sequence is the step number.

Parameter	Typical Return
ON | OFF | 1 | 0 *RST 1 point with a value of OFF	0 or 1 [,0 or 1]
Step 0 - 99 *RST 0	(none)
Output a trigger at the start of the steps: ARB:UDEF:BOST ON,ON,ON,ON,ON, (@1) Output triggers at the start of the steps during sequence step 0:ARB:SEQ:STEP:UDEF:BOST ON,ON,ON,ON,ON, 0, (@1)	

Remarks

- At the end of the user-defined Arb, the output state of the unit depends upon the ARB:TERMinate:LAST program settings.
- The order in which the values are entered determines the sequence of execution.
- To create a valid user-defined Arb, the current level, voltage level, BOST, EOST, 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.
- This command replaces the previous LIST:TOUTput:BOSTep[:DATA] command and should be used in new applications. LIST:TOUTput:BOSTep[:DATA] is still available for backward compatibility.

[SOURce:]ARB:UDEFined:BOSTep:POINts? [(@<chanlist>)]

The query returns the number of beginning-of-step points in the user-defined Arb, not the point values.

[SOURce:]ARB:SEQuence:STEP:UDEFined:BOSTep:POINts? <*step#*>[, (@<chanlist>)]

The query returns the number of beginning-of-step points in the user-defined Arb for Arb sequence steps.

Parameter	Typical Return
(none)	<number of points>
Step 0 - 99	(none)
*RST 0	
Returns the number of BOST points: ARB:CURR:UDEF:BOST:POIN? (@1) Returns the number of BOST points for step 0: ARB:SEQ:STEP:VOLT:UDEF:BOST:POIN? 0, (@1)	

Remarks

- This query replaces the previous LIST:TOUTput:BOSTep:POINts? query and should be used in new applications. LIST:TOUTput:BOSTep:POINts? is still available for backward compatibility.

[SOURce:]ARB:UDEFined:EOSTep[:DATA] ON | OFF | 1 | 0 {,0 or 1} [, (@<chanlist>)]
[SOURce:]ARB:UDEFined:EOSTep[:DATA]? [(@<chanlist>)]

The command specifies which user-defined Arb points or steps will generate a trigger-out signal at the end of the step (EOSTep). A comma-delimited list of up to 512 steps may be programmed. The state is either ON (1) or OFF (0). A trigger is only generated when the state is set to ON.

[SOURce:]ARB:SEQuence:STEP:UDEFined:EOSTep[:DATA] ON | OFF | 1 | 0 {,0 or 1}, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:UDEFined:EOSTep[:DATA]? <*step#*>[, (@<chanlist>)]

The command specifies which user-defined Arb points or steps will generate a trigger-out signal at the end of the step (EOSTep) for Arb sequence steps. The last value in the sequence is the step number.

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1 [,0 or 1]
*RST 1 point with a value of OFF	
Step 0 - 99	(none)
*RST 0	
Output a trigger at the end of the steps: ARB:UDEF:EOST ON,ON,ON,ON,ON, (@1) Output triggers at the end of the steps during sequence step 0:ARB:SEQ:STEP:UDEF:EOST ON,ON,ON,ON,ON, 0, (@1)	

Remarks

- At the end of the user-defined Arb, the output state of the unit depends upon the ARB:TERMinate:LAST program settings.
- The order in which the values are entered determines the sequence of execution.

- To create a valid user-defined Arb, the current level, voltage level, BOST, EOST, 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.
- This command replaces the previous LIST:TOUTput:EOSTep[:DATA] command and should be used in new applications. LIST:TOUTput:EOSTep[:DATA] is still available for backward compatibility.

[SOURce:]ARB:UDEFined:EOSTep:POINts? [(@<chanlist>)]

The query returns the number of end-of-step points in the user-defined Arb, not the point values.

[SOURce:]ARB:SEQuence:STEP:UDEFined:EOSTep:POINts? <*step#*>[, (@<chanlist>)]

The query returns the number of end-of-step points in the user-defined Arb for Arb sequence steps.

Parameter	Typical Return
(none)	<number of points>
Step 0 - 99	(none)
*RST 0	
Returns the number of EOST points: ARB:CURR:UDEF:EOST:POIN? (@1) Returns the number of EOST points for step 0: ARB:SEQ:STEP:VOLT:UDEF:EOST:POIN? 0, (@1)	

Remarks

- This query replaces the previous LIST:TOUTput:EOSTep:POINts? query and should be used in new applications. LIST:TOUTput:EOSTep:POINts? is still available for backward compatibility.

[SOURce:]ARB:UDEFined:DWELl <*value*>{<,*value*>}[, (@<chanlist>)]
[SOURce:]ARB:UDEFined:DWELl? [(@<chanlist>)]

The command specifies the dwell time for each user-defined current Arb point. A comma-delimited list of up to 512 points may be programmed. Dwell time is the time that the output will remain at a specific point. Dwell times can be programmed from 0 through 3600 seconds with the resolution of 100 microsecond.

[SOURce:]ARB:SEQuence:STEP:UDEFined:DWELl <*value*>{<,*value*>}, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:UDEFined:DWELl? <*step#*>[, (@<chanlist>)]

The command specifies the dwell time for each user-defined current Arb point for Arb Sequence steps. The last value in the sequence is the step number.

Parameter	Typical Return
0 - 3600	<dwell time>
*RST 0.001	

Parameter	Typical Return
Step 0 - 99	(none)
*RST 0	
Program a dwell list: ARB:UDEF:DWEL 0.1,0.2,0.3,0.4,0.5, (@1) Program a dwell list for step 0: ARB:SEQ:STEP:UDEF:DWEL 0.1,0.2,0.3,0.4,0.5, 0, (@1)	

Remarks

- At the end of the user-defined Arb, the output state of the unit depends upon the ARB:TERMinate:LAST program settings.
- The order in which the values are entered determines the sequence of execution.
- To create a valid user-defined Arb, the current level, voltage level, BOST, 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.
- This command replaces the previous LIST:DWELl command and should be used in new applications. LIST:DWELl is still available for backward compatibility.

[SOURce:]ARB:UDEFined:DWELl:POINts? [(@<chanlist>)]

The query returns the number of dwell points in the user-defined Arb, not the point values.

[SOURce:]ARB:SEQuence:STEP:CURRent:UDEFined:DWELl:POINts? <*step#*>[, (@<chanlist>)]

The query returns the number of dwell points in the user-defined Arb for Arb sequence steps.

Parameter	Typical Return
(none)	<number of points>
Step 0 - 99	(none)
*RST 0	
Returns the number of dwell points: ARB:CURR:UDEF:DWEL:POIN? (@1) Returns the number of dwell points for step 0: ARB:SEQ:STEP:VOLT:UDEF:DWEL:POIN? 0, (@1)	

Remarks

- This query replaces the previous LIST:DWELl:POINts? query and should be used in new applications. LIST:DWELl:POINts? is still available for backward compatibility.

[SOURce:]ARB:CURRent:UDEFined:LEVel <*value*>{<,*value*>}[,(@<chanlist>)]
[SOURce:]ARB:CURRent:UDEFined:LEVel ? [(@<chanlist>)]
[SOURce:]ARB:VOLTage:UDEFined:LEVel <*value*>{<,*value*>}[,(@<chanlist>)]
[SOURce:]ARB:VOLTage:UDEFined:LEVel ? [(@<chanlist>)]

The command specifies the level of each current or voltage point in a user-defined Arb. A comma-delimited list of up to 512 points may be programmed.

[SOURce:]ARB:SEQuence:STEP:CURRent:UDEFined:LEVel <*value*>{<,*value*>}, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:CURRent:UDEFined:LEVel ? <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:UDEFined:LEVel <*value*>{<,*value*>}, <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:UDEFined:LEVel ? <*step#*>[, (@<chanlist>)]

The command specifies the level of each current or voltage point in a user-defined Arb for Arb sequence steps. The last value in the sequence is the step number.

Parameter	Typical Return
minimum - maximum (Values are range and product-model dependent. See Programming Range). *RST 1 point set to the minimum programmable value.	<value> [,<value>]
Step 0 - 99 *RST 0	(none)
Program the current points: ARB:CURR:UDEF 5,4,3,2,1, (@1) Program the voltage points: ARB:VOLT:UDEF 20,21,22,23,24, (@1) Program the current points for step 0: ARB:SEQ:STEP:CURR:UDEF 5,4,3,2,1, 0, (@1) Program the voltage points for step 0: ARB:SEQ:STEP:VOLT:UDEF 20,21,22,23,24, 0, (@1)	

Remarks

- At the end of the user-defined Arb, the output state of the unit depends upon the ARB:TERMinate:LAST program settings.
- The order in which the values are entered determines the sequence of execution.
- To create a valid user-defined Arb, the current level, voltage level, BOST, 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.
- These commands replaces the previous LIST:CURRent and LIST:VOLTage commands and should be used in new applications. LIST:CURRent and LIST:VOLTage are still available for backward compatibility.

[SOURce:]ARB:CURRent:UDEFined:LEVel:POINts? [(@<chanlist>)]
[SOURce:]ARB:VOLTage:UDEFined:LEVel:POINts? [(@<chanlist>)]

The query returns the number of current or voltage points in the user-defined Arb, not the point values.

[SOURce:]ARB:SEQuence:STEP:CURRent:UDEFined:LEVel:POINts? <*step#*>[, (@<chanlist>)]
[SOURce:]ARB:SEQuence:STEP:VOLTage:UDEFined:LEVel:POINts? <*step#*>[, (@<chanlist>)]

The query returns the number of current or voltage points in the user-defined Arb for Arb sequence steps.

Parameter	Typical Return
(none)	<number of points>
Step 0 - 99	(none)
*RST 0	
Returns the number of current points: ARB:CURR:UDEF:LEV:POIN? (@1) Returns the number of voltage points: ARB:VOLT:UDEF:LEV:POIN? (@1) Returns the number of points for step 0: ARB:SEQ:STEP:VOLT:UDEF:LEV:POIN? 0, (@1)	

Remarks

- These queries replaces the previous LIST:CURRent:POINts? and LIST:VOLTage:POINts? queries and should be used in new applications. LIST:CURRent:POINts? and LIST:VOLTage:POINts? are still available for backward compatibility.

Arb Sequence (Option E36150ADVU)

Back to ARB Commands

[SOURce:]ARB:SEQuence:COUNt <*value*>, (@<chanlist>)
[SOURce:]ARB:SEQuence:COUNt? [MINimum | MAXimum,] (@<chanlist>)

The command sets the number of times that the Arb sequence is repeated. The repeat count range is 1 through 4096.

Parameter	Typical Return
1 - 4096 | MIN | MAX | INFinity	<sequence count>
*RST 1	
Sets the sequence count to 10: ARB:SEQ:COUN 10, (@1)	

Remarks

- Use the INFinity parameter to repeat the sequence continuously. If MAX, or a value greater than 4096 is programmed, the sequence will also repeat continuously.
- Use ABORt:TRANsient to stop the sequence at any time. When the sequence is aborted, the output returns to the settings that were in effect before the sequence started.
- If a count of 9.9E37 is returned, it means that the sequence is set to repeat continuously.

[SOURce:]ARB:SEQuence:LENGth? (@<chanlist>)

Returns the total number of steps in the Arb sequence.

Parameter	Typical Return
(none)	<number of steps>
Returns the number of steps in the Arb sequence: ARB:SEQ:LENG? (@1)	

[SOURce:]ARB:SEQuence:PRESet:VOLTage SP16750 | SP7637 | SPFW16750 | RSTB16750 | MDRP16750 | MTRG7637 | LDMP16750, (@<chanlist>)

NOTE In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.

This command only applies to instrument with serial number MY63000000 and above. For more information, please contact Keysight support at https://www.keysight.com/find/assist.

The command presets the Arb sequence according to the specified preset profile.

Preset Profile	Waveform Name	Automotive Standards
SP16750	Starting Profile	ISO_16750-2:2010 4_6_3
SP7637	Starting Profile V1	ISO_7637-2:2004 5_6_4 (DIN40839)
SPFW16750	Starting Profile FW	ISO_16750-2:2010 4_6_3
RSTB16750	Reset Behavior	ISO_16750-2:2010 4_6_2
MDRP16750	Momentary Drop	ISO_16750-2:2010 4_6_1
MTRG7637	Motor Regen	ISO_7637-2:2011 5_6_2b
LDMP16750 (Applicable for E36155A only)	Load Dump	ISO_16750-2:2010 4_6_4b

Parameter	Typical Return
SP16750 | SP7637 | SPFW16750 | RSTB16750 | MDRP16750 | MTRG7637 | LDMP16750	SP16750, SP7637, SPFW16750, RSTB16750, MDRP16750, MTRG7637, or LDMP16750
Preset the Arb sequence to Starting Point (ISO_16750-2:2010 4_6_3): ARB:SEQ:PRES:VOLT SP16750, (@1)	

[SOURce:]ARB:SEQuence:QUALity? (@<chanlist>)

The query returns the quality of the waveforms in the Arb sequence. This is a number between 16 and 100 that is an indication of how well waveforms are represented. It is the number of points used to represent ARBs that are smooth curves (ramp, trapezoid, exponential, and sine). 16 is the minimum number of points that defines the curve (worst quality). 100 is the maximum number of points that defines the curve (best quality).

Note that the quality of a given waveform is determined by the total number and type of waveforms that comprise the sequence. The maximum number of points allowed in a sequence is 511. As more waveforms are added to the sequence, the quality of the waveforms is reduced to accommodated the 511-point limit.

Parameter	Typical Return
(none)	a value between 16 and 100
Returns the quality of the Arb sequence : ARB:SEQ:QUAL? (@1)	

[SOURce:]ARB:SEQuence:RESet (@<chanlist>)

The command resets the Arb sequence to its power-on default setting: Step = 0; Shape = PULSe

Parameter	Typical Return
(none)	(none)
Returns the number of steps in the Arb sequence: ARB:SEQ:LENG? (@1)	

[SOURce:]ARB:SEQuence:STEP:COUNt <value>, <step#>, (@<chanlist>)
[SOURce:]ARB:SEQuence:STEP:COUNt? [MINimum | MAXimum,] <step#>, (@<chanlist>)

The command sets the number of times that the sequence step is repeated. The repeat count range is 1 through >16 million. The maximum number of steps that can be programmed is 100. This setting is only valid if the ARB:SEQuence:STEP:PACing is TRIGgered.

Parameter	Typical Return
Count: 1 - 16,777,216 | MIN | MAX | INFinity	<step count>
*RST 1	
Step: 0 - 99	(none)
*RST 0	
Sets the a repeat count of 10 for step 1: ARB:SEQ:STEP:COUN 10, 1, (@1)	

Remarks

- Use the INFinity parameter to repeat the sequence step continuously. If MAX, or a value greater than 16,777,216 is programmed, the step will also repeat continuously.
- Use ABORt:TRANsient to stop the sequence at any time. When the sequence is aborted, the output returns to the settings that were in effect before the sequence started.
- If a count of 9.9E37 is returned, it means that the sequence step is set to repeat continuously.

[SOURce:]ARB:SEQuence:STEP:CURRent
[SOURce:]ARB:SEQuence:STEP:VOLTage

The command programs the waveform steps within an Arb sequence.

All of the ARB:CURRent and ARB:VOLTage commands that set waveform parameters have corresponding commands that set those same parameters for waveform steps within an Arb sequence. They follow the pattern:

[SOURce:]ARB:CURRent:EXPonential:END <level>, (@<chanlist>)
[SOURce:]ARB:**SEQuence:STEP**:CURRent:EXPonential:END <level>, **<step#>**, (@<chanlist>)

where items in bold are added to the sequence step version of the single Arb command. The <step#> parameter indicates the step number within the Arb sequence. The following Arb functions use the ARB:SEQuence <step#> format:

Exponential, Pulse, Ramp, Sinusoid, Staircase, Step, Trapezoid, and User-defined.

[SOURce:]ARB:SEQuence:STEP:FUNCtion:SHAPe <function>, <step#>, (@<chanlist>)
[SOURce:]ARB::SEQuence:STEP:FUNCtion:SHAPe? <step#>, (@<chanlist>)

The command creates a new Arb sequence step or changes the waveform of an existing sequence step. The maximum number of steps that can be programmed is 100. The following waveforms may be assigned to a sequence step:

Shape	Descriptions
STEP	Specifies a step
RAMP	Specifies a ramp
STAircase	Specifies a staircase
SINusoid	Specifies a sine wave
PULSe	Specifies a pulse
TRAPezoid	Specifies a trapezoid
EXPonential	Specifies a exponential waveform
UDEFined	Specifies a user-defined waveform

Parameter	Typical Return
STEP | RAMP | STAircase | SINusoid | PULSe | TRAPezoid | EXPonential | UDEFined	STEP, RAMP, STA, SIN, PULS, TRAP, EXP, or UDEF
*RST PULS	
Step 0 - 99	(none)
*RST 0	
Specify a shape for sequence step 1: ARB:SEQ:STEP:FUNC:SHAP SIN, 1, (@1)	

Remarks

- If the <step#> specified is an existing sequence step, that step will be changed to the specified waveshape with all parameters reset to default values.
- New sequence steps must be specified sequentially. To create a new sequence step, <step#> should be specified to be the current length of the sequence (see ARB:SEQuence:LENGth?). When a step is added, all parameters of the step waveform are reset to their default values.

[SOURce:]ARB:SEQuence:STEP:PACing <pacing>, <step#>, (@<chanlist>)
[SOURce:]ARB::SEQuence:STEP:PACing? <step#>, (@<chanlist>)

The command specifies the type of pacing for the specified step number as follows:

Type	Descriptions
DWELl	When dwell paced, the step moves to the next step when the dwell time is finished.
TRIGgered	When trigger paced, the step waits at the last value of the step until a trigger is received. The next step is started upon receipt of the trigger.

Parameter	Typical Return
DWELl | TRIGgered	DWEL or TRIG
*RST DWELl	
Step 0 - 99	(none)
*RST 0	
Specify trigger pacing for sequence step 1: ARB:SEQ:STEP:PAC TRIG, 1, (@1)	

Remarks

- If the <step#> specified is an existing sequence step, that step will be changed to the specified waveshape with all parameters reset to default values.
- New sequence steps must be specified sequentially. To create a new sequence step, <step#> should be specified to be the current length of the sequence (see ARB:SEQuence:LENGth?). When a step is added, all parameters of the step waveform are reset to their default values.

[SOURce:]ARB:SEQuence:TERMinate:LAST <Bool>, (@<chanlist>)
[SOURce:]ARB:SEQuence:TERMinate:LAST? (@<chanlist>)

The command determines the output value when the Arb sequence terminates. The state is either ON (1) or OFF (0).

State	Descriptions
ON	The output remains at the last Arb sequence value. The last current, or voltage Arb value becomes the IMMediate value when the sequence completes.
OFF	The output returns to the settings in effect before the sequence started. This also applies when the sequence is aborted,

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1
*RST Off	
Sets the sequence to terminate with the voltage, or current set at the last sequence value: ARB:SEQ:TERM:LAST ON, (@1)	

Remarks

- The query returns 0 if the output returns to the settings that were in effect before the sequence started, and 1 if the output remains at the last sequence value.

CALibration:ASAVE ON | OFF | 1 | 0[, (@<chanlist>)]
CALibration:ASAVE?

The command enables or disables automatic saving of calibration constants. When the Auto Save feature is enabled when leaving the CAL state, the calibration data will automatically move the CAL data from volatile memory to non-volatile memory and the CALibration:COUNt value is increased by 1.

The query returns 0 (OFF) or 1 (ON).

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
Enables the CAL auto save feature: CAL:ASAVE 1	

CALibration:COUNt?

The query returns the number of times the power supply has been calibrated. Your power supply was calibrated before it left the factory. When you receive your power supply, read the count to determine its initial value.

Parameter	Typical return
(none)	<count>
Returns the calibration count: CAL:COUN?	

CALibration:DATE "<*string*>"
CALibration:DATE?

The command stores the date that the power supply was last calibrated in non-volatile memory. This command is equivalent to the CALibration:STRing command.

The query returns the date. If no date is stored, an empty quoted string ("") is returned.

Parameter	Typical return
"<string>"	"<string>"
Enters the calibration date: CAL:DATE "4/22/17"	

CALibration:SAVE

The command saves calibration constants in non-volatile memory after the calibration procedure has been completed. When you exit (CALibration:STATe OFF) without saving, the previous constants are restored. If the Auto Save feature is not enabled, you need to send CAL:SAVE command to store the new CAL data to non-volatile memory.

Parameter	Typical return
(none)	(none)
Saves calibration constants: CAL:SAVE	

CALibration:SECure:CODE <new passcode>

The command enters a new security passcode. To change the passcode, first unsecure the power supply using the old passcode. Then, enter the new passcode. The passcode can be set up to 9 digits.

Parameter	Typical return
<new code>	(none)
Sets the new security passcode to 12345: CAL:SEC:CODE 12345	

CALibration:SECure:STATe ON | OFF | 1 | 0, <passcode>
CALibration:SECure:STATe?

The command unsecures or secures the power supply for calibration, using the passcode specified by CALibration:SECure:CODE.

The query returns 0 (calibration unsecured) or 1 (calibration secured).

Parameter	Typical return
ON | OFF | 1 | 0, <passcode>	0 or 1
Secures the power supply for calibration: CAL:SEC:STAT ON, 0	

CALibration:STRing "<*string*>"
CALibration:STRing?

The command saves up to 40 characters of information, such as the last calibration date, the next calibration due date, or the power supply's serial number. You must unsecure the instrument before saving the string, but you can read the string regardless of the security status.

The query returns a quoted string.

Parameter	Typical return
"<*string*>"	"<string>"
Sets the string to "4/22/17": CAL:STR "4/22/17"	

CURRent Subsystem

Current commands program the output current and current protection functions. The SOURce keyword is optional in the following commands.

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <current> | MINimum | MAXimum | DEFault | UP | DOWN[, (@<chanlist>)]
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MINimum | MAXimum | DEFault,] [(@<chanlist>)]

The command sets the immediate current level of the output. Units are in amperes.

This command also increases or decreases the immediate current level using the "UP" or "DOWN" parameter by a predetermined amount. The command CURRent:STEP sets the amount of increase or decrease.

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

Parameter	Typical return
0 - maximum | MIN | MAX | DEF | UP | DOWN (Values are range and product-model dependent. See Programming Range) *RST MIN	<current level>
Sets the output current level to 3 A: `CURR 3`	

[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement] <current> | DEFault[, (@<chanlist>)]
[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault,] [(@<chanlist>)]

The command sets the step size for current programming with the CURRent UP and CURRent DOWN commands.

To set the step size to the minimum resolution, set the step size to "DEFault". The CURR:STEP? DEF returns the minimum resolution of your instrument. The immediate current level increases or decreases by the value of the step size. For example, the output current will increase or decrease 10 mA if the step size is 0.01.

This command is useful when you program the power supply to the allowed minimum resolution. At *RST, the step size is the value of the minimum resolution.

returns a number in the form +n.nnnnnnnnE+nn for each channel specified.

Parameter	Typical return
minimum - maximum | DEF (The maximum value is dependent on the voltage rating of the power module. See Programming Range) *RST DEF	<current level>
Sets the output current step size to 3: `CURR:STEP 3`	

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <current>| MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the triggered current level of the power supply. The triggered level is a stored value that will be programmed when a Step transient is triggered. Units are in amperes.

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

Parameter	Typical return
0 - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range)	<current level>
*RST MIN	
Sets the triggered current to 1 A: CURR:TRIG 1	

[SOURce:]CURRent:MODE FIXed | STEP | LIST | ARB[, (@<chanlist>)]
[SOURce:]CURRent:MODE? [(@<chanlist>)]

The command determines what happens to the output current when the transient system is initiated and triggered.

The query returns the current mode. Multiple responses are separated by commas.

Mode	Description
FIXed	Nothing happens. The output current remains at its immediate value.
STEP	The output goes to the triggered level when a trigger occurs.
LIST	The output follows the arbitrary waveform values when a trigger occurs. This mode still available for backward compatibility.
ARB	The output follows the arbitrary waveform values when a trigger occurs.

Parameter	Typical return
FIX | STEP | LIST	FIX, STEP, or LIST
*RST FIX	
Sets the current mode to Step: CURR:MODE STEP	

[SOURce:]CURRent:PROTection[:LEVel][:AMPLitude] <current> | MINimum | MAXimum[, (@<chan-list>)]
[SOURce:]CURRent:PROTection[:LEVel][:AMPLitude]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the level at which overcurrent protection trips, in amperes.

The query returns +n.nnnnnnnnE+nn in amperes.

Parameter	Typical return
1 - maximum | MIN | MAX (The maximum value is dependent on the current rating of the power module, the maximum OCP is 110% more than maximum setting current)	<current level>
*RST MAX	
Sets the level at which overcurrent protection trips to 20 A: CURR:PROT 20	

[SOURce:]CURRent:PROTection:CLEar[, (@<chanlist>)]

The command clears an overcurrent protection event.

Parameter	Typical return
(none)	(none)
Clears an overcurrent protection event: CURR:PROT:CLE	

[SOURce:]CURRent:PROTection:DELay[:TIME] <time> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]CURRent:PROTection:DELay[:TIME]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the over-current protection delay time in milliseconds. The over-current protection function will not be triggered during the delay time. After the delay time has expired, the over-current protection function will be active. This prevents momentary changes in output status from triggering the over-current protection function. Values up to 3600 seconds can be programmed, with a resolution of 1 millisecond.

The query returns the overcurrent protection delay in milliseconds in the form +n.nnnnnnnnE+nn.

Parameter	Typical return
0 – 3600 | MIN | MAX	<delay value>
*RST 0.05	
Sets the protection delay to 0.2 seconds: CURR:PROT:DEL 0.2	

Remarks

- The operation of over-current protection is affected by the setting of the current protection delay start event, which is specified by CURRent:PROTection:DELay:STARt.

[SOURce:]CURRent:PROTection:DELay:STARt SCHange | CCTRans | LEVel[, (@<chanlist>)]
[SOURce:]CURRent:PROTection:DELay:STARt? [(@<chanlist>)]

The command specifies the condition that starts the overcurrent protection delay timer.

Mode	Description
SCHange	Transitions into constant current mode are automatically ignored during a programmed settings change in voltage, current, or output state. At the end of the settings change, the delay timer starts, allowing additional protection delay time. There is no protection delay outside of these time windows.
CCTRans	The overcurrent protection delay timer is started by any transition of the output into constant current mode.
LEVel	The overcurrent protection delay timer is started when the output current level meets the set level.

The query returns SCH or CCTR.

Parameter	Typical return
SCH | CCTR | LEV	SCH, CCTR or LEV
*RST SCH	
Sets CCTRans as the current protection delay mode: CURR:PROT:DEL:STAR CCTR	

[SOURce:]CURRent:PROTection:STATe ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]CURRent:PROTection:STATe? [(@<chanlist>)]

The command enables or disables overcurrent protection, which causes the instrument to go into a protected state when the output status is in constant current mode for a time longer than the OCP delay. Output will be OFF after OCP is tripped. An overcurrent condition can be cleared with the CURR:PROT:CLE command after the condition that caused the OCP trip is removed.

The query returns 1 (ON) or 0 (OFF) for the over current protection state.

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
*RST OFF	
Enables the current protection state: CURR:PROT:STAT ON	

[SOURce:]CURRent:PROTection:TRIPped? [(@<chanlist>)]

The query indicates whether an overcurrent protection occurred (1) or not (0). This is reset to 0 by CURRent:PROTection:CLEar.

Parameter	Typical return
(none)	1 or 0
Indicates whether an overcurrent protection occurred: CURR:PROT:TRIP?	

[SOURce:]CURRent:RANGe HIGH | LOW[, (@<chanlist>)]
[SOURce:]CURRent:RANGe? [(@<chanlist>)]

The command sets the output current measurement range.

High range current allows the instrument to measure current up to 82.4 A, whereas the low range current allows the instrument to measure current up to 0.8 A (E36154A only). Refer table below for details.

Range	E36154A	E36155A
HIGH	up to 82.4 A	up to 41.2 A
LOW	up to 0.8 A	up to 0.6 A

The query returns the presently selected range.

Parameter	Typical return
LOW | HIGH	<current range>
*RST HIGH	
Set the output current range to HIGH: SOUR:CURR:RANG HIGH	

[SOURce:]DIGital:INPut:DATA?

The query reads the state of the digital control port. Returns the binary-weighted value of the state of pins 1 through 3 in bits 0 through 2 respectively.

Parameter	Typical return
(none)	<bit value>
Reads the state of the digital control port: DIG:INP:DATA?	

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

The command sets the output data on the digital control port when that port is configured for Digital I/O operation. The port has three 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	1	2	3
Bit number	0	1	2
Decimal value	1	2	4

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

Parameter	Typical return
0 - 7	<bit value>
*RST 0	
Programs pins 1 and 3 with bit number 0 = 1 and bit number 2 = 4: DIG:OUTP:DATA 5	

[SOURce:]DIGital:PIN<1-3>:FUNCtion DIO | DINPut | TOUTput | TINPut | FAULt | INHibit | ONCouple | OFFCouple
[SOURce:]DIGital:PIN<1-3>:FUNCtion?

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

The query returns the setting of pins 1, 2, or 3.

Function	Description
DIO	The pin is a general-purpose ground-referenced digital input/output. The output can be set with [SOURce:]DIGital:OUTPut:DATA.
DINPut	The pin is in digital input-only mode. The digital output data of the corresponding pin is ignored.
TOUTput	The pin is configured as a trigger output. When configured as a trigger output, the pin will only generate output triggers if the List transient system has been configured to generated trigger signals. See [SOURce:]STEP:TOUTput, [SOURce:]LIST:TOUTput:BOSTep, and [SOURce:]LIST:TOUTput:EOSTep.
TINPut	The pin is configured as a trigger input. When configured as a trigger input, the pin can be selected as the source for transient trigger signals. See TRIGger[:SEQuence]:SOURce.
FAULt	Applies only to pin 1. 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). 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	Applies only to pin 3. When pin 3 is configured as an inhibit input, a true signal at the pin will disable all output channels.
ONCouple	When configured as an On control, the pin will synchronize the output On state between power supplies. 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	When configured as an Off control, the pin will synchronize the output Off state between power supplies. 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.

Parameter	Typical return
DIO | DINP | TOUT | TINP | FAUL | INH | ONC | OFFC *RST DINP	DIO, DINP, TOUT, TINP, FAUL, INH, ONC, or OFFC
Sets pin 1 to Fault mode: DIG:PIN1:FUNC FAUL	

[SOURce:]DIGital:PIN<1-3>:POLarity POSitive | NEGative
[SOURce:]DIGital:PIN<1-3>:POLarity?

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

The query returns the polarity, POS or NEG.

Polarity	Description
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.

Parameter	Typical return
POS | NEG	POS or NEG
*RST POS	
Sets pin 1 to negative polarity: DIG:PIN1:POL NEG	

[SOURce:]DIGital:TOUTput:BUS[:ENABle] ON | OFF | 1 | 0
[SOURce:]DIGital:TOUTput:BUS[:ENABle]?

The command enables/disables BUS triggers on digital port pins. This allows a BUS trigger to be sent to any digital port pin that has been configured as a trigger output. The state is either ON (1) or OFF (0). A trigger is generated when the state is True (ON). A trigger is not generated when the state is False (OFF). A BUS trigger is generated using the *TRG command.

The query returns 0 (OFF) if the trigger signal will not be generated when a BUS trigger command occurs, and 1(ON) if a trigger signal will be generated when a BUS trigger command occurs.

Pins 1 to 3 must be configured as trigger outputs before they can generate a trigger signal. See [SOURce:]DIGital:PIN<1-3>:FUNCtion and [SOURce:]DIGital:PIN<1-3>:POLarity.

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
*RST OFF	
Enables BUS-generated trigger signals on the digital pins: DIG:TOUT:BUS ON	

DISPlay[:WINDow][:STATe] ON | OFF | 1 | 0
DISPlay[:WINDow][:STATe]?

The command turns the front-panel display off or on. When the display is turned off, outputs are not sent to the display and all annunciators except ERROR are disabled. The display state is automatically turned on when you return to the local mode. Press Lock/Unlock key to return to the local mode.

The query returns 0 (OFF) or 1 (ON).

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
*RST ON	
Turns the front panel display off: DISP OFF	

DISPlay[:WINDow]:TEXT[:DATA] "<string>"
DISPlay[:WINDow]:TEXT[:DATA]?

The command displays a message of up to 30 characters on the front panel. Additional characters are truncated.

The query returns the quoted string.

Parameter	Typical return
"<string>"	"<string>"
Sets the string displayed on the front panel to "Keysight": DISP:TEXT "Keysight"	

DISPlay[:WINDow]:TEXT:CLEar

The command clears the message displayed on the front panel.

Parameter	Typical return
none	none
Clears the message displayed on the front panel: DISP:TEXT:CLE	

DISPlay:LCD:BRIGhtness <brightness level>

The command sets the brightness level of the LCD display.

Parameter	Typical return
10 - 100	< brightness level>
*RST 100	
Sets the LCD display brightness to maximum:: DISP:LCD:BRIG 100	

FETCh Subsystem

Fetch commands return measurement data that has been previously acquired by a MEASurement command or triggered acquisition. FETCh queries do not generate a new measurement, but allow additional measurement calculations from the same acquired data. The data is valid until the next MEASure or INITiate command occurs.

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

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

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

The query returns the averaged measurement. Values are either in amperes, volts, or watts.The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<DC value>
Returns the averaged current: FETC:CURR? Returns the averaged voltage: FETC:VOLT? Returns the averaged power: FETC:POW?	

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

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

The query returns the RMS measurement (AC + DC). Values returned are either in amperes, or volts.The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<ACDC value>
Returns the measured RMS current: FETC:CURR:ACDC? Returns the measured RMS voltage: FETC:VOLT:ACDC?	

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

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

FETCh[:SCALar]:POWer:MAXimum? [(@<chanlist>)]

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

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

FETCh[:SCALar]:POWer:MINimum? [(@<chanlist>)]

The query returns the maximum or minimum value. Values returned are either in amperes, volts, or watts. The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<MIN value> <MAX value>
Returns the measured minimum current: FETC:CURR:MIN? Returns the measured minimum voltage: FETC:VOLT:MIN?) Returns the measured minimum power: FETC:POW:MIN?	
Returns the measured maximum current: FETC:CURR:MAX? Returns the measured maximum voltage: FETC:VOLT:MAX? Returns the measured maximum power: FETC:POW:MAX?	

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

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

FETCh:ARRay:POWer[:DC]? [(@<chanlist>)]

The query returns an array of measurement data. Values are either in amperes, volts, or watts.

The return format depends on the settings of the FORMat:BORDer and FORMat[:DATA] commands.
When the data format is set to ASCII, returned values are comma separated. When the data format is set to REAL, data is returned as single precision floating point values in definite length arbitrary block response format.

Parameter	Typical Return
(none)	<value> [,<value>] or <Block>
Returns the measured current array: FETC:ARR:CURR? Returns the measured voltage array: FETC:ARR:VOLT? Returns the measured power array: FETC:ARR:POW?	

Remarks

- The sampling rate is set by SENSe:SWEep:TINTerval. The position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet. The number of points returned is set by SENSe:SWEep:POINts.

FETCh[:SCALar]:DLOG? <number>[, (@<chanlist>)]

The query returns the specified number of logged data from channels. The data is fetched start from the data next to the last fetched data in the previous fetch.

The return format depends on the settings of the FORMat:BORDer and FORMat[:DATA] commands.
When the data format is set to ASCII, returned values are comma separated. When the data format is set to REAL, data is returned as single precision floating point values in definite length arbitrary block response format.

Parameter	Typical Return
0 - 65536	<value> [,<value>] or <Block>
Returns 10 logged data from channel 1 to 2. Channel 1 is enabled with voltage data logging only while channel 2 is enabled with both voltage and current data logging.: FETC:DLOG? 10	

FETCh[:SCALar]:ELOG? <maxrecords>[, (@<chanlist>)]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The query returns the most recent external datalog (Elog) records. Data must be read from the buffer periodically to avoid the buffer overflowing. Whenever data is read using FETCh:ELOG? then that buffer space is made available in the instrument for storing more acquired data.

Maxrecords specifies the maximum number of records of datalog data that the controller will return for each channel. A record is one set of voltage and current readings for one time interval. The exact format of a record depends on which functions have been enabled for external datalog sensing. If ALL datalog sense functions are enabled, one record will contain:
Current average, Current minimum, Current maximum, Voltage average, Voltage minimum, and Voltage maximum, in that order. If any of these sense functions are not enabled, that data is not part of the returned record.

The return format depends on the settings of the FORMat:BORDer and FORMat[:DATA] commands.
When the data format is set to ASCII, returned values are comma separated. When the data format is set to REAL, data is returned as single precision floating point values in definite length arbitrary block response format.

Parameter	Typical Return
1 to 65536	<value> [,<value>] or <Block>
Returns 100 data records.: FETC:ELOG? 100	

FORMat Subsystem

FORMat commands specify the format for transferring measurement data.

FORMat[:DATA]ASCII | REAL

FORMat[:DATA]?

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

Return Format	Description
ASCII	Returns data as ASCII bytes in numeric format as appropriate. The numbers are separated by commas.
REAL	Returns data in a definite length block as IEEE single precision floating point values. In this case the 4 bytes of each value can be returned in either big-endian or little-endian byte order, determined by the FORMat:BORDer setting.

Parameter	Typical Return
ASCII | REAL	ASCII or REAL
*RST ASCII	
Sets the data format to ASCII: FORM ASCII	

Remarks

- The data format is used by a small subset of SCPI queries that can return large quantities of data. These include: FETC:ARR:CURR? and FETC:ARR:VOLT?.

FORMat:BORDer NORMal | SWAPped

FORMat:BORDer?

The command specifies how binary data is transferred. This only applies when the FORMat:DATA is set to REAL.

Transfer Mode	Description
NORMal	Transfers data in normal order. The most significant byte is returned first, and the least significant byte is returned last (big-endian).
SWAPped	Transfers data in swapped-byte order. The least significant byte is returned first, and the most significant byte is returned last (little-endian).

Parameter	Typical Return
NORMal | SWAPped	NORM or SwAP
Sets the data transfer to Swapped: FORM:BORD SWAP	

Remarks

- The byte order is used when fetching real data from SCPI measurements. These include FETC:ARR:CURR? and FETC:ARR:VOLT?.

*CLS

The command clears all event registers, and the Status Byte register.

Parameter	Typical return
(none)	(none)
Clears all event registers, and the Status Byte register: *CLS	

*ESE <enable value>
*ESE?

The command enables bits in the Standard Event Enable register. The selected bits are then reported to the Status Byte.

The query returns the decimal value of the binary-weighted sum of all bits in the Standard Event enable register.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 3 and 4 in the enable register: *ESE 34	

Standard event status enable register

Bit	Bit Name	Decimal Value	Definition
0	Operation Complete	1	All commands before 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, a new command line was received before a previous query has been read, or both input and output buffers are full.
3	Device-Specific Error	8	A device-specific error, including a selftest error, calibration error or other device-specific error occurred. See Error Messages.
4	Execution Error	16	An execution error occurred. See Error Messages.
5	Command Error	32	A command syntax error occurred. See Error Messages.
6	not used	not used	0 is returned
7	Power On	128	Power has been cycled since the last time the event register was read or cleared.

*ESR?

The query returns the decimal value of the binary-weighted sum of all bits in the Standard Event register.

Parameter	Typical return
(none)	<bit value>
Read event status enable register: *ESR?	

*IDN?

The query returns the instrument's identification string. An example is shown below.

Keysight Technologies,E36154A,MY00000001,X.X.X-X.X.X-X.X

The four comma-separated fields are the manufacturer's name, the model number, the serial number, and the revision code. The first "X.X.X" in the revision codes is the firmware revision number for the controller firmware; the second is for the front panel firmware; the third is for the mainboard firmware.

Parameter	Typical return
(none)	<ASCII string with comma-separated fields>
Return the instrument's identification string: *IDN?	

*OPC
*OPC?

The command sets the “Operation Complete” bit (bit 0) of the Standard Event register after the command is executed.

The query returns 1 to the output buffer after the command is executed.

Parameter	Typical return
(none)	1
Sets the Operation Complete bit: *OPC	
Return a 1 when the command is complete: *OPC?	

*PSC 0 | 1
*PSC?

The command clears the Status Byte and the Standard Event register enable masks when power is turned on (*PSC 1). When *PSC 0 is in effect, the Status Byte and Standard Event register enable masks are not cleared when power is turned on.

The query returns a 0 (*PSC 0) or a 1 (*PSC 1).

Parameter	Typical return
0 | 1	0 or 1
Clear the Status Byte and Standard Event register enable masks: *PSC 1	

*RCL 0 | 1 | 2 | . . . | 8 | 9

The command recalls a previously stored state from one of ten non-volatile storage locations. To recall a stored state, you must use the same memory location used previously to store the state. You recall *RST states or values of the power supply from a memory location that was not previously specified as a storage location.

Parameter	Typical return
0 - 9	(none)
Recall the state from location 1: *RCL 1	

*RST

The command resets the instrument to its power-on default state. Refer to Factory Reset State for a complete listing of the instrument's factory configuration. It does not clear any of the status registers or the error queue. It also does not affect any interface error conditions.

*RST also forces the ABORt commands. This cancels any output trigger actions presently in process and resets the WTG bits in the Status Questionable Instrument Summary Registers.

Parameter	Typical return
(none)	(none)
Reset the instrument: *RST	

*SAV 0 | 1 | 2 | . . . | 8 | 9

The command saves the current instrument state using one of ten non-volatile storage locations.

The instrument states includes:

- Voltage, current, OVP, OCP delay, and OCP delay start
- Voltage slew, output preference and output sense
- Output state
- Operation mode (Independent)
- Output on/off sequencing
- Arb settings
- Trigger settings
- Digital I/O output data and bus setting
- Data logger settings
- Scope settings

Saving a state overwrites the previous state (if any) stored in that location.

When shipped from the factory, storage locations 0 through 9 are empty.

Parameter	Typical return
0 - 9	(none)
Save the state to location 1: *SAV 1	

*SRE <enable value>
*SRE?

The command enables the bits in the Status Byte Enable register.

The query returns the decimal value of the binary-weighted sum of all bits set in the register.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 3 and 4 in the enable register: *SRE 34	

*STB?

The query queries the Status Byte Summary register and returns the same result as a serial poll but the “Request Service” bit (bit 6) is not cleared if a serial poll has occurred.

Parameter	Typical return
(none)	<bit value>
Read the status byte: *STB?	

*TRG

The command generates an event trigger to the trigger system when the trigger system has a BUS (software) trigger as its trigger source (TRIG:SOUR BUS). If the trigger system is not initiated, the *TRG command is simply ignored.

Parameter	Typical return
(none)	(none)
Generates an immediate trigger: *TRG	

*TST?

The query returns a 0 if the self-test passes or a non-zero value if it fails. If the self-test fails, the instrument also generates an error message with additional information on why the test failed. Use SYSTem:ERRor? to read the error queue. See SCPI Error Messages for more information.

Parameter	Typical return
(none)	0 or 1
Performs an instrument self-test: *TST?	

*WAI

The command waits for all pending operations to complete before executing any additional remote interface commands. This command is used only in the triggered mode to wait for a pending delayed trigger.

Parameter	Typical return
(none)	(none)
Performs an instrument self-test: *WAI	

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

The command initiates the measurement trigger system. When a measurement trigger is initiated, an event on a selected trigger source causes the specified triggering action to occur. If a trigger occurs before the trigger system is ready for it, the trigger will be ignored. Check the WTG-meas bit in the operation status register to know when the instrument is ready to receive a trigger after initiating.

Parameter	Typical return
(@<chanlist>)	(none)
Initiates the trigger system: INIT:ACQ	

Remarks

- It takes a few milliseconds for the power supply to be ready to receive a trigger signal after receiving the INITiate:ACQuire command, and it can take longer if the value of SENSe:SWEep:OFFSet is negative.
- Use the appropriate ABORt command to return the instrument to the idle state.

INITiate[:IMMediate]:DLOG <"filename">

The command initiates the data logging session. The filename in which to save the data should be the full path and filename. Forinternal memory data logging, you will have to specify the exact file name, which is either Internal:/log1.dlog or Internal:/log2.dlog. Any other filename is not accepted for the internal data logging session.

Parameter	Typical return
<"filename">	(none)
Sets the setting to log the data as “log_1.dlog” file in the drive “External”: INIT:DLOG "External:/log_1.dlog"	

INITiate[:IMMediate]:ELOG [(@<chanlist>)]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command initiates the external data logging. When the external data log is initiated, an event on a selected external data log trigger source starts the data log.

Parameter	Typical return
(none)	(none)
Initiates an external data log”: INIT:ELOG	

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

The command initiates the scope data measurement trigger system. When a measurement trigger is initiated, an event on a selected trigger source causes the specified triggering action to occur. If a trigger occurs before the trigger system is ready for it, the trigger will be ignored. Check the WTG-meas bit in the operation status register to know when the instrument is ready to receive a trigger after initiating.

Parameter	Typical return
(@<chanlist>)	(none)
Initiates the scope data measurement trigger system: INIT:SCOP:ACQ	

Remarks

- It takes a few milliseconds for the power supply to be ready to receive a trigger signal after receiving the INITiate:SCOPe:ACQuire command, and it can take longer if the value of SENSe:SWEep:SCOPe:OFFSet is negative.
- Use the appropriate ABORt command to return the instrument to the idle state.

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

The command initiates the transient trigger system. When initiated, an event on the selected trigger source causes the specified triggered action to occur. If a trigger occurs before the trigger system is ready for it, the trigger will be ignored. Check the WTG-meas bit in the operation status register to know when the instrument is ready to receive a trigger after initiating.

Parameter	Typical return
(@<chanlist>)	(none)
Initiates the transient trigger system: INIT:TRAN	

INITiate:CONTinuous:TRANsient ON | OFF | 1 | 0[, (@<chanlist>)]
INITiate:CONTinuous:TRANsient ? [(@<chanlist>)]

The command continuously initiates the transient trigger system. This allows multiple triggers to generate multiple output transients with no intermediate commands. The enabled state is ON (1); the disabled state is OFF (0). With continuous triggering disabled, the trigger system must be initiated for each trigger using the INITiate:TRANsient command.

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

Parameter	Typical return
ON | OFF | 1 | 0	1 or 0
*RST OFF	
Initiates the trigger system continuously: INIT:CONT:TRAN ON	

Remark

- ABORt:TRANsient does not abort continuous triggers if INITiate:CONTinuous:TRANsient ON has been programmed. In this case, the trigger system will automatically re-initiate.

OUTPut Subsystem

These commands control the output, power-on and protection clear function.

OUTPut[:STATe] ON | OFF | 1 | 0[, (@<chanlist>)]
OUTPut[:STATe]? [(@<chanlist>)]

The command enables or disables the output. The state of a disabled output is a condition of zero output voltage and zero source current. A query returns 0 if the output is off or 1 if it is on. At *RST, the output state is off.

The query returns the output state of the power supply. The returned value is “0” (OFF) or “1” (ON).

NOTE

If output sequencing is enabled, the query returns the configuration state instead of the actual output state. For example, if you have a 10 s output delay and query the output state right after you turn the output on, the query will return 1 (ON) even though the actual output will be off until the delay ends.

Enabling or disabling any coupled output causes all coupled outputs to turn on or off according to their user-programmed delays and programming levels. If one coupled channel trips (overvoltage, overcurrent, or over-temperature), the other coupled channels are not impacted.

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1
*RST OFF	
Turns the output off: OUTP OFF, (@1)	

OUTPut[:STATe]:COUPle:CHANNel ALL | NONE | CH1
OUTPut[:STATe]:COUPle:CHANNel?

The command specifies which output channels 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.

There can be only one set of coupled channels; setting a new coupling replaces an existing coupling.

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

NOTE **When this command is sent, all output channels go to the output OFF state. This parameter is non-volatile and not affected by *RST.**

Parameter	Typical return
ALL | NONE | CH1	<coupled channel>
*RST NONE	
Couples channel 1: OUTP:COUP:CHAN CH1	

OUTPut[:STATe]:DELay:FALL <delay> | MINimum | MAXimum[, (@<chanlist>)]
OUTPut[:STATe]:DELay:FALL? [MINimum|MAXimum,] [(@<chanlist>)]

This command sets the delay in seconds that the instrument waits before disabling the specified output. This allows multiple output channels to turn off in sequence. Each output will not turn off until its delay time has elapsed. This command effects on-to-off transitions including changes in the OUTPut[:STATe]. It does NOT affect transitions to off caused by protection functions. Delay times can be programmed from 0 to 3600 seconds with the resolution of one millisecond.

The query returns the parameter in the form +n.nnnnnnnnE+nn. The parameter returned is the programmed delay time.

Parameter	Typical Return
0 - 3600 | MIN | MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s before turning off the output: OUTP:DEL:FALL 0.5	

OUTPut[:STATe]:DELay:RISE <delay> | MINimum | MAXimum[, (@<chanlist>)]
OUTPut[:STATe]:DELay:RISE? [MINimum|MAXimum,] [(@<chanlist>)]

The command sets the delay in seconds that the instrument waits before enabling the specified output. This allows multiple output channels to turn on in sequence. Each output will not turn on until its delay time has elapsed. This command affects all off-to-on transitions including changes in the OUTPut[:STATe] as well as transitions due to OUTPut:PROTection:CLEar. Delay times can be programmed from 0 to 3600 seconds with the resolution of 1 millisecond.

The query returns the parameter in the form +n.nnnnnnnnE+nn. The parameter returned is the programmed delay time.

Parameter	Typical Return
0 - 3600 | MIN | MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s before turning on the output: OUTP:DEL:RISE 0.5	

OUTPut[:STATe]:PMODe VOLTage | CURRent[, (@<chanlist>)]
OUTPut[:STATe]:PMODe? [(@<chanlist>)]

The command sets the preferred mode for output on or output off transitions. It allows output state transitions to be optimized for either constant voltage or constant current operation. Turn-on and turn-off overshoots are minimized for the preferred mode of operation.

VOLTage minimizes output on/off voltage overshoots in constant voltage operation.
CURRent minimizes output on/off current overshoots in constant current operation.

The query returns VOLT or CURR.

Mode	Description
VOLTage	Output on/off voltage overshoots in constant voltage operation are minimized.
CURRent	Output on/off voltage overshoots in constant current operation are minimized.

Parameter	Typical return
VOLTage | CURRent	VOLT | CURR
*RST VOLTage	
Sets the preferred mode to CURRent: OUTP:PMOD CURR	

OUTPut:INHibit:MODE LATChing | LIVE | OFF
OUTPut:INHibit:MODE?

The command sets the operating mode of the Inhibit input (INH) digital pin. The inhibit function shuts down ALL the outputs in response to an external signal on the Inhibit input pin. The parameter is saved in non-volatile memory.

If an output channel has been turned off by INPut[: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 query returns LATC, LIVE, or OFF.

Mode	Description
LATChing	A transition to True on the Inhibit input disables all output, and they remain disabled until the Inhibit input goes False and the latched INH status bit is cleared by OUTPut:PROTection:CLEar or a front-panel protection clear.
LIVE	The enabled outputs follow the state of the Inhibit input. Outputs are disabled if Inhibit is true and enabled if Inhibit is false.
OFF	The Inhibit input is ignored.

Parameter	Typical Return
LATC | LIVE | OFF	LATC, LIVE, or OFF
*RST OFF	
Sets the Inhibit input to latching mode: OUTP:INH:MODE LATC	

OUTPUT PAIR OFF | PARallel
OUTPut:PAIR?

The command specifies the power supply operation mode. Coupling must not be used in when the instrument is operating in parallel.

The query returns the power supply operation mode.

Parameter	Typical Return
OFF | PAR	OFF or PAR
*RST OFF	
Specifies the power supply operation mode to parallel: OUTP:PAIR PAR	

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

The command clears the latch that disables the output due to an overvoltage or overcurrent condition. You must clear the conditions that cause the fault before executing this command. You can then restore the output to the state that existed before the fault condition occurred.

Parameter	Typical return
(none)	(none)
Clears the latched protection status: OUTP:PRO:CLE	

OUTPut:PON:STATe RST | RCL0 | RCL1 | RCL2 | .... | RCL8 | RCL9
OUTPut:PON:STATe?

This determines whether the power-on state is set to the *RST state (RST) or the state stored in one of the ten memory locations. Instrument states can be stored using the *SAV command. This parameter is saved in non-volatile memory.

Parameter	Typical Return
RST | RCL0 | RCL1 | RCL2 | RCL3 | RCL4 | RCL5 | RCL6 | RCL7 | RCL8 | RCL9	RST, RCL0, RCL1, RCL2, RCL3, RCL4, RCL5, RCL6, RCL7, RCL8 or RCL9
*RST RST	
Sets the power-on state to the *RST state: INP:PON:STAT RST	
Sets the power-on state stored in memory location 1 at power on: OUTP:PON:STAT RCL1	

LIST Subsystem

List commands program an output sequence of multiple current and voltage settings. These commands are similar to Arb User-Defined.

NOTE **For standard option, a comma-delimited list of up to 100 steps may be programmed. With Option E36150ADVU, you are able to programmed up to 512 steps.**

[SOURce:]LIST:COUNt <count> | MINimum | MAXimum | INFinity[, (@<chanlist>)]
[SOURce:]LIST:COUNt? [MINimum | MAXimum | INFinity,] [(@<chanlist>)]

The command sets the number of times that the list is executed before it is completed. The list count range is 1 to 9999.

The query returns the list count for each channel specified. Multiple responses are separated by commas. If a repeat count of 9.9E37 is returned, it means the list is set to repeat continuously.

Parameter	Typical return
1 - 9999 | MIN | MAX | INF	<list count>
*RST 1	
Sets the list count to 10: LIST:COUN 10	

Remarks

- Use the INFinity parameter to execute a list continuously.
- Use ABORt 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.
- This command has been superseded by ARB:COUNt command, but is still available for backward compatibility.

[SOURce:]LIST:CURRent[:LEVel] <value>{,<value >}[, (@<chanlist>)]
[SOURce:]LIST:CURRent[:LEVel]]? [(@<chanlist>)]

The command specifies the current setting for each list step in amperes. A comma-delimited list of up to 512 steps may be programmed.

The query returns the programmed current level in the form +n.nnnnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical return
0 - maximum (Values are range and product-model dependent. See Programming Range)	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step set to the minimum programmable value.	
Programs a current list containing 5 steps: LIST:CURR 5,4,3,2,1	

Remarks

- The order in which the current values are entered determines the sequence when the list executes. To create a valid list, the Voltage, Current, BOST, EOST, and Dwell lists must either all be the same length, or have a length of 1, which is interpreted as having the same length as the list with the maximum length.
- The command overwrites any previously programmed current list; it does not append to the previous list.
- This command has been superseded by ARB:CURRent:UDEFined command, but is still available for backward compatibility.

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

The query returns the number of points (steps) in the current list, not the point values. Multiple responses are separated by commas.

Parameter	Typical return
(none)	<points>
Returns the number of points in the current list: LIST:CURR:POIN?	

Remarks

- This query has been superseded by ARB:CURRent:UDEFined:POINts?, but is still available for backward compatibility.

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

The command specifies the dwell time for each list step. 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 0 through 3600 seconds with the resolution of 100 microsecond.

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

Parameter	Typical return
0 – 3600 *RST 0.001	<list value 1>, <list value 2>, <list value 3>...
Programs a dwell list containing 5 steps: LIST:DWEL 0.2,0.8,1.5,0.8,0.2	

Remarks

- At the end of the dwell time, the output state of the unit depends upon the [SOURce:]LIST:STEP program settings. The order in which the dwell values are entered determines the sequence when the list executes.
- To create a valid list, the Voltage, Current, BOST, EOST, and Dwell lists must either all be the same length, or have a length of 1, which is interpreted as having the same length as the list with the maximum length.

- This command overwrites any previously programmed dwell list; it does not append to the previous list.
- This command has been superseded by ARB:UDEFined:DWELl command, but is still available for backward compatibility.

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

The query returns the number of points (steps) in the dwell list, not the point values. Multiple responses are separated by commas.

Parameter	Typical return
(none)	<points>
Returns the number of points in the dwell list: LIST:DWEL:POIN?	

Remarks

- This query has been superseded by ARB:UDEFined:DWELl:POINts?, but is still available for backward compatibility.

[SOURce:]LIST:STEP AUTO | ONCE[, (@<chanlist>)]
[SOURce:]LIST:STEP? [(@<chanlist>)]

The command specifies how the list responds to triggers.

The query returns the list step setting. Multiple responses are separated by commas.

Step	Description
AUTO	The output automatically advances 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. This specifies a dwell-paced list.
ONCE	The output remains at the present step until a trigger advances it to the next step. Triggers that arrive during the dwell time are ignored. This specifies a trigger-paced list.

Parameter	Typical return
AUTO | ONCE	AUTO or ONCE
*RST AUTO	
Sets the list step setting to ONCE: LIST:STEP ONCE	

[SOURce:]LIST:TERMinate:LAST ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]LIST:TERMinate:LAST? [(@<chanlist>)]

The command specifies the output value when the list terminates. The state is either 1 (ON) or 0 (OFF). When ON, the output remains at the last step value, and that value becomes the IMMediate value when the list completes. When OFF, or when the list is aborted, the output returns to the settings that were in effect before the list started. The query returns 0 (OFF) or 1 (ON).

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
*RST OFF	
Sets the list to terminate output at the last step value: LIST:TERM LAST ON	

Remarks

- This command has been superseded by ARB:TERMinate:LAST command, but is still available for backward compatibility.

[SOURce:]LIST:TOUTput:BOSTep[:DATA] ON | OFF | 1 | 0{,ON | OFF | 1 | 0}[, (@<chanlist>)]
SOURce:]LIST:TOUTput:BOSTep[:DATA]? [(@<chanlist>)]

The command specifies which list steps generate a trigger signal at the beginning of the step (BOSTep). A comma-delimited list of up to 512 steps may be programmed. The state is either ON (1) or OFF (0).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for transients of other channels, and for digital port pins configured as trigger outputs.

The query returns 0 if no trigger is generated, and 1 if a trigger is generated. Multiple responses are separated by commas.

Parameter	Typical return
ON | OFF | 1 | 0 *RST 1 step with a value of OFF	<list value 1>, <list value 2>, <list value 3>...
Specifies that triggers will be generated at the beginning of the second step of a 5-step list: LIST:TOUT:BOST 1,1,1,1,1	

Remarks

- The order in which the BOSTep values are entered determines the sequence when the list executes.
- To create a valid list, the Voltage, Current, BOST, EOST, and Dwell lists must either all be the same length, or have a length of 1, which is interpreted as having the same length as the list with the maximum length.
- This command overwrites any previously programmed BOSTep list; it does not append to the previous list.
- This command has been superseded by ARB:UDEFined:BOSTep command, but is still available for backward compatibility.

[SOURce:]LIST:TOUTput:BOSTep:POINts? [(@<chanlist>)]

The query returns a comma-separated list of the number of points (steps) in the beginning of the step trigger list (BOSTep), not the point values for the specified channels. A comma-delimited list of up to 512 steps may be programmed. The state is either ON (1) or OFF (0).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for measurements and transients of other units, and for digital port pins configured as trigger outputs.

Parameter	Typical return
(none)	<points>
Returns the number of points in the BOSTep list: LIST:TOUT:BOST:POIN?	

Remarks

- This query has been superseded by ARB:UDEFined:BOSTep:POINts?, but is still available for backward compatibility.

[SOURce:]LIST:TOUTput:EOSTep[:DATA] ON | OFF | 1 | 0{,ON | OFF | 1 | 0}[, (@<chanlist>)]
[SOURce:]LIST:TOUTput:EOSTep[:DATA]? [(@<chanlist>)]

The command specifies which list steps generate a trigger signal at the end of the step (EOSTep). A comma-delimited list of up to 512 steps may be programmed. The state is either ON (1) or OFF (0).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for transients of other channels, and for digital port pins configured as trigger outputs.

The query returns 0 if no trigger is generated, and 1 if a trigger is generated. Multiple responses are separated by commas.

Parameter	Typical return
ON | OFF | 1 | 0 *RST 1 step with a value of OFF	<list value 1>, <list value 2>, <list value 3>...
Specifies that triggers will be generated at the end of the second step of a 5-step list: LIST:TOUT:EOST 1,1,1,1,1	

Remarks

- The order in which the EOSTep values are entered determines the sequence when the list executes.
- To create a valid list, the Voltage, Current, BOST, EOST, and Dwell lists must either all be the same length, or have a length of 1, which is interpreted as having the same length as the list with the maximum length.
- This command overwrites any previously programmed EOSTep list; it does not append to the previous list.
- This command has been superseded by ARB:UDEFined:EOSTep command, but is still available for backward compatibility.

[SOURce:]LIST:TOUTput:EOSTep:POINts? [(@<chanlist>)]

The query returns a comma-separated list of the number of points (steps) in the end of the step trigger list (EOSTep), not the point values for the specified channels.

Parameter	Typical return
(none)	<points>
Returns the number of points in the EOSTep list: LIST:TOUT:EOST:POIN?	

Remarks

- This query has been superseded by ARB:UDEFined:EOSTep:POINts?, but is still available for backward compatibility.

[SOURce:]LIST:VOLTage[:LEVel] <value> {,<value >}[, (@<chanlist>)]
[SOURce:]LIST:VOLTage[:LEVel]? [(@<chanlist>)]

The command specifies the voltage setting for each list step in volts. A comma-delimited list of up to 512 steps may be programmed.

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

Parameter	Typical return
0 - maximum (Values are range and product-model dependent. See Programming Range) *RST 1 step set to the minimum programmable value.	<list value 1>, <list value 2>, <list value 3>...
Programs a voltage list containing 3 steps: LIST:VOLT 20,10,5	

Remarks

- The order in which the voltage values are entered determines the sequence when the list executes.
- To create a valid list, the Voltage, Current, BOST, EOST, and Dwell lists must either all be the same length, or have a length of 1, which is interpreted as having the same length as the list with the maximum length.
- This command overwrites any previously programmed voltage list; it does not append to the previous list.
- This command has been superseded by ARB:VOLTage:UDEFined command, but is still available for backward compatibility.

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

The query returns the number of points (steps) in the voltage list, not the point values. Multiple responses are separated by commas.

Parameter	Typical return
(none)	<points>
Returns the number of points in the voltage list: LIST:VOLT:POIN?	

Remarks

- This query has been superseded by ARB:VOLTage:UDEFined:POINts?, but is still available for backward compatibility.

LXI:IDENtify[:STATe] ON | OFF | 1 | 0
LXI:IDENtify[:STATe?] ON | OFF | 1 | 0

The command allows you to set the property to ON to change the LXI status indicator to the "Identify" state. Setting this property OFF changes the LXI status indicator to "No Fault".

The query returns the state of the LXI status indicator.

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
Sets the LXI status indicator to "No Fault": `LXI:IDEN 0`	

LXI:MDNS[:STATe] ON | OFF | 1 | 0
LXI:MDNS[:STATe?] ON | OFF | 1 | 0

The command allows you to control the mDNS service.

The query returns the control of the mDNS service.

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
Turns on the mDNS service: `LXI:MDNS 1`	

MEASure Subsystem

Measure commands measure the output voltage, current, or power. The MEASure queries start a new measurement immediately. They are not synchronized to any trigger event.

Use the INITiate, TRIGger, and FETCh commands if a synchronized measurement is necessary.

MEASure[:SCALar]:CURRent[:DC]? [CH1] [(@<chanlist>)]

MEASure[:SCALar]:VOLTage[:DC]? [CH1] [(@<chanlist>)]

MEASure[:SCALar]:POWer[:DC]? [CH1] [(@<chanlist>)]

The query returns the averaged output measurement. Values are either in amperes, volts, or watts.The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<DC value>
Returns the averaged current: MEAS:CURR? Returns the averaged power: MEAS:POW? Returns the averaged voltage: MEAS:VOLT?	

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

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

The query returns the total RMS measurement (AC + DC). Values returned are either in amperes, or volts.The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<ACDC value>
Returns the measured RMS current: MEAS:CURR:ACDC? Returns the measured RMS voltage: MEAS:VOLT:ACDC?	

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

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

MEASure[:SCALar]:POWer:MAXimum? [(@<chanlist>)]

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

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

MEASure[:SCALar]:POWer:MINimum? [(@<chanlist>)]

The query returns the maximum or minimum values of a measurement. Values returned are either in amperes, volts, or watts. The reading is in the form +n.nnnnnnE+nn. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<MIN value> <MAX value>
Returns the measured minimum current: MEAS:CURR:MIN? Returns the measured minimum voltage: MEAS:VOLT:MIN? Returns the measured minimum power: MEAS:POW:MIN?	
Returns the measured maximum current: MEAS:CURR:MAX? Returns the measured maximum voltage: MEAS:VOLT:MAX? Returns the measured maximum power: MEAS:POW:MAX?	

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

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

MEASure:ARRay:POWer[:DC]? [(@<chanlist>)]

The query returns an array containing measurement of instantaneous output samples. Values are either in amperes, volts, or watts.

The sampling rate is set by SENSe:SWEep:TINTerval. The position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet:POINts.The number of points returned is set by SENSe:SWEep:POINts.

The return format depends on the settings of the FORMat:BORDer and FORMat[:DATA] commands.
When the data format is set to ASCII, returned values are comma separated. When the data format is set to REAL, data is returned as single precision floating point values in definite length arbitrary block response format.

Parameter	Typical Return
(none)	<value> [,<value>] or <Block>
Returns the measured current array MEAS:ARR:CURR? Returns the measured power array: MEAS:ARR:POW? Returns the measured voltage array: MEAS:ARR:VOLT?	

MEASure:ARRay:SCOPe:CURRent[:DC]? [(@<chanlist>)]

MEASure:ARRay:SCOPe:VOLTage[:DC]? [(@<chanlist>)]

MEASure:ARRay:SCOPe:POWer[:DC]? [(@<chanlist>)]

The query returns an array containing scope data measurement of instantaneous output samples. Values are either in amperes, volts, or watts.

The sampling rate is set by SENSe:SWEep:SCOPe:TINTerval. The position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:SCOPe:OFFSet:POINts.The number of points returned is set by SENSe:SWEep:SCOPe:POINts.

Parameter	Typical Return
(none)	<value> [,<value>] or <Block>
Returns the measured current array in scope data: MEAS:ARR:SCOP:CURR? Returns the measured power array in scope data: MEAS:ARR:SCOP:POW? Returns the measured voltage array in scope data: MEAS:ARR:SCOP:VOLT?	

MMEMory:EXPort:DLOG <"filename">

The command saves the logged data in CSV format into the path and filename specified in the <"filename">. This command does not change the saved path and filename settings.

Parameter	Typical return
<"filename">	(none)
Exports the logged data into “External:\datalog.csv”: MMEM:EXP:DLOG "External:\datalog.csv"	

SENSe:DLOG:FUNCtion:CURRent ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:DLOG:FUNCtion:CURRent? [(@<chanlist>)]

The command enables or disables current data logging.

The query returns the status (on or off) of the current data logging.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables current data logging: SENS:DLOG:FUNC:CURR 1	

SENSe:DLOG:FUNCtion:MINMax ON | OFF | 1 | 0
SENSe:DLOG:FUNCtion:MINMax?

The command enables or disables the logging of minimum and maximum values for each sample of data.

The query returns whether the logging of minimum and maximum values is enabled.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables the logging of the minimum and maximum for each sample of data: SENS:DLOG:FUNC:MINM 1	

SENSe:DLOG:FUNCtion:VOLTage ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:DLOG:FUNCtion:VOLTage? [(@<chanlist>)]

The command enables or disables voltage data logging.

The query returns the status (on or off) of the voltage data logging.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST ON	
Enables voltage data logging: SENS:DLOG:FUNC:VOLT 1	

SENSe:DLOG:OFFSet <offset percent>
SENSe:DLOG:PERiod?

The command specifies the datalog trigger offset as a percent of the total datalog duration. This lets you specify the percent of pre-trigger data that will be logged to the datalog file.

A percent of 0 means the trigger occurs at the beginning of the running datalog, while 100 means that the trigger occurs at the end of the running datalog. Any value between 0 and 100 can be set.

The query returns the trigger offset in percent.

Parameter	Typical return
0 - 100	<offset percent>
*RST 0	
Specifies a trigger offset of 50%: SENS:DLOG:OFFS 50	

SENSe:DLOG:PERiod <time> | MINimum | MAXimum
SENSe:DLOG:PERiod? [MINimum | MAXimum]

The command sets the sample period, the entered value is in seconds. For sample period, minimum is 10 ms, and it has to be in integral of 10 ms, maximum is 60 seconds. The entered value will be rounded to the nearest 10 ms integral.

With Option E36150ADVU and Option E36150ATMU enabled, the sample period can be set from 1 ms to 60 s.

The query returns the sample period in seconds.

Parameter	Typical return
Standard option: 0.01 - 60 | MIN | MAX	<time>
Option E36150ADVU and Option E36150ATMU enabled: 0.001 - 60 | MIN | MAX	
*RST 0.1	
Sets the sample period to 400 ms: SENS:DLOG:PER 0.4	

Remarks

- This command replaces the previous SENSe:DLOG:TINTerval command and should be used in new applications. SENSe:DLOG:TINTerval is still available for backward compatibility.

SENSe:DLOG:TIME <time> | MINimum | MAXimum
SENSe:DLOG:TIME? [MINimum | MAXimum]

The command sets the sample duration, the entered value is in seconds. For sample duration, the maximum is about 10,000 hours (for single output, depending on the memory size) and up to 5 MB of data.

The query returns the sample duration in seconds.

Parameter	Typical return
1 - 36,000,000 | MIN | MAX	<time>
*RST 30	
Sets the sample duration to 2 minutes: SENS:DLOG:TIME 120	

SENSe:DLOG:TINTerval <time> | MINimum | MAXimum
SENSe:DLOG:TINTerval? [MINimum | MAXimum]

This command sets the sample period, the entered value is in seconds. For sample period, minimum is 10 ms, and it has to be in integral of 10 ms, maximum is 60 seconds. The entered value will be rounded to the nearest 10 ms integral.

With Option E36150ADVU and Option E36150ATMU enabled, the sample period can be set from 1 ms to 60 s.

The query returns the sample period in seconds.

Parameter	Typical return
Standard option: 0.01 - 60 | MIN | MAX Option E36150ADVU and Option E36150ATMU enabled: 0.001 - 60 | MIN | MAX	<time>
*RST 0.1	
Sets the sample period to 400 ms: SENS:DLOG:TINT 0.4	

Remarks

- This command has been superseded by SENSe:DLOG:PERiod, but is still available for backward compatibility.

SENSe:ELOG:FUNCtion:CURRent ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:ELOG:FUNCtion:CURRent? [(@<chanlist>)]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command enables or disables Elog current data logging.

The query returns the status (on or off) of the Elog current data logging.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables Elog current data logging: SENS:DLOG:FUNC:CURR 1	

SENSe:ELOG:FUNCtion:CURRent:MINMax ON | OFF | 1 | 0
SENSe:ELOG:FUNCtion:CURRent:MINMax?

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command enables or disables the logging of minimum and maximum current for each sample of data.

The query returns whether the logging of minimum and maximum current is enabled.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables the logging of the minimum and maximum current for each sample of data: SENS:ELOG:FUNC:CURR:MINM 1	

SENSe:ELOG:FUNCtion:VOLTage ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:ELOG:FUNCtion:VOLTage? [(@<chanlist>)]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command enables or disables Elog voltage data logging.

The query returns the status (on or off) of the Elog voltage data logging.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST ON	
Enables Elog voltage data logging: SENS:DLOG:FUNC:VOLT 1	

SENSe:ELOG:FUNCtion:VOLTage:MINMax ON | OFF | 1 | 0
SENSe:ELOG:FUNCtion:CURRent:MINMax?

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command enables or disables the logging of minimum and maximum voltage for each sample of data.

The query returns whether the logging of minimum and maximum voltage is enabled.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables the logging of the minimum and maximum voltage for each sample of data: SENS:ELOG:FUNC:CURR:MINM 1	

SENSe:ELOG:PERiod <time> | MINimum | MAXimum
SENSe:ELOG:PERiod? [MINimum | MAXimum]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command sets the sample period of an ELog measurement, the entered value is in seconds. For sample period, minimum is 1 ms, and it has to be in integral of 1 ms, maximum is 60 seconds. The entered value will be rounded to the nearest 1 ms integral.

The query returns the sample period in seconds.

Parameter	Typical return
0.001 - 60 | MIN | MAX	<time>
*RST 0.1	
Sets the sample period to 400 ms: SENS:DLOG:PER 0.4	

SENSe:ELOG:RUNning? [(@<chanlist>)]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The query returns the status of the Elog.

Parameter	Typical return
(none)	0 or 1
Read the status of Elog: SENS:ELOG:RUN?	

SENSe:FUNCtion:CURRent ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:FUNCtion:CURRent? [(@<chanlist>)]

The command enables or disables the current measurement function.

The query returns the status (on or off) of the current function.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables current function: SENS:FUNC:CURR 1	

SENSe:FUNCtion:VOLTage ON | OFF | 1 | 0[, (@<chanlist>)]
SENSe:FUNCtion:VOLTage? [(@<chanlist>)]

The command enables or disables the voltage measurement function.

The query returns the status (on or off) of the voltage function.

Parameter	Typical return
ON | OFF | 1 | 0	(none)
*RST OFF	
Enables voltage function: SENS:FUNC:VOLT 1	

SENSe:SWEep:POINts <data point> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the number of points in a measurement. The number of points can be specified from 1 to 131,072. This command applies to both voltage and current measurement.

The number of samples (points) that can be specified depends on the number of measurement parameter selected. You can measure up to two parameters (voltage and current x 1 outputs).

1 parameters : up to 128K points

2 parameters: up to 64K points

Parameter	Typical return
1 - 131,072 | MIN | MAX	<data point>
*RST 30	
Specifies 2048 points: SENS:SWE:POIN 2048	

SENSe:SWEep:OFFSet:POINt <offset points> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:OFFSet:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the offset in a data sweep for triggered measurements. Positive values represent the delay after the trigger occurs but before the samples are acquired. Negative values represent data samples taken prior to the trigger.

Parameter	Typical return
-131,071 - 2,000,000,000 | MIN | MAX	<offset points>
*RST 0	
Specifies -2048 offset points: SENS:SWE:OFFS:POIN -2048	

SENSe:SWEep:TINTerval <time> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:TINTerval? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the time period between samples in seconds. Programmed values can range from 10 ms to 40,000 s. Note that the shortest time interval (fastest speed) that can be specified depends on the number of parameters that are being measured and the model that is doing the measuring. Values above 10 ms are rounded to the nearest 10 ms increment.

Parameter	Typical return
0.01 - 40,000 | MIN | MAX	<time interval>
*RST 0.01	
Specifies an interval of 1 s between points: SENS:SWE:TINT 1	

SENSe:SWEep:SCOPe:POINts <data point> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:SCOPe:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the number of points in a scope data measurement. The number of points can be specified from 1 to 131,072. This command applies to both voltage and current measurement.

The number of samples (points) that can be specified depends on the number of measurement parameter selected. You can measure up to two parameters (voltage and current x 1 outputs).

1 parameters : up to 128K points

2 parameters: up to 64K points

Parameter	Typical return
1 - 131,072 | MIN | MAX	<data point>
*RST 4096	
Specifies 2048 points: SENS:SWE:SCOP:POIN 2048	

SENSe:SWEep:SCOPe:OFFSet:POINt <offset points> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:SCOPe:OFFSet:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the offset in a data sweep for triggered scope data measurements. Positive values represent the delay after the trigger occurs but before the samples are acquired. Negative values represent data samples taken prior to the trigger.

Parameter	Typical return
-131,071 - 2,000,000,000 | MIN | MAX	<offset points>
*RST 0	
Specifies -2048 offset points: SENS:SWE:SCOP:OFFS:POIN -2048	

SENSe:SWEep:SCOPe:TINTerval <time> | MINimum | MAXimum[, (@<chanlist>)]
SENSe:SWEep:SCOPe:TINTerval? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the time period between samples in seconds for scope data measurement. Programmed values can range from 10 µs to 40,000 s. Note that the shortest time interval (fastest speed) that can be specified depends on the number of parameters that are being measured and the model that is doing the measuring. Values above 10 ms are rounded to the nearest 10 ms increment.

Parameter	Typical return
0.00001 - 40,000 | MIN | MAX	<time interval>
*RST 0.00002	
Specifies an interval of 1 s between points: SENS:SWE:SCOP:TINT 1	

SOURce Subsystem

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

Subsystems Using the Optional SOURce Keyword

Because SOURce subsystem commands are often used without the SOURce keyword, these commands are listed by their individual subsystems, below:

CURRent

DIGital

LIST

VOLTage

STATus Subsystem

Status commands let 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. Refer to SCPI Status Registers for more information

Instrument status is also programmed using the IEEE 488.2 Common commands: *CLS, *ESE, *ESR?, *OPC, *OPC?, *SRE, *STB? and *WAI discussed under IEEE 488.2 Common Commands.

STATus:OPERation[:EVENt]?

The query returns the value of the event register for the Operation Status group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and PTR filter. Reading the Operation Status Event register clears it.

Parameter	Typical return
(none)	<bit value>
Reads the operation status event register: STAT:OPER?	

Remarks

- *RST has no effect on this register.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:OPERation:CONDition?

The query returns the value of the condition register for the Operation Status group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Operation Status Condition register does not clears it.

Parameter	Typical return
(none)	<bit value>
Reads the operation status condition register: STAT:OPER:COND?	

Remarks

- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:OPERation:ENABle <enable value>
STATus:OPERation:ENABle?

The command sets bits in the enable register for the Operation Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the OPER (operation summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register.

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

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 9 in the enable register: STAT:OPER:ENAB 512	

Remarks

- *CLS does not clear the enable register, but does clear the event register.

STATus:OPERation:NTRansition <value>
STATus:OPERation:NTRansition?

The command sets and queries the value of the NTR (Negative-Transition) register.

This register serves as a polarity filter between the Operation Condition and Operation Event registers.

When a bit in the 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.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the NTR register: STAT:OPER:NTR 24	

Remarks

- 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 value returned is the binary-weighted sum of all bits set in the register.

STATus:OPERation:PTRansition <value>
STATus:OPERation:PTRansition?

The command sets and queries the value of the PTR (Positive-Transition) register.

This registers serves as a polarity filter between the Operation Condition and Operation Event registers.

When a bit in the 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.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the PTR register:: STAT:OPER:PTR 24	

Remarks

- 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 value returned is the binary-weighted sum of all bits set in the register.

STATus:PRESet

The command presets the OPERation and QUEStionable ENABle, PTRansition and NTRansition registers.

Operation register	Questionable register	Preset setting
STAT:OPER:ENAB	STAT:QUES:ENAB	all defined bits are disabled
STAT:OPER:NTR	STAT:QUES:NTR	all defined bits are disabled
STAT:OPER:PTR	STAT:QUES:PTR	all defined bits are disabled

Parameter	Typical return
(none)	(none)
Presets the Operation and Questionable registers: STAT:PRES	

STATus:QUEStionable:[:EVENt]?

The query returns the value of the event register for the Questionable Status group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and PTR filter. Reading the Questionable Status Event register clears it.

Parameter	Typical return
(none)	<bit value>
Reads the questionable status event register: STAT:QUES?	

Remarks

- *RST has no effect on this register.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:QUEStionable:CONDition?

The query returns the value of the condition register for the Questionable Status group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Questionable Status Condition register does not clear it.

Parameter	Typical return
(none)	<bit value>
Reads the condition register: STAT:QUES:COND?	

Remarks

- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- *RST clears this register, other than those bits where the condition still exists after *RST.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:QUEStionable:ENABle <value>
STATus:QUEStionable:ENABle?

The command sets bits in the enable register for the Questionable Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the QUES (questionable summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register.

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

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 4 in the questionable enable register: STAT:QUES:ENAB 16	

STATus:QUEStionable:NTRansition <value>
STATus:QUEStionable:NTRansition?

The command sets and queries the value of the NTR (Negative-Transition) register.

This register serves as a polarity filter between the Questionable Condition and Questionable Event registers.

When a bit in the 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.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the NTR register: STAT:QUES:NTR 24	

Remarks

- 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 value returned is the binary-weighted sum of all bits set in the register.

STATus:QUEStionable:PTRansition <value>
STATus:QUEStionable:PTRansition?

The command sets and queries the value of the PTR (Positive-Transition) register.

This registers serves as a polarity filter between the Questionable Condition and Questionable Event registers.

When a bit in the 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.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the PTR register:: STAT:QUES:PTR 24	

Remarks

- 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 value returned is the binary-weighted sum of all bits set in the register.

SYSTem:BEEPer[:IMMediate]

The command issues a single beep immediately.

Parameter	Typical return
(none)	(none)
Issues a single beep immediately: SYST:BEEP	

SYSTem:BEEPer:STATe ON | OFF | 1 | 0
SYSTem:BEEPer:STATe?

The command enables or disables the beeper.

The query returns 0 (OFF) or 1 (ON).

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
*RST ON	
Turns on the beeper: SYST:BEEP:STAT ON	

SYSTem:COMMunicate:RLSTate LOCal | REMote | RWLock
SYSTem:COMMunicate:RLSTate?

The command sets the power supply to remote or local mode. The LOCal parameter is the same as SYSTem:LOCal, the REMote parameter is the same as SYSTem:REMote, and the RWLock parameter is the same as SYSTemRWLock.

The query returns LOC, REM, or RWL.

Parameter	Typical return
LOC | REM | RWL	LOC, REM, or RWL
*RST LOC	
Sets the power supply to remote: SYST:COMM:RLST REM	

SYSTem:COMMunicate:TCPip:CONTrol?

The command returns the initial socket control connection port number. After the control port number is obtained, a control socket connection can be opened.

Parameter	Typical return
(none)	5000 (0 if sockets are not suported)
Queries the control connection port number: SYST:COMM:TCP:CONT?	

NOTE The control socket connection can only be used by a client to send a device clear to the instrument or to detect Service Request (SRQ) events.

Refer to "Using Sockets" in the *User's Guide* for more information.

SYSTem:DATE <yyyy>,<mm>,<dd>
SYSTem:DATE?

The command sets the date of the real time clock in year (yyyy), month (mm), and day (dd). The range of values for the year is from 2000 to 2099.

The query returns comma-seperated values that correspond to the year, month, and day. For example: +2020,+1,+26.

Parameter	Typical return
<yyyy>,<mm>,<dd>	<+yyyy,+mm,+dd>
Sets the date to November 27, 2019: SYST:DATE 2019,11,27	

SYSTem:ERRor[:NEXT]?

The query returns the power supply error queue of up to 20 errors. The power supply beeps once and turns on the front-panel ERR annunciator when an error has been detected. Up to 20 errors can be stored in the error queue. See Error Messages.

NOTE Errors are retrieved in first-in-first-out (FIFO) order. The ERR annunciator turns off after the last error is read. The power supply beeps once each time an error is generated.

If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with -350, "Queue overflow". No additional errors are stored until you remove errors from the queue. If no errors have occurred, the SYST:ERR? query returns +0, "No error".

The error queue is cleared when power has been off or after a *CLS (clear status) command has been executed. The *RST (reset) command does not clear the error queue.

Parameter	Typical return
(none)	<+0,"No error">
Reads and clears the first error in error queue: SYST:ERR?	

SYSTem:LOCal

The command places the power supply in the local mode. All front-panel keys are fully functional.

Parameter	Typical return
(none)	(none)
Sets the power supply in the local mode: SYST:LOC	

SYSTem:REMote

The command places the power supply into remote mode for remote operation. All front-panel keys are disabled except for the Lock/Unlock key. You can unlock the front-panel keys by holding the Lock/Unlock key for a few seconds.

Parameter	Typical return
(none)	(none)
Sets to remote mode: SYST:REM	

SYSTem:RWLock

The command places the power supply in the remote mode. This command is the same as SYSTem:REMote except that all front-panel keys are disabled including the [Lock/Unlock] key. You can unlock the front-panel keys by using SYSTem:LOCal.

Parameter	Typical return
(none)	(none)
Sets to remote mode disabling all front-panel keys: SYST:RWL	

SYSTem:SECurity:IMMediate

The command clears all user memory and reboots the instrument. This command is typically used to prepare the instrument for removal from a secure area. It sanitizes all user data by writing all zeros to flash memory and then performing a chip erase as per manufacturer's data sheet. Identification data (instrument firmware, model number, serial number, MAC address and calibration data) is not erased. After the data is cleared, the instrument is rebooted.

This procedure is not recommended for use in routine applications because of the possibility of unintended loss of data.

Parameter	Typical return
(none)	(none)
Sanitizes the power supply: SYST:SEC:IMM	

SYSTem:SET <block_data>
SYSTem:SET?

The command sets the instrument as defined by the data returned by SYSTem:SET? query.

Parameter	Typical return
<block_data> = The block data returned by SYSTem:SET? query.	#nN<instrument state> where the first digit after the # indicates the number of following digits. The following digits indicate the length of the data.

SYSTem:TIME <hh>,<mm>,<ss>
SYSTem:TIME?

The command sets the real time clock in hours (hh), minutes (mm), and seconds (ss). The values may range from 0,0,0 (midnight) to 23,59,59 (one second before midnight).

The query returns the real time clock in hours (hh), minutes (mm), and seconds (ss).

Parameter	Typical return
<0 - 23>,<0 - 59>,<0 - 59>	+<hh>,+<mm>,+<ss>
Sets the real time clock to 13:30:15: SYST:TIME 13,30,15	

SYSTem:VERSion?

The query returns the present SCPI version of the power supply. The returned value is a string in the form of YYYY.V where “YYYY” represent the year of the version, and the “V” represents the current version number of the SCPI.

Parameter	Typical return
(none)	"<version>"
Returns the SCPI version:: SYST:VERS?	

TRIGger Subsystem

Trigger commands control the transient and acquisition subsystems.

TRIGger:ACQuire[:IMMediate] [(@<chanlist>)]

The command triggers the measurement immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Generates an acquisition trigger immediately: TRIG:ACQ	

Remarks

- You must initiate the measurement trigger system before you can send any trigger.
- At acquire trigger completion, the WTG-meas bit in the Status Operation Condition register is cleared.

TRIGger:ACQuire:CURRent[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
TRIGger:ACQuire:CURRent[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the current triggered level of the output. Applies when the measurement trigger source is set to a level. Values are specified in amperes. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
0 - maximum | MIN | MAX (The maximum values are 102% of the rating of the power supply.)	<triggered level>
*RST MIN	
Sets the triggered current level to 3 A: TRIG:ACQ:CURR 3	

TRIGger:ACQuire:CURRent:SLOPe POSitive | NEGative[, (@<chanlist>)]
TRIGger:ACQuire:CURRent:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the measurement trigger source is set to a level.

Slope	Description
POSitive	Specifies a rising slope of the output signal.
NEGative	Specifies a falling slope of the output signal.

Parameter	Typical return
POS | NEG	POS or NEG
*RST POS	
Sets the current slope to falling edge: TRIG:ACQ:CURR:SLOP NEG	

TRIGger:ACQuire:SOURce BUS | CURRent1 | EXTernal | IMMediate | VOLTage1 | PIN<n>[, (@<chan-list>)]
TRIGger:ACQuire:SOURce? [(@<chanlist>)]

The command selects the trigger source for the acquisition system.

The query returns BUS, EXT, IMM, CURR1, VOLT1, or PIN<n>.

The following trigger sources can be selected:

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects all connector pins configured as trigger sources.
IMMediate	Generates a trigger as soon as the trigger system is INITiated.
CURRent1	Selects the output current level of channel 1
VOLTage1	Selects the output voltage level of channel 1
PIN<n>	Selects a digital port pin configured as a trigger input, where <n> indicates the pin number.

Parameter	Typical return
BUS | CURR1 | EXT | IMM | VOLT1 | PIN<1-3> *RST BUS	BUS, CURR1, EXT, IMM, VOLT1, or PIN<n>
Selects a current level on channel 1 as the acquire trigger: TRIG:ACQ:SOUR CURR1	

TRIGger:ACQuire:VOLTage[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
TRIGger:ACQuire:VOLTage[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the voltage triggered level of the output. Applies when the measurement trigger source is set to a level. Values are specified in volts. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
0 - maximum | MIN | MAX (The maximum values are 102% of the rating of the power supply.)	<triggered level>
*RST MIN	
Sets the triggered voltage level to 50 V: TRIG:ACQ:VOLT 50	

TRIGger:ACQuire:VOLTage:SLOPe POSitive | NEGative[, (@<chanlist>)]
TRIGger:ACQuire:VOLTage:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the measurement trigger source is set to a level.

Positive specifies a rising slope of the output signal.

NEGative specifies a falling slope of the output signal.

Parameter	Typical return
POSitive | NEGative	POS or NEG
*RST POS	
Sets the voltage slope to rising edge: TRIG:ACQ:VOLT:SLOP POS	

TRIGger:DLOG[:IMMediate]

The command triggers the internal data logger immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Triggers the data logger immediately: TRIG:DLOG	

Remarks

- You must initiate the data logger before you can send any trigger.

TRIGger:DLOG:CURRent[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
TRIGger:DLOG:CURRent[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the current triggered level of the data logger. Applies when the data logger trigger source is set to a level. Values are specified in amperes. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
0 - maximum | MIN | MAX (The maximum values are 102% of the rating of the power supply.)	<triggered level>
*RST MIN	
Sets the triggered current level to 3 A: TRIG:DLOG:CURR 3	

TRIGger:DLOG:CURRent:SLOPe POSitive | NEGative[, (@<chanlist>)]
TRIGger:DLOG:CURRent:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the data logger trigger source is set to a level.

Slope	Description
POSitive	Specifies a rising slope of the output signal.
NEGative	Specifies a falling slope of the output signal.

Parameter	Typical return
POS | NEG	POS or NEG
*RST POS	
Sets the current slope to falling edge: TRIG:ACQ:CURR:SLOP NEG	

TRIGger:DLOG:SOURce BUS | EXTernal | IMMediate | CURRent1 | VOLTage1 | PIN<1-3> | ARSKey | OOOKey
TRIGger:DLOG:SOURce?

The command selects the trigger source for the data logger.

The query returns BUS, EXT, IMM, CURR1, VOLT1, PIN<1-3>, ARSK, or OOOK.

The following trigger sources can be selected:

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects all connector pins that have been configured as trigger sources
IMMediate	Sets the trigger source to true. As soon as the data logger is INITiated, it will send the trigger immediately.
CURRent1	Selects the measured current level of channel 1.
VOLTage1	Selects the measured voltage level of channel 1.
PIN<n>	Selects a digital port pin configured as a trigger input where <n> indicates the pin number.
ARSKey	Selects the Arb Run/Stop key (equivalent to **[List Run/Stop]** on the front panel)
OOOKey	Selects the Output On/Off key.

Parameter	Typical return
BUS | EXT | IMM | CURR1 | VOLT1 | PIN<1-3> | ARSK | OOOK *RST IMM	BUS, EXT, IMM, CURR1, VOLT1, PIN<n>, ARSK, or OOOK
Selects remote interface trigger command as the data log trigger source: TRIG:DLOG:SOUR BUS	

TRIGger:DLOG:VOLTage[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]
TRIGger:DLOG:VOLTage[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the voltage triggered level of the data logger. Applies when the data logger trigger source is set to a level. Values are specified in volts. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
0 - maximum | MIN | MAX (The maximum values are 102% of the rating of the power supply.) *RST MIN	<triggered level>
Sets the triggered voltage level to 50 V: TRIG:ACQ:VOLT 50	

TRIGger:DLOG:VOLTage:SLOPe POSitive | NEGative[, (@<chanlist>)]
TRIGger:DLOG:VOLTage:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the data logger trigger source is set to a level.

Positive specifies a rising slope of the output signal.

NEGative specifies a falling slope of the output signal.

Parameter	Typical return
POSitive | NEGative	POS or NEG
*RST POS	
Sets the voltage slope to rising edge: TRIG:ACQ:VOLT:SLOP POS	

TRIGger:ELOG[:IMMediate]

NOTE **In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.**

The command triggers the Elog immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Triggers the Elog immediately: TRIG:ELOG	

Remarks

- You must initiate the data logger before you can send any trigger.

TRIGger:ELOG:SOURce BUS | EXTernal | IMMediate | CURRent1 | VOLTage1 | PIN<1-3> | ARSKey | OOOKey
TRIGger:DLOG:SOURce?

NOTE In order to use this command, Option E36150ADVU and Option E36150ATMU must be enabled.

The command selects the trigger source for the Elog.

The query returns BUS, EXT, IMM, CURR1, VOLT1, PIN<1-3>, ARSK, or OOOK.

The following trigger sources can be selected:

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects all connector pins that have been configured as trigger sources
IMMediate	Sets the trigger source to true. As soon as the data logger is INITiated, it will send the trigger immediately.
CURRent1	Selects the measured current level of channel 1.
VOLTage1	Selects the measured voltage level of channel 1.
PIN<n>	Selects a digital port pin configured as a trigger input where <n> indicates the pin number.
ARSKey	Selects the Arb Run/Stop key (equivalent to **[List Run/Stop]** on the front panel)
OOOKey	Selects the Output On/Off key.

Parameter	Typical return
BUS | EXT | IMM | CURR1 | VOLT1 | PIN<1-3> | ARSK | OOOK *RST IMM	BUS, EXT, IMM, CURR1, VOLT1, PIN<n>, ARSK, or OOOK
Selects remote interface trigger command as the Elog trigger source: TRIG:ELOG:SOUR BUS	

TRIGger[:TRANsient | SEQuence][:IMMediate] [(@<chanlist>)]

The command triggers the output immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Generates a transient trigger immediately: TRIG:TRAN	

Remarks

- You must initiate the trigger system before you can send any trigger.
- At transient trigger completion, the WTG-tran bit in the Status Operation Condition register is cleared.

TRIGger[:TRANsient | SEQuence]:DELay <value> | MINimum | MAXimum[, (@<chanlist>)]
TRIGger[:TRANsient | SEQuence]:DElay? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the time delay between the detection of an event on the specified trigger source and the start of any corresponding trigger action on the power supply output. Programmed value can range from 0 to 3600 seconds.

Parameter	Typical return
0 - 3600 | MIN | MAX	<delay value>
*RST 0	
Sets the trigger time delay to 0.1 s: TRIG:TRAN:DEL 0.1	

TRIGger[:TRANsient | SEQuence]:SOURce BUS | EXTernal | IMMediate | PIN1 | PIN2 | PIN3[, (@<chanlist>)]
TRIGger[:TRANsient | SEQuence]:SOURce? [(@<chanlist>)]

The command selects the trigger source for the output trigger system.

The query returns BUS, EXT, IMM or PIN<n>.

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.

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects ALL connector pins that have been configured as trigger sources.
IMMediate	Sets the trigger source to true. As soon as the output is INITiated, it will send the trigger immediately.
PIN<n>	Selects a digital port pin configured as a trigger input where <n> indicates the pin number.

Parameter	Typical return
BUS | EXT | IMM | PIN1 | PIN2 | PIN3	BUS, EXT, IMM, PIN1, PIN2, or PIN3
*RST BUS	
Selects digital port pin 2 as the output trigger source: TRIG:SOUR PIN2	

Triggering Commands

The instrument's triggering system allows you to change current and voltage output when a trigger is received. The typical process is:

1. Configure the triggered output levels by using CURRent:TRIGgered and VOLTage:TRIGgered.
2. Configure the voltage and current mode to Step by using VOLTage:MODE and CURRent:MODE.
3. Specify the trigger source, either BUS, EXTernal, IMMediate, or PIN<n>.
4. If you are using the BUS trigger source, you may choose to set a time delay between the detection of the trigger and the start of any corresponding output change.
5. Send an INITiate[:IMMediate]command. If the IMMediate source is selected, the selected output is set to the triggered level immediately. If the BUS trigger source is selected, the output is set to the triggered level after the instrument receives the *TRG command.

VOLTage Subsystem

Voltage commands program the output voltage and voltage protection functions. The SOURce keyword is optional in the following commands.

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>| MINimum | MAXimum | DEFault | UP | DOWN[, (@<chanlist>)]
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum | MAXimum | DEFault,] [(@<chanlist>)]

The command programs the immediate voltage level of the power supply. Units are in volts.

This command also increases or decreases the immediate voltage level using the "UP" or "DOWN" parameter by a predetermined amount. The command VOLTage:STEP sets the amount of increase or decrease.

The query returns the programmed voltage level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. MAX and MIN return the highest and lowest programmable voltage levels respectively for the selected range.

Parameter	Typical return
0 - maximum | MIN | MAX | DEF | UP | DOWN (Values are range and product-model dependent. See Programming Range)	<voltage level>
*RST MIN	
Sets the output voltage level to 20 V: VOLT 20	

[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement] <voltage> | DEFault[, (@<chanlist>)]
[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault,] [(@<chanlist>)]

The command sets the step size for current programming with the VOLTage UP and VOLTage DOWN commands.

To set the step size to the minimum resolution, set the step size to "DEFault". The VOLT:STEP? DEF returns the minimum resolution of your instrument. The immediate voltage level increases or decreases by the value of the step size. For example, the output voltage will increase or decrease 10 mV if the step size is 0.01.

This command is useful when you program the power supply to the allowed minimum resolution. At *RST, the step size is the value of the minimum resolution.

The query returns a number in the form +n.nnnnnnnnE+nn for each channel specified.

Parameter	Typical return
minimum - maximum | DEF (The maximum value is dependent on the voltage rating of the power module. See Programming Range)	<voltage level>
*RST DEF	
Sets the output voltage step size to 3: VOLT:STEP 3	

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <voltage>| MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MINimum | MAXimum,] [(@<chanlist>)]

The command programs the triggered voltage level of the power supply. The triggered level is a stored value that is transferred to the output when an output step is triggered. Units are in volts.

The query returns the programmed voltage level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. MAX and MIN return the highest and lowest programmable voltage levels respectively for the selected range.

Parameter	Typical return
0 - maximum | MIN | MAX (Values are range and product-model dependent. See Programming Range)	<voltage level>
*RST MIN	
Sets the triggered voltage level to 5 V: VOLT:TRIG 5	

[SOURce:]VOLTage:MODE FIXed | STEP | LIST | ARB[, (@<chanlist>)]
[SOURce:]VOLTage:MODE? [(@<chanlist>)]

The command determines what happens to the output voltage when the transient system is initiated and triggered.

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

Mode	Description
FIXed	Nothing happens. The output voltage remains at the immediate value.
STEP	The output goes to the triggered level when a trigger occurs.
LIST	The output follows the arbitrary waveform values when a trigger occurs. This mode still available for backward compatibility.
ARB	The output follows the arbitrary waveform values when a trigger occurs.

Parameter	Typical return
FIXed | STEP | LIST | ARB	FIX, STEP, LIST, or ARB
*RST FIX	
Sets the voltage mode of channel 1 to Step: VOLT:MODE STEP	

[SOURce:]VOLTage:PROTection[:LEVel][:AMPLitude] <voltage> | MINimum | MAXimum[, (@<chan-list>)]
[SOURce:]VOLTage:PROTection[:LEVel][:AMPLitude]? MINimum | MAXimum[, (@<chanlist>)]

The command sets the level at which overvoltage protection trips, in volts.

The query returns +n.nnnnnnnnE+nn in volts.

Parameter	Typical return
1 - maximum | MIN | MAX (The maximum value is dependent on the voltage rating of the power module, the maximum OVP is 110% more than maximum setting voltage) *RST MAX	<voltage level>
Sets the level at which overvoltage protection trips to 20 V: VOLT:PROT 20	

[SOURce:]VOLTage:PROTection:STATe ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]VOLTage:PROTection:STATe? [(@<chanlist>)]

The command enables or disables overvoltage protection, which causes the instrument to go into a protected state when the power supply status is in constant voltage mode for a time longer than the OVP delay. Output will be OFF after OVP is tripped. An overvoltage condition can be cleared with the VOLT:PROT:CLE command after the condition that caused the OVP trip is removed.

The query returns 1 (ON) or 0 (OFF) or the overvoltage protection state.

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST OFF	
Enable the current protection state: VOLT:PROT:STAT ON	

[SOURce:]VOLTage:PROTection:TRIPped?

The query indicates whether an overvoltage protection occurred (1) or not (0). This is reset to 0 by VOLTage:PROTection:CLEar.

Parameter	Typical return
(none)	1 or 0
Indicates whether an overvoltage protection occurred: VOLT:PROT:TRIP?	

[SOURce:]VOLTage:PROTection:CLEar [(@<chanlist>)]

The command clears an overvoltage protection event.

Parameter	Typical return
(none)	(none)
Clears an overvoltage protection event: VOLT:PROT:CLE	

[SOURce:]VOLTage:SENSe[:SOURce] INTernal | EXTernal[, (@<chanlist>)]
[SOURce:]VOLTage:SENSe[:SOURce]? [(@<chanlist>)]

The command specifies whether the power supply uses remote or local sensing.

The query returns the selected state of the remote sense relay.

Sense	Description
INTernal	Sets the remote sense relays to local sensing. The front panel remote sense terminals are internally connected to the output terminals. The 4 wire indicator is off.
EXTernal	Sets the remote sense relays to remote sensing. The front panel remote sense terminals are not internally connected to the output terminals and must be connected to the external load. The 4 wire indicator is on.

NOTE This command specifies whether the instrument uses remote or local sensing. The query returns 0 (INT) or 1 (EXT). The Internal setting closes a relay within the power supply to short the output and sense connectors. This means that only two wires are used, and remote sensing is disabled. The External setting opens the relay in order to separate the output and remote sensing inputs. The Internal setting displays 2w in the upper left corner of the display, and the External setting shows 4w in the upper left corner.

Parameter	Typical return
INT | EXT	INT or EXT
*RST INT	
Sets the sense mode to external (4-wire): VOLT:SENS:SOUR EXT	

[SOURce:]VOLTage:SLEW:FALLing[:IMMediate] <slew rate> | MINimum | MAXimum | INFinity[, (@<chanlist>)]
[SOURce:]VOLTage:SLEW:FALLing[:IMMediate]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the falling voltage slew rate. The slew rate is set in volts per second and affects all falling programmed voltage changes, including those due to the output state turning on or off. The slew rate can be set from 0.003 (for E36155A model) or 0.0015 (E36154A model) up to any value, however, if the value is more than the maximum slew rate, the DUT will slew based on the maximum slew rate. For very large values, the slew rate will be limited by the analog performance of the output circuit. The keywords MAX or INFinity set the slew rate to maximum.

Parameter	Typical Return
0 – 9.9E+37 | MIN | MAX | INFinity	<slew rate>
*RST MAX	
Sets the falling output slew rate to 5 V per second: VOLT:SLEW:FALL 5	

Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using VOLTage:SLEW:FALLing? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E+37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]VOLTage:SLEW:FALLing:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]VOLTage:SLEW:FALLing:MAXimum? [(@<chanlist>)]

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by the VOLTage:SLEW:FALLing command. Use VOLTage:SLEW:FALLing? MAX to query the maximum slew rate.

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1
*RST ON	
Enables the falling maximum slew rate: VOLT:SLEW:FALL:MAX ON	

Remark

- The VOLTage:SLEW:FALLing:MAX command is coupled to the VOLTage:SLEW:FALLing command. If the VOLTage:SLEW:FALLing command sets the slew rate to MAX or INFinity, VOLTage:SLEW:FALLing:MAX is enabled. If the slew rate is set to any other value, VOLTage:SLEW:FALLing:MAX is disabled.

[SOURce:]VOLTage:SLEW:RISing][:IMMediate] <slew rate> | MINimum | MAXimum | INFinity[, (@<chanlist>)]
[SOURce:]VOLTage:SLEW:RISing[:IMMediate]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the rising voltage slew rate. The slew rate is set in volts per second and affects all rising programmed voltage changes, including those due to the output state turning on or off. The slew rate can be set from 0.003 (for E36155A model) or 0.0015 (E36154A model) up to any value, however, if the value is more than the maximum slew rate, the DUT will slew based on the maximum slew rate. For very large values, the slew rate will be limited by the analog performance of the output circuit. The keywords MAX or INFinity set the slew rate to maximum.

Parameter	Typical Return
0 – 9.9E+37 | MIN | MAX | INFinity	<slew rate>
*RST MAX	
Sets the rising output slew rate to 5 V per second: VOLT:SLEW RIS 5	

Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using VOLTage:SLEW:RISing? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E+37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]VOLTage:SLEW:RISing:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]VOLTage:SLEW:RISing:MAXimum? [(@<chanlist>)]

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by the VOLTage:SLEW:RISing command. Use VOLTage:SLEW:RISing? MAX to query the maximum slew rate.

Parameter	Typical Return
ON | OFF | 1 | 0	0 or 1
*RST ON	
Enables the rising maximum slew rate: VOLT:SLEW:RIS:MAX ON	

Remark

- The VOLTage:SLEW:RISing:MAX command is coupled to the VOLTage:SLEW:RISing command. If the VOLTage:SLEW:RISing command sets the slew rate to MAX or INFinity, VOLTage:SLEW:RISing:MAX is enabled. If the slew rate is set to any other value, VOLTage:SLEW:RISing:MAX is disabled.

Edition 2, November 2023
Printed in Malaysia
E36151-90008
www.keysight.com