Programming Guide

E36200 Series Autoranging DC Power Supplies

KEYSIGHT TECHNOLOGIES

Notices 7
Copyright Notice **7**
Manual Part Number **7**
Edition **7**
Published by **7**
Warranty **7**
Technology Licenses **7**
U.S. Government Rights **8**
Third Party Licenses **8**
Waste Electrical and Electronic Equipment (WEEE) **8**
Declarations of Conformity **8**
Safety Information **9**
1 E36200 Series Remote Operation 10
Introduction to the SCPI Language **11**
Command Format Used in this Manual 12
Command Separators 12
Querying Parameter Settings 13
SCPI Command Terminators 13
IEEE-488.2 Common Commands 13
SCPI Parameter Types 14
Halting an Output in Progress 15
Programming Ranges and Output Identifiers **15**
Channels and Channel Lists **16**
Channel list parameter (E36233A and E36234A only) 16
SCPI Error Messages **17**
Execution error codes 17
Calibration error codes 24
Self-test error codes 24
Reset and Non-volatile Settings **25**
Reset (*RST) settings 25
Non-volatile settings 28
Non-volatile LAN settings 28
SCPI Status Registers **29**
What is an event register? 32
What is an enable register? 32
What is a multiple logical output? 32
The Questionable Status register 33
The Questionable Instrument register 33
The Questionable Instrument Summary register 33
The Standard Event register 34
The Status Byte summary register 34
Using Service Request (SRQ) and serial POLL 35
Using *STB? to read the Status Byte 35
Using the message available bit (MAV) 35
To interrupt your bus controller using SRQ 36
To determine when a command sequence is completed 36
Using *OPC to signal when data is in the output buffer 36
2 SCPI Programming 37
ABORt Subsystem **38**
ABORt (@<chanlist>) 38

ABORt:DLOG ... 38
APPLy Subsystem ... **39**
APPLy CH1 | CH2 [,<voltage> | DEFault | MINimum | MAXimum [,<current> | DEFault | MINimum | MAXimum]]APPLy? [CH1 | CH2] ... 39
CALibration Subsystem ... **40**
CALibration:ASAVE ON | OFF | 1 | 0CALibration:ASAVE? ... 40
CALibration:COUNt? ... 40
CALibration:CURRent[:DATA][:HIGH] <current>, (@<chanlist>) ... 40
CALibration:CURRent[:DATA]:LOW <current>, (@<chanlist>) ... 41
CALibration:CURRent:LEVel [:HIGH] MINimum | MAXimum, (@<chanlist>) ... 41
CALibration:CURRent:LEVel:LOW MINimum | MAXimum, (@<chanlist>) ... 41
CALibration:DATE "<string>"CALibration:DATE? ... 41
CALibration:SAVE ... 42
CALibration:SECure:CODE <new code> ... 42
CALibration:SECure:STATe ON | OFF | 1 | 0, <code>CALibration:SECure:STATe? ... 42
CALibration:STRing "<string>"CALibration:STRing? ... 42
CALibration:VOLTage[:DATA] <numeric value>, (@<chanlist>) ... 43
CALibration:VOLTage:LEVel MINimum | MAXimum, (@<chanlist>) ... 43
CURRent Subsystem ... **44**
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <current> | MINimum | MAXimum | DEFault | UP | DOWN, (@<chanlist>) [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? MINimum | MAXimum | DEFault, (@<chanlist>) ... 44
[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement] <current> | DEFault, (@<chanlist>)
[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault,] (@<chanlist>) ... 44
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <current>| MINimum | MAXimum, (@<chanlist>)
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MINimum | MAXimum,] (@<chanlist>) ... 45
[SOURce:]CURRent:MODE FIXed | STEP | LIST, (@<chanlist>)[SOURce:]CURRent:MODE? (<@chanlist>) ... 46
[SOURce:]CURRent:PROTection[:LEVel][:AMPLitude] <current> | MINimum | MAXimum, (@<chanlist>)
[SOURce:]CURRent:PROTection[:LEVel][:AMPLitude]? [MINimum | MAXimum,] (@<chanlist>) ... 46
[SOURce:]CURRent:PROTection:CLEar (@<chanlist>) ... 46
[SOURce:]CURRent:PROTection:DELay[:TIME] <time> | MINimum | MAXimum, (@<chanlist>)
[SOURce:]CURRent:PROTection:DELay[:TIME]? [MINimum | MAXimum,] (@<chanlist>) ... 47
[SOURce:]CURRent:PROTection:DELay:STARt SCHange | CCTRans, (@chanlist)[SOURce:]CURRent:PROTection:DELay:STARt? (@chanlist) ... 47
[SOURce:]CURRent:PROTection:STATe ON | OFF | 1 | 0, (@<chanlist>)[SOURce:]CURRent:PROTection:STATe? (@<chanlist>) ... 48
[SOURce:]CURRent:PROTection:TRIPped? ... 48
DIGital Subsystem ... **49**
[SOURce:]DIGital:INPut:DATA? ... 49
[SOURce:]DIGital:OUTPut:DATA <value>[SOURce:]DIGital:OUTPut:DATA? ... 49
[SOURce:]DIGital:PIN<1-3>:FUNCtion[SOURce:]DIGital:PIN<1-3>:FUNCtion? ... 50
[SOURce:]DIGital:PIN<1-3>:POLarity POSitive|NEGative[SOURce:]DIGital:PIN<1-3>:POLarity? ... 51
[SOURce:]DIGital:TOUTput:BUS[:ENABle] 0 | OFF | 1| ON[SOURce:]DIGital:TOUTput:BUS[:ENABle]? ... 51
DISPlay Subsystem ... **52**
DISPlay[:WINDow][:STATe] ON | OFF | 1 | 0DISPlay[:WINDow][:STATe]? ... 52
DISPlay[:WINDow]:TEXT[:DATA] "<string>"DISPlay[:WINDow]:TEXT[:DATA]? ... 52
DISPlay[:WINDow]:VIEW METER1 | METER2DISPlay[:WINDow]:VIEW? ... 52
FETCh Subsystem ... **53**
FETCh[:SCALar]:DLOG? <number>, (@<chanlist>) ... 53
IEEE-488 Subsystem ... **54**
*CLS ... 54

*ESE <enable value>*ESE? 54
*ESR? 54
*IDN? 55
*OPC*OPC? 55
*PSC 0|1*PSC? 55
*RCL 0 | 1 | 2 | . . . | 8 | 9 56
*RST 56
*SAV 0 | 1 | 2 | . . . | 8 | 9 57
*SRE <enable value>*SRE? 57
*STB? 57
*TRG 58
*TST? 58
*WAI 58
INITiate Subsystem 59
INITiate[:IMMediate] (@<chanlist>) 59
INITiate:CONTinuous ON | OFF | 1 | 0, (@<chanlist>)INITiate:CONTinuous ? (@<chanlist>) 59
INITiate[:IMMediate]:DLOG <"filename"> 59
INSTrument Subsystem 60
INSTrument[:SELect] CH1 | CH2INSTrument[:SELect]? 60
INSTrument:NSELect 1 | 2INSTrument:NSELect? 60
LIST Subsystem 61
[SOURce:]LIST:COUNt <count> | INFinity, (@<chanlist>)[SOURce:]LIST:COUNt? [MINimum | MAXimum | INFinity,] (@<chanlist>) 61
[SOURce:]LIST:CURRent[:LEVel] <value>{, <value >}, (@<chanlist>) [SOURce:]LIST:CURRent[:LEVel]]? (@<chanlist>) 61
[SOURce:]LIST:CURRent:POINts? (@<chanlist>) 62
[SOURce:]LIST:DWELl <value> {, <value >}, (@<chanlist>)[SOURce:]LIST:DWELl? (@<chanlist>) 62
[SOURce:]LIST:DWELl:POINts? (@<chanlist>) 62
[SOURce:]LIST:STEP AUTO | ONCE, (@<chanlist>)[SOURce:]LIST:STEP? (@<chanlist>) 63
[SOURce:]LIST:TERMinate:LAST ON | OFF | 1 | 0, (@<chanlist>)[SOURce:]LIST:TERMinate:LAST? (@<chanlist>) 63
[SOURce:]LIST:TOUTput:BOSTep[:DATA] <bool>{, <bool>}, (@<chanlist>)SOURce:]LIST:TOUTput:BOSTep[:DATA]? (@<chanlist>) 63
[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>) 64
[SOURce:]LIST:TOUTput:EOSTep[:DATA] <bool>{, <bool>}, (@<chanlist>)[SOURce:]LIST:TOUTput:EOSTep[:DATA]? (@<chanlist>) 65
[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>) 65
[SOURce:]LIST:VOLTage[:LEVel] <value> {, <value >}, (@<chanlist>)[SOURce:]LIST:VOLTage[:LEVel]? (@<chanlist>) 66
[SOURce:]LIST:VOLTage:POINts? (@<chanlist>) 66
LXI Subsystem 67
LXI:IDENtify[:STATe] ON | OFF | 1 | 0LXI:IDENtify[:STATe?] ON | OFF | 1 | 0 67
LXI:MDNS[:STATe] ON | OFF | 1 | 0LXI:MDNS[:STATe?] ON | OFF | 1 | 0 67
MEASure Subsystem 68
MEASure[:SCALar]:CURRent[:DC]? [CH1 | CH2,] (@<chanlist>) 68
MEASure[:SCALar]:VOLTage[:DC]? [CH1 | CH2,] (@<chanlist>) 68
MMEMory Subsystem 69
MMEMory:EXPort:DLOG <"filename"> 69
OUTPut Subsystem 70
OUTPut[:STATe] ON | 1 | OFF | 0, (@<chanlist>)OUTPut[:STATe]? (@<chanlist>) 70
OUTPut[:STATe]COUPle:CHANNel ALL | NONE | <list>OUTPut[:STATe]:COUPle:CHANNel? 70

OUTPut[:STATe]:DELay:FALL <delay>, (@<chanlist>)OUTPut[:STATe]:DELay:FALL? [MINimum | MAXimum,] (@<chanlist>) ... 71
OUTPut[:STATe]:DELay:RISE <delay>, (@<chanlist>)OUTPut[:STATe]:DELay:RISE? [MINimum | MAXimum,] (@<chanlist>) ... 71
OUTPut[:STATe]:PMODe VOLTage | CURRent, (@<chanlist>)OUTPut[:STATe]:PMODe? (@<chanlist>) ... 72
OUTPut:INHibit:MODE LATChing | LIVE | OFFOUTPut:INHibit:MODE? ... 72
OUTPut:PAIR OFF | PARallel | SERiesOUTPut:PAIR? ... 73
OUTPut:PON:STATe RST | RCL0 | RCL1 | RCL2 | . . . . | RCL8 | RCL9, (@<chanlist>)OUTPut:PON:STATe? ... 73
OUTPut:PROTection:CLEar (@<chanlist>) ... 73
OUTPut:RELay ON | OFF | 1 | 0, (@<chanlist>)OUTPut:RELay? (@<chanlist>) ... 74
OUTPut:TRACk[:STATe] ON | OFF | 1 | 0OUTPut:TRACk[:STATe]? ... 74
SENSe Subsystem ... **75**
SENSe:DLOG:FUNCtion:CURRent ON | OFF | 1 | 0, (@<chanlist>)SENSe:DLOG:FUNCtion:CURRent? ... 75
SENSe:DLOG:FUNCtion:MINM ON | OFF | 1 | 0 SENSe:DLOG:FUNCtion:MINM? ... 75
SENSe:DLOG:FUNCtion:VOLTage ON | OFF | 1 | 0, (@<chanlist>)SENSe:DLOG:FUNCtion:VOLTage? ... 75
SENSe:DLOG:PER <time>SENSe:DLOG:PER? ... 76
SENSe:DLOG:TIME <time>SENSe:DLOG:TIME? ... 76
SENSe:DLOG:TINTerval <time>SENSe:DLOG:TINTerval? ... 76
STATus Subsystem ... **77**
STATus:OPERation:CONDition? ... 77
STATus:OPERation[:EVENt]? ... 77
STATus:OPERation:ENABle <enable value>STATus:OPERation:ENABle? ... 77
STATus:PRESet ... 77
STATus:QUEStionable:CONDition? ... 78
STATus:QUEStionable:[:EVENt]? ... 78
STATus:QUEStionable:ENABle <enable value>STATus:QUEStionable:ENABle? ... 78
STATus:QUEStionable:INSTrument[:EVENt]? ... 78
STATus:QUEStionable:INSTrument:ENABle <enable value>STATus:QUEStionable:INSTrument:ENABle? ... 79
STATus:QUEStionable:INSTrument:ISUMmary<n>[:EVENt]? ... 79
STATus:QUEStionable:INSTrument:ISUMmary<n>:CONDition? ... 79
STATus:QUEStionable:INSTrument:ISUMmary<n>:ENABle <enable value>STATus:QUESTionable:INSTrument:ISUMmary<n>:ENABle? ... 80
SYSTem Subsystem ... **81**
SYSTem:BEEPer[:IMMediate] ... 81
SYSTem:BEEPer:STATe ON | OFF | 1 | 0SYSTem:BEEPer:STATe? ... 81
SYSTem:COMMunicate:RLSTate LOCal | REMote | RWLockSYSTem:COMMunicate:RLSTate? ... 81
SYSTem:COMMunicate:TCPip:CONTrol? ... 82
SYSTem:DATE <yyyy>, <mm>, <dd>SYSTem:DATE? ... 82
SYSTem:ERRor[:NEXT]? ... 83
SYSTem:LOCal ... 83
SYSTem:PERSona:MANufacturer "<string>" | DEFaultSYSTem:PERSona:MANufacturer?SYSTem:PERSona:MANufacturer? DEFault ... 84
SYSTem:PERSona:MODel E3623XA | E363XA | DEFaultSYSTem:PERSona:MODel?SYSTem:PERSona:MODel? DEFault ... 84
SYSTem:REMote ... 84
SYSTem:RWLock ... 85
SYSTem:SECurity:IMMediate ... 85
SYSTem:SET <block_data>SYSTem:SET? ... 85
SYSTem:TIME <hh>,<mm>,<ss>SYSTem:TIME? ... 86
SYSTem:VERSion? ... 86
TRIGger Subsystem ... **87**
TRIGger:DLOG[:IMMediate] ... 87

TRIGger:DLOG:SOURce BUS | EXTernal | IMMediate TRIGger:DLOG:SOURce? 87
TRIGger[:SEQuence]:DELay <seconds> | MINimum | MAXimum, (@<chanlist>)TRIGger[:SEQuence]:DELay? [MINimum | MAXimum] (@<chanlist>) 87
TRIGger[:SEQuence]:SOURce BUS | EXTernal | IMMediate | PIN1 | PIN2 | PIN3, (@<chanlist>)TRIGger [:SEQuence]:SOURce? (@<chanlist>) 88
Triggering Commands **89**
VOLTage Subsystem **90**
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>| MINimum | MAXimum | DEFault | UP | DOWN, (@<chanlist>)[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? MINimum | MAXimum | DEFault, (@<chanlist>) 90
[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement] <voltage> | DEFault, (@<chanlist>)
[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]? [DEFault,] (@<chanlist>) 91
91
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <voltage>| MINimum | MAXimum, (@<chanlist>)
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? MINimum | MAXimum, (@<chanlist>) 91
[SOURce:]VOLTage:MODE FIXed | STEP | LIST, (@<chanlist>) [SOURce:]VOLTage:MODE? (@<chanlist>) 92
[SOURce:]VOLTage:PROTection[:LEVel][:AMPLitude] <voltage> | MINimum | MAXimum, (@<chanlist>)
[SOURce:]VOLTage:PROTection[:LEVel][:AMPLitude]? MINimum | MAXimum, (@<chanlist>) 92
[SOURce:]VOLTage:PROTection:STATe ON | OFF | 1 | 0, (@<chanlist>)[SOURce:]VOLTage:PROTection:STATe? (@<chanlist>) 93
[SOURce:]VOLTage:PROTection:TRIPped? 93
[SOURce:]VOLTage:PROTection:CLEar (@<chanlist>) 93
[SOURce:]VOLTage:RANGe P8V | P25V | P20V | P50V | LOW | HIGH, (@<chanlist>)[SOURce:]VOLTage:RANGe? (@<chanlist>) 93
[SOURce:]VOLTage:SLEW:RISing[:IMMediate] <slew> | MINimum | MAXimum | INFinity, (@<chanlist>)
[SOURce:]VOLTage:SLEW:RISing[:IMMediate]? MINimum | MAXimum (@<chanlist>) 94
[SOURce:]VOLTage:SLEW:RISing:MAXimum <bool>, (@<chanlist>)
[SOURce:]VOLTage:SLEW:RISing:MAXimum? (@<chanlist>) 94
[SOURce:]VOLTage:SLEW:FALLing[:IMMediate] <slew> | MINimum | MAXimum | INFinity, (@<chanlist>)
[SOURce:]VOLTage:SLEW:FALLing[:IMMediate]? MINimum | MAXimum (@<chanlist>) 95
[SOURce:]VOLTage:SLEW:FALLing:MAXimum <bool>, (@<chanlist>)
[SOURce:]VOLTage:SLEW:FALLing:MAXimum? (@<chanlist>) 95
[SOURce:]VOLTage:SENSe[:SOURce] INTernal | EXTernal, (@<chanlist>)[SOURce:]VOLTage:SENSe [:SOURce]? (@<chanlist>) 96

Notices

Copyright Notice

© Keysight Technologies 2019, 2021

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

Manual Part Number

E36200-90008

Edition

Edition 1, September 2021

Published by

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

Warranty

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

Technology Licenses

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

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, copies of which may be found in http://www.opensource.apple.com/source/CommonCrypto/CommonCrypto-7/Source/.

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 E36200 Series Remote Operation

Introduction to the SCPI Language

Programming Ranges and Output Identifiers

Channels and Channel Lists

SCPI Error Messages

Reset and Non-volatile Settings

SCPI Status Registers

This chapter describes the remote operation for the E36200 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 power supply 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).

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 power supply 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 CH1,3.5,1.5
```

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 power supply 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.

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 power supply will accept OFF or 0. For a true condition, the power supply will accept ON or 1. When you query a boolean setting, the power supply 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 power supply's input and output buffers are cleared.

– The power supply 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
```

– The following statement shows how to send a device clear over the GPIB interface using the GPIB Command Library for C or QuickBASIC.

Programming Ranges and Output Identifiers

Output setting commands require a parameter for programming ranges and an output name or an output number as the identifier of each output. Most queries will return a parameter. The programming range for a parameter varies according to the selected output of the power supply. The following table shows the maximum voltage, current, and power that can be programmed for each model. The DEFault voltage is always 0 V.

NOTE **Refer to this table to identify parameters when programming the power supply.**

Model	Max voltage (V)	Max current (A)	Max power (W)
E36231A	30.9	20.6	200
E36232A	61.8	10.3	200
E36233A	Single/Series: 30.9 Parallel: 61.8	Single/Series: 20.6 Parallel: 41.2	Single: 200 Series/Parallel: 400
E36234A	Single/Series: 61.8 Parallel: 123.6	Single/Series: 10.3 Parallel: 20.6	Single: 200 Series/Parallel: 400

Channels and Channel Lists

The E36231A and E36232A models have a single output channel, referred to a CH1.

The E36233A and E36234A models have two output channels, referred to as CH1 and CH2 respectively.

Channel list parameter (E36233A and E36234A only)

You can also use the channel list parameter to address one or more channels. You can either list channels:

(@1)

(@2)

or you can list both channels:

(@1:2)

(@1,2)

A channel list always starts with an @ and is enclosed in parentheses.

For example, (@1) specifies channel 1 and (@1,2) specifies channels 1 and 2. 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.**

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?
-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	Numeric overflow 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
-134	Suffix too long A suffix for a numeric parameter contained too many characters.

Code	Text
-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.
-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

Code	Text
-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
-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

Code	Text
-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.
-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.

Code	Text
-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 One or more lists are not the same length.
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
562	Analog board (CH2) does not respond
564	Analog board (CH1) over temperature
565	Analog board (CH2) over temperature
567	Analog board (CH1) command timed out
568	Analog board (CH2) command timed out
600	Analog board (CH1) failed to enter boot loader
601	Analog board (CH2) failed to enter boot loader
603	Analog board (CH1) failed to comm (SEM Fail)
604	Analog board (CH2) failed to comm (SEM Fail)
606	Analog board (CH1) failed to comm (UNSPECIFIED)
607	Analog board (CH2) failed to comm (UNSPECIFIED)
610	Fan test failed
611	EEPROM load failed
612	EEPROM checksum failed
613	EEPROM save failed
614	Invalid serial number
615	Invalid MAC address
616	Front panel does not respond

Code	Text
720	CH2 is not allowed when serial/parallel/track is enabled CH2 is not allowed to be used/disabled when serial/parallel/track is enabled.
721	Calibration is not allowed when serial/parallel is enabled Calibration is not allowed to be used/disabled when serial/parallel/track is enabled.
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 Data can't be alter as trigger has already been initiated. Abort the trigger to change data.
736	Trig init disallow as channel is coupled with another channel that has trig initiated
737	This command is only supported in E363XA persona mode Not supported by E3623XA models.
738	This command is only supported in E3623XA persona mode Not supported by E363XA models.
739	Channel list is ignored by this command in E363XA persona mode Channel list is not supported by E363XA persona mode. The command is still executed on the selected channel but is not using the channel list provided.
742	Illegal operation. List system has started for CH1 or CH2 Stop the list before doing this operation.
743	Operation not allowed when in parallel/series mode Change the mode before doing this operation.
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.
800	CH1 and CH2 coupled by track system The OUTP:TRAC should be off when coupling between the CH1 output and CH2 output.
801	CH1 and CH2 coupled by trigger subsystem The CH1 output and CH2 output should be uncoupled to enable the tracking operation for those outputs.
900	Firmware update failed

Calibration error codes

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

Code	Text
577	Analog board (CH1) calibration failed
578	Analog board (CH2) calibration failed
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
591	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
531	(CH2) Analog bias output 15V test failed
533	(CH1) System ADC test failed
534	(CH2) System ADC test failed
536	(CH1) System DAC test failed
537	(CH2) System DAC test failed
539	(CH1) Incorrect Model number
540	(CH2) Incorrect Model number
542	(FP) Incorrect Model number

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
INSTrument:SELect	1
INSTrument:NSELect	1
INSTrument:COUPle	None
APPLy	Same as CURRent and VOLTage commands
OUTPut:STATe	OFF
OUTPut:STATe:DELay:FALL	0
OUTPut:STATe:DELay:RISE	0
OUTPut:STATe: PMODe	VOLT
OUTPut:TRACk:STATe	OFF
OUTPut:PAIR	OFF
LIST:VOLTage	MIN
LIST:CURRent	MIN
LIST:DWELl	0.01
LIST:COUNt	1
LIST:TERMinate:LAST	OFF
LIST:STEP	AUTO
LIST:TOUTput:BOSTep	OFF
LIST:TOUTput:EOSTep	OFF
CURRent	Output dependent value For E36232A/E36234A (60 V) models - MAX = 10.3 A - MIN = 0 A - DEF = 10 A for persona mode, 1 A for general mode For E36231A/E36233A (30 V) models - MAX = 20.6 A - MIN = 0 A - DEF = 20 A for persona mode, 2 A for general mode
CURRent:STEP	Minimum voltage calibration resolution
CURRent:TRIGgered	MIN (0) If no triggered level is programmed, the CURR level is returned.

SCPI command	Default value
CURRent:PROTection:STATe	OFF
CURRent:PROTection:DELay	50 ms
CURRent:PROTection:DELay:STARt	SCH
CURRent:MODE	FIX
VOLTage	MIN (0)
VOLTage:STEP	Minimum current calibration resolution
VOLTage:TRIGgered	MIN (0) If no triggered level is programmed, the VOLT-level is returned.
VOLTage:MODE	FIX
VOLTage:PROTection	Output dependent value (MAX)
VOLTage:RANGe	-
VOLTage:SENSe	Internal (0)
DIGital:OUTPut:DATA	0
DIGital:TOUTput:BUS	FALSE
INITiate:CONTinuous	FALSE
TRIGger:DELay	0
TRIGger:SOURce	BUS
DISPlay	ON
DISPlay:TEXT	“” (empty)
DISPlay:VIEW	METER2
SYSTem:BEEPer:STATe	ON
SYSTem:COMMunicate:RLSTate	-
LXI:IDENtify:STATe	-
LXI:MDNS:STATe	-
CALibration:CURRent	-
CALibration:CURRent:LOW	-
CALibration:CURRent:LEVel	-
CALibration:CURRent:LEVel:LOW	-
CALibration:VOLTage	-
CALibration:VOLTage:LEVel	-
CALibration:CURRent:LEVel:LOW	-
CALibration:VOLTage	-
CALibration:VOLTage:LEVel	-
CALibration:SECure:CODE	-
CALibration:SECure:STATe	OFF
CALibration:STRing	-

SCPI command	Default value
STATus:QUEStionable:ENABle	-
STATus:QUEStionable:INSTrument:ENABle	-
STATus:QUEStionable:INSTrument:ISUMmary	-
STATus:QUEStionable:INSTrument:ISUMmary:ENABle	-
SENSe:DLOG:FUNCtion:CURRent	All channels off
SENSe:DLOG:FUNCtion:MINM	Off
SENSe:DLOG:FUNCtion:VOLTage	All channels on
SENSe:DLOG:PER	0.2 s
SENSe:DLOG:TIME	30 s
SENSe:DLOG:TINTerval	0.2 s

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
Calibration password	0
Calibration date	-
GPIB Address	5
SYSTem:BEEPer:STATe	ON
DIAGnostic:LCD:BRIGhtness	100
SYSTem:DATE	-
SYSTem:TIME	-
SYSTem:PERSona:MODel	E3623XA, E363XA
SYSTem:PERSona:MANufacturer	Keysight Technologies

Non-volatile LAN settings

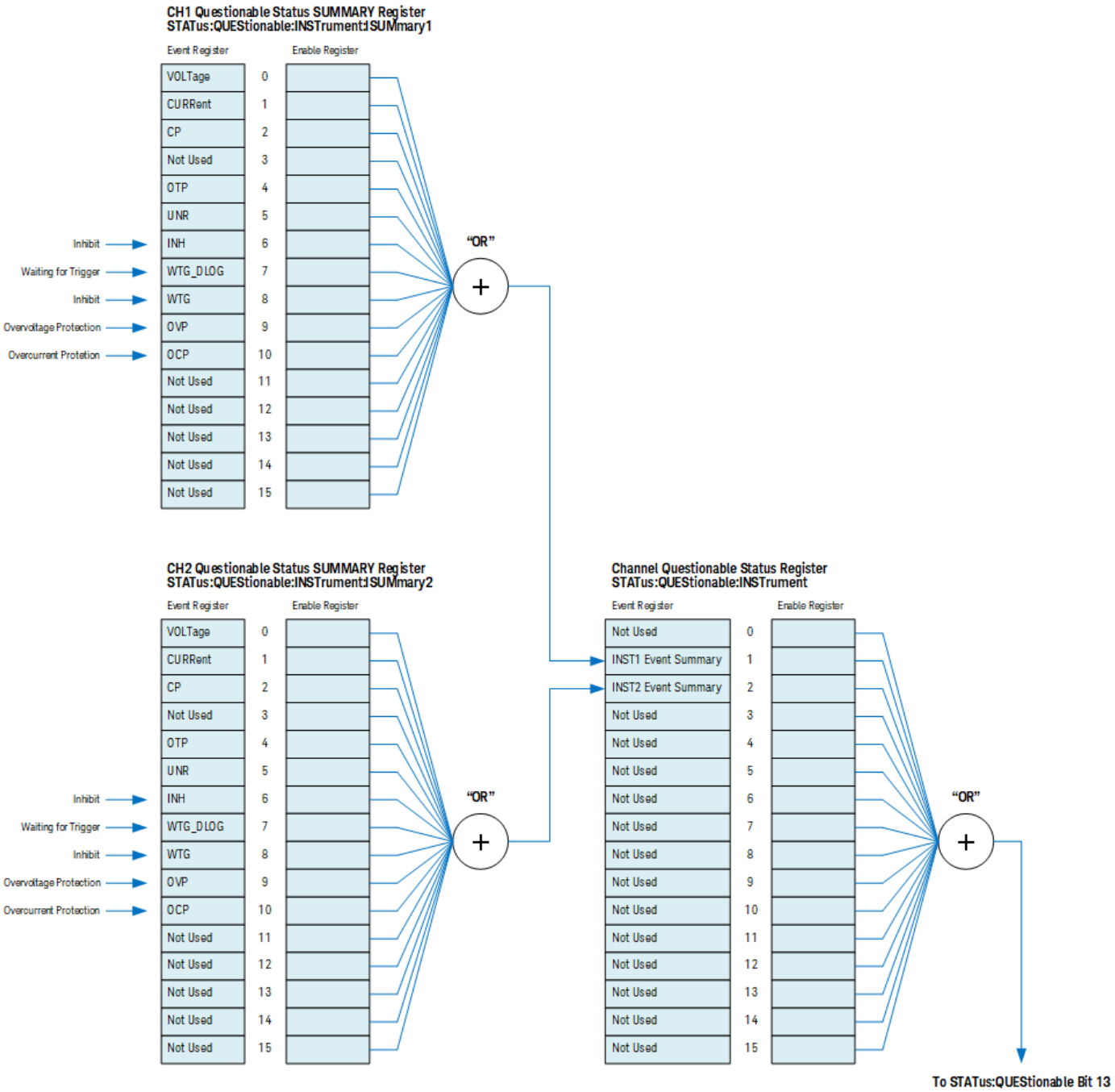
The instrument's non-volatile LAN settings are listed below:

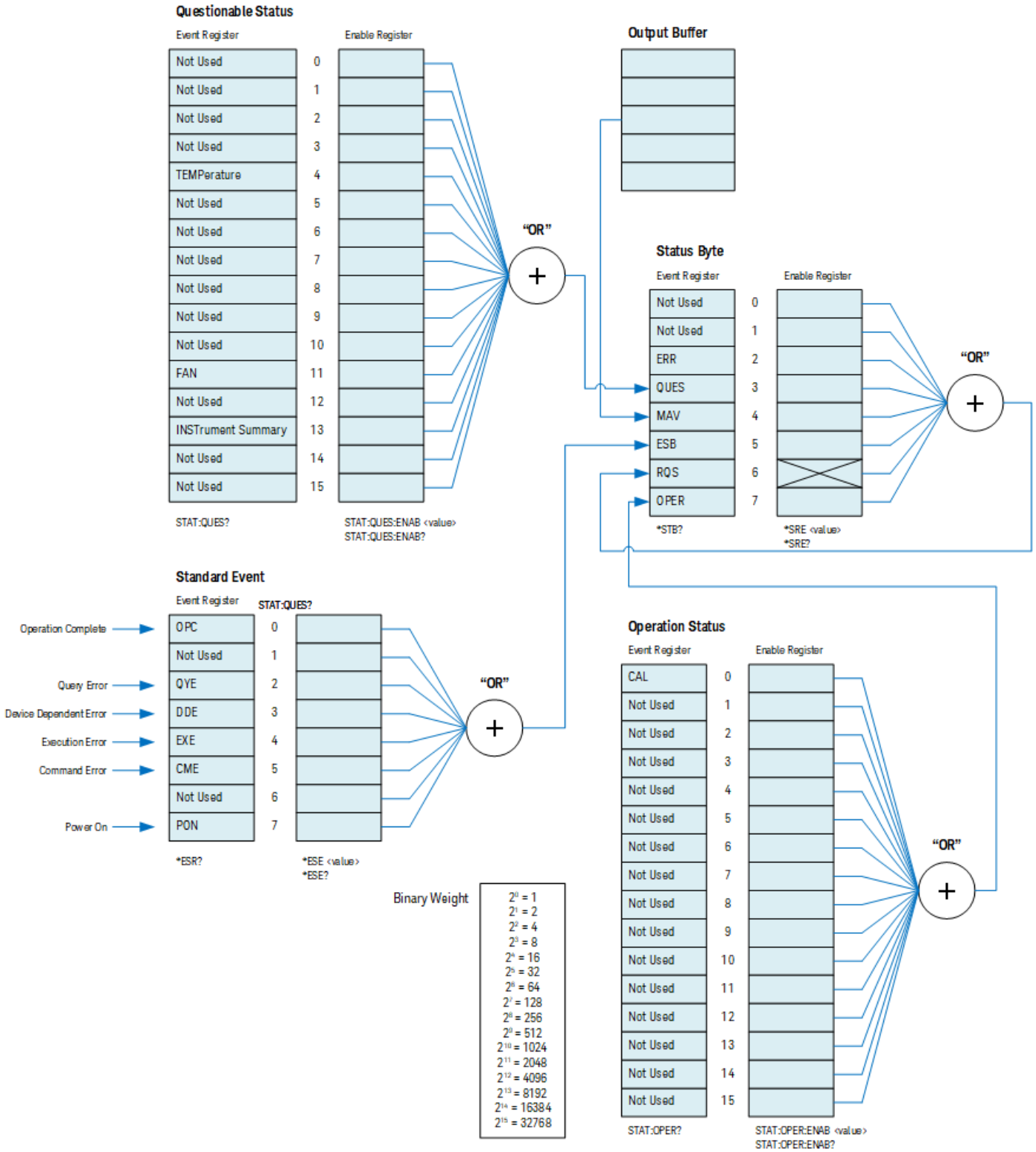
SCPI command	Default value
Get IP Address	Automatic
IP Address	192.168.10.1
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.1
Obtain DNS server from DHCP	Enabled
DNS server	Blank
Host name	K-x-xxxxx
Dynamic DNS naming service	Enabled
Domain name	Blank
Web password	Keysight

SCPI Status Registers

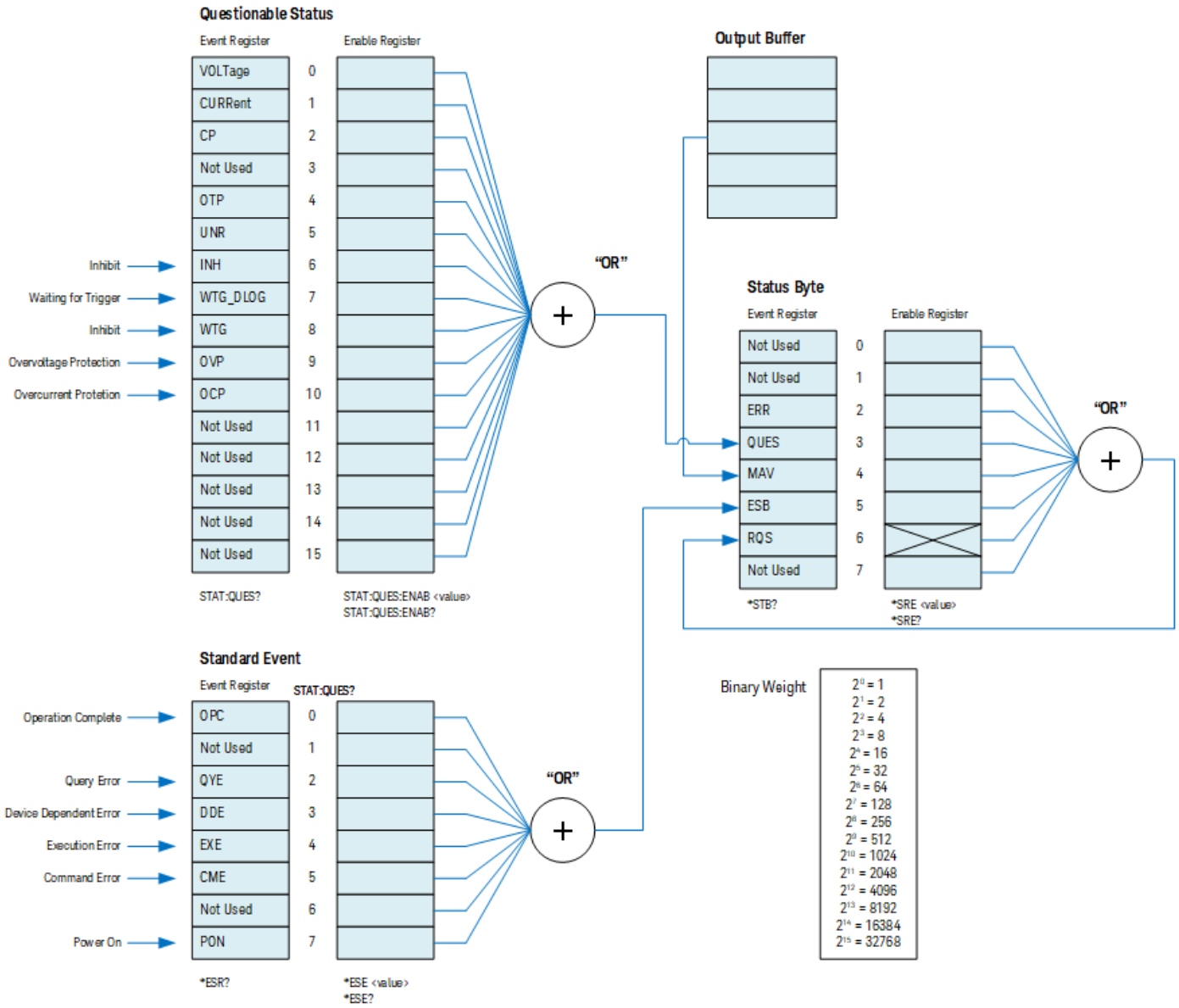
All SCPI instruments implement status registers in the same way. The status system records various instrument conditions in three register groups: the Status Byte register, the Standard Event register, and the Questionable Status register groups. The Status Byte register records high-level summary information reported in the other register groups. The figure below illustrates the SCPI status system used by the power supply.

E36233A and E36234A models:




Questionable Status
Event Register
Enable Register
Not Used 0
Not Used 1
Not Used 2
Not Used 3
TEMPerature 4
Not Used 5
Not Used 6
Not Used 7
Not Used 8
Not Used 9
Not Used 10
FAN 11
Not Used 12
INSTrument Summary 13
Not Used 14
Not Used 15
STAT:QUES?
STAT:QUES:ENAB <value>
STAT:QUES:ENAB?
"OR"
Output Buffer
Status Byte
Event Register
Enable Register
Not Used 0
Not Used 1
ERR 2
QUES 3
MAV 4
ESB 5
RQS 6
OPER 7
*STB?
*SRE <value>
*SRE?
"OR"
Standard Event
Event Register
STAT:QUES?
Operation Complete
OPC 0
Not Used 1
Query Error
QYE 2
Device Dependent Error
DDE 3
Execution Error
EXE 4
Command Error
CME 5
Not Used 6
Power On
PON 7
*ESR?
*ESE <value>
*ESE?
"OR"
Operation Status
Event Register
Enable Register
CAL 0
Not Used 1
Not Used 2
Not Used 3
Not Used 4
Not Used 5
Not Used 6
Not Used 7
Not Used 8
Not Used 9
Not Used 10
Not Used 11
Not Used 12
Not Used 13
Not Used 14
Not Used 15
STAT:OPER?
STAT:OPER:ENAB <value>
STAT:OPER:ENAB?
"OR"
Binary Weight
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024
2^11 = 2048
2^12 = 4096
2^13 = 8192
2^14 = 16384
2^15 = 32768

E36231A and E36232A models:



What is an event register?

An event register is a read-only register that reports defined conditions within the instrument. Bits in an event register are latched. Once an event bit is set, subsequent state changes are ignored. Bits in an event register are automatically cleared by a query of that register (such as *ESR? or STAT:QUES:EVEN?) or by sending the *CLS (clear status) command. A reset (*RST) or device clear will not clear bits in event registers. Querying an event register returns a decimal value of the binary-weighted sum of all bits set in the register.

What is an enable register?

An enable register defines which bits in the corresponding event register are logically ORed together to form a single summary bit. Enable registers are both readable and writable. Querying an enable register will not clear it. The *CLS (clear status) command does not clear enable registers but it does clear the bits in the event registers. To enable bits in an enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

What is a multiple logical output?

The two logical outputs of the power supply include an INSTrument summary status register and an individual instrument ISUMmary register for each logical output. The ISUMmary registers report to the INSTrument register, which in turn reports to bit 13 of the Questionable status register, as shown in the diagram below.

This status register configuration allows a status event to be cross-referenced by output and type of event. The INSTrument register indicates which output(s) have generated an event. The ISUMmary register is a pseudo-questionable status register for a particular logical output.

The Questionable Status register

The Questionable Status register provides information about unexpected instrument operation.

For E36231A and E36232A models:

See the details in the The Questionable Instrument Summary register.

For E36233A and E36234A models:

Bit 13 summarizes questionable outputs. For example, if one output is in constant voltage mode and loses regulation due to an overload, bit 13 is set (latched). Send the STAT:QUES? query to read the register. To make use of bit 13, you must first enable the registers you wish to summarize with bit 13. Send STAT:QUES:INST:ENAB 14 to enable the Questionable Instrument register. Then send STAT:QUES:INST:ISUM<n>:ENAB 3 for each supply to enable the Questionable Instrument Summary register, where n is 1 or 2.

Bit	Value	Name	Description
0-3	0	(unused)	Always set to 0.
4	16	(unused)	Always set to 0.
5-12	0	(unused)	Always set to 0.
13	8192	ISUM	Summary of QUES:INST and QUES:INST:ISUM registers.
14-15	0	(unused)	Always set to 0.

The Questionable Instrument register

The Questionable Instrument register provides information about unexpected operations for each of the two supplies. For example if CH1 is in the constant voltage mode and loses regulation, then bit 1 set indicates a possible overload in CH1. CH2 is reported as bit 2. Send the command STAT QUES:INST? to read the register. The STAT:QUES:INST:ISUM<n> registers must be enabled to make use of the Questionable Instrument register. Send STAT:QUES:INST:ISUM<n>:ENAB 3 to enable output n.

The Questionable Instrument Summary register

There are two Questionable Instrument Summary registers, one for each supply output. These registers provide information about voltage and current regulation. Bit 0 is set when the voltage becomes unregulated, and bit 1 is set if the current becomes unregulated. For example if a supply which is operating as a voltage source (constant voltage mode) momentarily goes to constant current mode, bit 0 is set to indicate that the voltage output is not regulated. To read the register for each supply, send STAT:QUES:INST:ISUM<n>?, where n is 1 or 2.

To determine the operating mode (CV, CC, or CP) for the power supply send STAT:QUES:INST:ISUM<n>:COND?, where n is 1 or 2 depending on the output. Bit 1 true indicates the output is in constant voltage mode, bit 2 true indicates constant power mode, both bits true indicates neither the voltage, current, or power is regulated, and all bits false indicates the outputs of the power supply are off.

The Questionable Status Event register is cleared when you execute the *CLS (clear status) command or query the event register using the STATus:QUEStionable[:EVENt]? query.

The Questionable Status Enable register is cleared by the STATus:QUEStionable:ENABle 0 command.

The Standard Event register

The Standard Event register reports the following types of instrument events: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or when an *OPC command is executed. Any or all of these conditions can be reported in the Standard Event Summary bit (ESB, bit 5) of Status Byte register through the enable register. To set the enable register mask, you write a decimal value to the register using the *ESE (Event Status Enable) command.

An error condition (Standard Event register bits 2, 3, 4, or 5) will always record one or more errors in the power supply's error queue. Read the error queue using the SYSTem:ERRor? command.

Bit	Value	Name	Description
0	1	OPC	Operation Complete. All commands prior to and including an *OPC command have been executed.
1	0	(unused)	Always set to 0.
2	4	QYE	Query Error. The power supply tried to read the output buffer but it was empty. Or, a new command line was received before a previous query had been read. Or, both the input and output buffers are full.
3	8	DDE	Device Error. A self-test or calibration error occurred.
4	16	EXE	Execution Error. An execution error occurred.
5	32	CME	Command Error. A command syntax error occurred.
6	0	(unused)	Always set to 0.
7	128	PON	Power On. Power has been turned off and on since the last time the event register was read or cleared

The Standard Event register is cleared when you execute the *CLS (clear status) command or you query the event register using the *ESR? (Event Status register) command.

The Standard Event Enable register is cleared when you execute the *ESE 0 command or you turn on the power and have previously configured the power supply using the *PSC 1 command.

For example, you must send the *ESE 24 (8 + 16) to enable DDE and EXE bits. Similarly, 24 is returned when you have queried the status of the Standard Event register and the DDE and EXE conditions have occurred.

The Status Byte summary register

The Status Byte summary register reports conditions from the other status registers. Query data waiting in the output buffer is immediately reported through the “Message Available” bit (bit 4) of Status Byte register. Bits in the summary register are not latched. Clearing an event register clears the corresponding Status Byte summary register bits. Reading all messages in the output buffer, including pending queries, clears the message available bit.

Bit	Value	Name	Description
0-2	0	(unused)	Always set to 0.
3	8	QUES	One or more bits are set in the questionable status register (bits must be “enabled” in the enable register).
4	16	MAV	Data is available in the power supply output buffer.
5	32	ESB	One or more bits are set in the standard event register (bits must be “enabled” in the enable register).
6	64	RQS	The power supply is requesting service (serial poll).
7	0	(unused)	Always set to 0.

The *CLS (clear status) command clears the Status Byte Summary Register. Querying the Standard Event register (*ESR?) clears only bit 5 in the Status Byte summary register. For example, 24 (8 + 16) is returned when you query the Status Byte register and the QUES and MAV conditions have occurred.

The Status Byte Enable register (Request Service) is cleared when you execute the *SRE 0 command or you turn on the power and have previously configured the power supply using the *PSC 1 command. For example, you must send the *SRE 96 (32 + 64) to enable ESB and RQS bits.

Using Service Request (SRQ) and serial POLL

You must configure your bus controller to respond to the IEEE-488 service request (SRQ) interrupt to use this capability. Use the Status Byte enable register (*SRE command) to select which summary bits will set the low-level IEEE-488 service request signal. When bit 6 (request service) is set in the Status Byte, an IEEE-488 service request interrupt message is automatically sent to the bus controller. The bus controller may then poll the instruments on the bus to identify which one requested service (the instrument with bit 6 set in its Status Byte).

The request service bit is cleared only by reading the Status Byte using an IEEE-488 serial poll or by reading the event register whose summary bit is causing the service request.

To read the Status Byte summary register, send the IEEE-488 serial poll message. Querying the summary register will return a decimal value which corresponds to the binary-weighted sum of the bits set in the register. Serial poll will automatically clear the “request service” bit in the Status Byte summary register. No other bits are affected. Performing a serial poll will not affect instrument throughput.

CAUTION **The IEEE-488 standard does not ensure synchronization between your bus controller program and the instrument. Use the *OPC? command to guarantee that commands previously sent to the instrument have completed. Executing a serial poll before a *RST, *CLS, or other commands have completed can cause previous conditions to be reported.**

Using *STB? to read the Status Byte

The *STB? (Status Byte) query is like a serial poll, but it is processed like any other query. The *STB? command returns the same result as a serial poll, but the “request service” bit (bit 6) is not cleared.

The*STB? query is not handled automatically by the IEEE-488 bus interface hardware and will be executed only after previous commands have completed. Polling is not possible using the *STB? query. The *STB? query does not clear the Status Byte summary register.

Using the message available bit (MAV)

You can use the Status Byte “message available” bit (bit 4) to determine when data is available to read into your bus controller. The power supply subsequently clears bit 4 only after all messages have been read from the output buffer.

To interrupt your bus controller using SRQ

1. Send a device clear message to clear the power supply's output buffer (e.g., CLEAR 705).
2. Clear the event registers with the *CLS (clear status) command.
3. Set up the enable register masks. Execute the *ESE command to set up the Standard Event register and the *SRE command for the Status Byte.
4. Send the *OPC? (operation complete query) command and enter the result to ensure synchronization.
5. Enable your bus controller's IEEE-488 SRQ interrupt.

To determine when a command sequence is completed

1. Send a device clear message to clear the power supply's output buffer (e.g., CLEAR 705).
2. Clear the event registers with the *CLS (clear status) command.
3. Enable the “operation complete” bit (bit 0) in the Standard Event register by executing the *ESE 1 command.
4. Send the *OPC? (operation complete query) command and enter the result to ensure synchronization.
5. Execute your command string to program the desired configuration, and then execute the *OPC (operation complete) command as the last command. When the command sequence is completed, the “operation complete” bit (bit 0) is set in the Standard Event register.
6. Use a serial poll to determine when bit 5 (standard event) is set in the Status Byte summary register. You could also configure the instrument for an SRQ interrupt by sending *SRE 32 (Status Byte enable register, bit 5).

Using *OPC to signal when data is in the output buffer

Generally, it is best to use the “operation complete” bit (bit 0) in the Standard Event register to signal when a command sequence finishes. This bit is set in the register by an *OPC command. If you send *OPC after a command that loads a query response in the instrument's output buffer, you can use the “operation complete” bit to determine when the message is available. However, if too many messages are generated before the *OPC command executes (sequentially), the output buffer will fill and the instrument will stop processing commands.

2 SCPI Programming

ABORt Subsystem
APPLy Subsystem
CALibration Subsystem
CURRent Subsystem
DIGital Subsystem
DISPlay Subsystem
FETCh Subsystem
IEEE-488 Subsystem
INITiate Subsystem
INSTrument Subsystem
LIST Subsystem
LXI Subsystem
MEASure Subsystem
MMEMory Subsystem
OUTPut Subsystem
SENSe Subsystem
STATus Subsystem
SYSTem Subsystem
TRIGger Subsystem
Triggering Commands
VOLTage Subsystem

This chapter describes the subsystem commands available to the E36200 Series autoranging DC power supplies.

ABORt Subsystem

ABORt (@<chanlist>)

The command clears any pending delayed trigger and returns the trigger system to idle. If INIT:CONT is enabled, ABORt does not turn off continuous triggers.This command also resets the WTG-tran bit in the Operation Condition Status register.

Parameter	Typical return
(@<chanlist>)	(none)
Aborts the triggered action of channel 1: ABOR (@1)	

ABORt:DLOG

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

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

NOTE For the E36231A and E36232A single channel models, the CH1 | CH2 parameter is not applicable. The CH1 | CH2 parameter is only applicable for the E36233A and E36234A dual channel models. You can identify each output by the output name (CH1 or CH2). The CH1 | CH2 parameter is optional. If it is not included in the syntax then the executed command will apply to the selected channel.

APPLy CH1 | CH2 [,<voltage> | DEFault | MINimum | MAXimum [,<current> | DEFault | MINimum | MAXimum]]
APPLy? [CH1 | CH2]

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

For the voltage and current parameters of the APPLy command, the ranges depend on the output currently selected. 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 for the selected output. MAX selects the highest voltage and current values allowed. The default voltage values are 0 volts for all outputs. The default current values are 10 A for the 60 V model and 20 A for the 30 V model in persona mode, whereas for general mode, default current values are 1 A for the 60 V model and 2 A for the 30 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. If you do not specify any value for the parameter, the APPLy command only selects the output specified and acts as the INSTrument command.

The query returns the power supply's present voltage and current values for each output as a quoted string as shown in the sample string below (the quotation marks are returned as part of the string). If any output identifier is not specified, the voltage and the current of the currently selected output are returned.

"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 specified output.

Output	Range of values
<voltage>	E36232A/E36234A (60 V) models = (MAX = 61.8 V | MIN = 0 V | DEF = 0 V) E36231A/E36233A (30 V) models = (MAX = 30.9 V | MIN = 0 V | DEF = 0 V)
<current>	E36232A/E36234A (60 V) models = (MAX = 10.3 A | MIN = 0 A | DEF = 10 A for persona mode, 1 A for general mode) E36231A/E36233A (30 V) models = (MAX = 20.6 A | MIN = 0 A | DEF = 20 A for persona mode, 2 A for general mode)

Parameter	Typical return
<voltage> | DEF | MIN | MAX, <current> | DEF | MIN | MAX *RST<DEF in Range of values>	<voltage>, <current>
Sets the maximum voltage and current of channel 1 output : APPL CH1, MAX, MAX	

NOTE The calibration can only be done one channel at a time, hence the channel list for all calibration commands will only accept one channel.

CALibration:ASAVE ON | OFF | 1 | 0
CALibration:ASAVE?

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

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. Since the value increments by one for each calibration point, a complete calibration for both outputs for the E36233A and E36234A models will increase the value by two counts.

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

CALibration:CURRent[:DATA][:HIGH] <current>, (@<chanlist>)

The command allows you to enter an output current value that you obtained by reading a digital multimeter (DMM). Select a calibration level (CAL:CURR:LEV) for the value being entered. This command can only be used when calibration is unsecured and output is ON.

Parameter	Typical return
<current>	(none)
Sets the output current value obtained by reading a DMM for example 5 A: CAL:CURR 5	

CALibration:CURRent[:DATA]:LOW <current>, (@<chanlist>)

The command allows you to enter an output current value that you obtained by reading a digital multimeter (DMM). This command is similar to CALibration:CURRent[:DATA][:HIGH], but it is only used for small current calibration. This command can only be used when calibration is unsecured and output is ON.

Parameter	Typical return
<current>	(none)
Sets the output current value obtained by reading a DMM for example 0.5 A: CAL:CURR:LOW 0.5	

CALibration:CURRent:LEVel [:HIGH] MINimum | MAXimum, (@<chanlist>)

Before using this command, you must select the output which is to be calibrated by using the INSTrument command. This command can only be used after calibration is unsecured. It sets the power supply to a calibration point that is entered with CALibration:CURRent [:DATA] command. During calibration, two points must be entered and the low-end point (MIN) must be selected and entered first.

Parameter	Typical return
MIN | MAX	(none)
Calibrates the output current for channel 1 to the minimum range: CAL:CURR:LEV MIN, (@1)	

CALibration:CURRent:LEVel:LOW MINimum | MAXimum, (@<chanlist>)

Before using this command, you must select the output which is to be calibrated by using the INSTrument command. This command can only be used after calibration is unsecured. This command is similar to CALibration:CURRent:LEVel [:HIGH], but it is only used for small current calibration. It sets the power supply to a calibration point that is entered with CALibration:CURRent [:DATA] command. During calibration, two points must be entered and the low-end point (MIN) must be selected and entered first.

Parameter	Typical return
MIN | MAX	(none)
Calibrates the output current for channel 1 to the minimum range: CAL:CURR:LEV:LOW MIN, (@1)	

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

The command stores the date that the power supply was last calibrated in nonvolatile 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 code>

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, <code>
CALibration:SECure:STATe?

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

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

Parameter	Typical return
ON | OFF | 1 | 0, <code>	1 or 0
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"	

CALibration:VOLTage[:DATA] <numeric value>, (@<chanlist>)

The command enters a voltage value of a selected output that you obtained by reading an external meter. Before using this command, you must first unsecure calibration and select a calibration level (CAL:VOLT:LEV) for the value being entered. Two successive values (one for each end of the calibration range) must be selected and entered. The power supply then computes new calibration constants and stores them in non-volatile memory.

Parameter	Typical return
<numeric value>	(none)
Sets the output voltage value (for channel 1), obtained by reading a DMM, for example 5 V: CAL:VOLT 5, (@1)	

CALibration:VOLTage:LEVel MINimum | MAXimum, (@<chanlist>)

The command sets the power supply to a calibration point that is entered with CAL:VOLT[:DATA] command. Before using this command, you must first unsecure calibration and use the INSTrument command to select the output to be calibrated. During calibration, the low-end point (MIN) must be selected and entered first, followed by the high-end point (MAX).

Parameter	Typical return
MIN | MAX	(none)
Calibrates the minimum voltage: CAL:VOLT:LEV MIN	

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

The command directly programs the immediate current level of the power supply in amperes. The immediate level is the current limit value of the output selected with the INSTrument command.

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

Model	Range of values
E36232A/E36234A (60 V) models	MAX = 10.3 A MIN = 0 A DEF = 10 A for persona mode, 1 A for general mode
E36231A/E36233A (30 V) models	MAX = 20.6 A MIN = 0 A DEF = 20 A for persona mode, 2 A for general mode

Parameter	Typical return
0 - maximum | MIN | MAX | DEF | UP | DOWN (The maximum value is dependent on the current rating of the power module)	<current level>
*RST <DEF in Default value>	
Sets the output current level to 3 A: CURR 3, (@1)	

[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.

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

Model	Range of values
E36232A/E36234A (60 V) models	MAX = 10.3 A MIN = 0 A DEF = 10 A for persona mode, 1 A for general mode
E36231A/E36233A (30 V) models	MAX = 20.6 A MIN = 0 A DEF = 20 A for persona mode, 2 A for general mode

Parameter	Typical return
0 - maximum | DEF (The maximum value is dependent on the current rating of the power module) *RST <DEF in Default value>	<current level>
Sets the output current step size to 3 A: CURR:STEP 3, (@1)	

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

The command programs the triggered current level of the specified output channel. Units are in amperes. The triggered level is a stored value that is transferred to the output when an output step is triggered.

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

Model	Range of values
E36232A/E36234A (60 V) models	Min (0 A) - 10 A
E36231A/E36233A (30 V) models	Min (0 A) to 20 A

Parameter	Typical return
0 - 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) *RST<DEF in Default values>	<current level>
Sets the triggered current to 1 A on channels 1 and 2: CURR:TRIG 1, (@1,2)	

[SOURce:]CURRent:MODE FIXed | STEP | LIST, (@<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 for each channel specified. Multiple responses are separated by commas.

Mode	Description
FIXed	Nothing happens. The output current remains at the immediate value.
STEP	The output goes to the triggered level when a trigger occurs.
LIST	The output follows the list values when a trigger occurs.

Parameter	Typical return
FIX | STEP | LIST	FIX, STEP, or LIST
*RST FIX	
Sets the current mode of channel 3 to Step: CURR:MODE STEP, (@3)	

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

The command sets the level at which overcurrent protection trips, in amperes. This command is only applicable during persona mode.

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

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 in Maximum value>	
Sets the level at which overvoltage protection trips to 20 V: VOLT:PROT 20, (@2)	

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

The command clears an overcurrent protection event.

Parameter	Typical return
(none)	(none)
Clears an overcurrent protection event for channel 1: CURR:PROT:CLE (@1)	

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

Sets the time in milliseconds that the overcurrent protection is temporarily disabled after a current level change. Minimum value is 0 and maximum value is 3600 seconds.

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

Parameter	Typical return
0 – 3600.000 | MIN | MAX |	<delay value>
*RST 0.050	
Sets the protection delay to 0.2 seconds for channel 1: CURR:PROT:DEL 0.2, (@1)	

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

The command specifies the conditions under which the overcurrent protection delay timer starts:

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.

The query returns SCH or CCTR.

Parameter	Typical return
SCH | CCTR	SCH or CCTR
*RST SCH	
Sets CCTRans as the current protection delay mode for channel 1: CURR:PROT:DEL:STAR CCTR, (@1)	

[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 power supply 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) or the over current protection state.

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST OFF	
Enable the current protection state: CURR:PROT:STAT ON, (@1)	

[SOURce:]CURRent:PROTection:TRIPped?

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:]DIGital:INPut:DATA?

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

Parameter	Typical return
(none)	<bit value>
Returns 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
[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 mainframes. Only one pin can be configured as an On control. The pin will function as both an input and an output. The polarity of the pin is fixed and cannot be programmed.
OFFCouple	When configured as an Off control, the pin will synchronize the output Off state between mainframes. Only one pin can be configured as an Off control. The pin will function as both an input and an output. The polarity of the pin is fixed and cannot be programmed.
RELay	The pin is configured as an output relay. See OUTPut:RELay.

Parameter	Typical return
DIO | DINP | TOUT | TINP | FAUL | INH | ONC | OFFC | REL *RST DINP	DIO, DINP, TOUT, TINP, FAUL, INH, ONC, OFFC, or REL
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
Sets pin 1 to negative polarity: DIG:PIN1:POL NEG	

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

The command 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
OFF | 0 | ON | 1	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 and hold **[Local]** for a few seconds to return to the local mode.

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

Parameter	Typical return
ON | OFF | 1 | 0	0 or 1
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. Commas, periods, and semicolons share a display space with the preceding character, and are not considered individual characters.

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]:VIEW METER1 | METER2
DISPlay[:WINDow]:VIEW?

The command selects 1- or 2-channel meter view for E36233A and E36234A models. METER1 displays one large output channel with one small less detailed channel. METER2 displays both output channels.

The query returns METER1 or METER2.

Parameter	Typical return
METER1 | METER2	METER1 or METER2
*RST METER2	
Sets the front panel to display both output channels: `DISP:VIEW METER2`	

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

The query fetches numbers of logged data from the channels. The fetched data is the next data after the last fetched data in the previous fetch.

Parameter	Typical return
0 to 65536	<binary list>
Fetches 10 logged data from channel 1 enabled with both voltage and current data logging: FETC:DLOG? 10, (@1)	

*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	Value	Name	Description
0	1	OPC	Operation Complete. All commands prior to and including an *OPC command have been executed.
1	0	(unused)	Always set to 0.
2	4	QYE	Query Error. The power supply tried to read the output buffer but it was empty. Or, a new command line was received before a previous query had been read. Or, both the input and output buffers are full.
3	8	DDE	Device Error. A self-test or calibration error occurred.
4	16	EXE	Execution Error. An execution error occurred.
5	32	CME	Command Error. A command syntax error occurred.
6	0	(unused)	Always set to 0.
7	128	PON	Power On. Power has been turned off and on 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,E36233A,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; and the third is for the main board 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, OCP state, and OCP delay start

– Output sense

– Output state, selected output and couple trigger channel operation mode (Independent, Series, Parallel or Track)

– Output On/Off sequencing

– Output list settings

– Trigger settings

– Digital I/O output data and bus settings

– Data logger trigger source

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.

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] (@<chanlist>)

The command enables the output trigger system. When an output trigger is initiated, an event on a selected trigger source causes the specified triggering action to occur. If the trigger system is not initiated, all triggers are ignored.

Parameter	Typical return
(@<chanlist>)	(none)
Initiates the trigger system on channel 1: INIT (@1)	

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

The command continuously initiates the output 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 output trigger system must be initiated for each trigger using the INIT 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
Initiates the trigger system on channel 1 continuously: INIT:CONT ON, (@1)	

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

The command starts the internal or external data logging session. The filename in which to save the data should be the full path and filename. For internal data logging session, you will have to specify the exact file name, which is either Internal:/log1.dlog, Internal:/log2.dlog, or Internal:/log3.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"	

NOTE For the E36231A and E36232A single channel models, the CH1 | CH2 parameter is not applicable. The CH1 | CH2 parameter is only applicable for the E36233A and E36234A dual channel models. You can identify each output by the output name (CH1 or CH2). The CH1 | CH2 parameter is optional. If it is not included in the syntax then the executed command will apply to the selected channel.

INSTrument[:SELect] CH1 | CH2
INSTrument[:SELect]?

This command selects the output to be programmed among two outputs by the output identifier. The outputs of the E36233A and E36234A power supplies are considered two logical instruments. The INSTrument command provides a mechanism to identify and select an output. When one output is selected, the other outputs are unavailable for programming until selected. The commands which are affected by the INSTrument command are output setting commands (SOURce), measurement commands (MEASure), and calibration commands (CALibration).

The query returns the currently selected output by the INSTrument [:SELect] or INSTrument:NSELect command.

Parameter	Typical return
CH1 | CH2	CH1 | CH2
Selects channel 1 to be programmed: `INST CH1`	

INSTrument:NSELect 1 | 2
INSTrument:NSELect?

This command selects the output to be programmed among two outputs by a numeric value instead of the output identifier used in the INSTrument [:SELect] command. “1” selects CH1 output and “2” selects CH2 output.

The query returns the currently selected output by the INSTrument:NSELect or INSTrument[:SELect] command. The returned parameter is “1” for CH1 output, “2” for CH2 output.

Parameter	Typical return
1 | 2	1 | 2
Selects channel 1 output to be programmed: `INST:NSEL 1`	

[SOURce:]LIST:COUNt <count> | 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.

NOTE 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.

Parameter	Typical return
1 - 9999 | MIN | MAX | INF *RST 1	<list count>
Sets the list count for channel 1 to 10: LIST:COUN 10, (@1)	

[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 100 steps may be programmed.

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

NOTE 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.

Parameter	Typical return
0 to maximum. The maximum value depends on rating of the power module. *RST 1 step set to the minimum programmable value.	<list value 1>, <list value 2>, <list value 3>...
Programs a current list containing 5 steps for channel 1: LIST:CURR 5, 4, 3, 2, 1, (@1)	

[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 for channel 1: LIST:CURR:POIN? (@1)	

[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 100 steps may be programmed. Dwell time is the time that the output will remain at a specific step. Dwell times can be programmed from 0.01 through 3600 seconds with the resolution of 1 millisecond.

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

NOTE

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.

Parameter	Typical return
0.01 – 3600.000 *RST 0.01	<list value 1>, <list value 2>, <list value 3>...
Programs a dwell list containing 5 steps for channel 1: LIST:DWEL 0.2, 0.8, 1.5, 0.8, 0.2, (@1)	

[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? (@1)	

[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 of channel 1 to ONCE: LIST:STEP ONCE, (@1)	

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

Specifies the value when the list terminates. The state is either 1 (ON) or 0 (OFF). When ON, the output voltage or current 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
OFF | 0 | ON | 1	0 or 1
*RST OFF	
Sets the list step setting of channel 1 to ONCE: LIST:STEP ONCE, (@1)	

[SOURce:]LIST:TOUTput:BOSTep[:DATA] <bool>{, <bool>}, (@<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 100 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.

NOTE 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.

Parameter	Typical return
OFF | 0 | ON | 1 *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 for channel 1: LIST:TOUT:BOST 0, 1, 1, 1, 1, (@1)	

[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.

Parameter	Typical return
(none)	<points>
Returns the number of points in the BOSTep list for channel 1: LIST:TOUT:BOST:POIN? (@1)	

[SOURce:]LIST:TOUTput:EOSTep[:DATA] <bool>{, <bool>}, (@<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 100 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.

NOTE

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.

Parameter	Typical return
OFF | 0 | ON | 1 *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 for channel 1: LIST:TOUT:EOST 0, 1, 1, 1, 1, (@1)	

[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 for channel 1: LIST:TOUT:EOST:POIN? (@1)	

[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 100 steps may be programmed.

NOTE

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.

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 to maximum. The maximum value depends on rating of the power module. *RST 1 step set to the minimum programmable value.	<list value 1>, <list value 2>, <list value 3>...
Programs a voltage list containing 5 steps for channel 1: LIST:VOLT 1, 2, 3, 4, 5, (@1)	

[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 for channel 1: LIST:VOLT:POIN? (@1)	

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`	

NOTE For the E36231A and E36232A single channel models, the CH1 | CH2 parameter is not applicable. The CH1 | CH2 parameter is only applicable for the E36233A and E36234A dual channel models. You can identify each output by the output name (CH1 or CH2). The CH1 | CH2 parameter is optional. If it is not included in the syntax then the executed command will apply to the selected channel.

MEASure[:SCALar]:CURRent[:DC]? [CH1 | CH2,] (@<chanlist>)

The query returns the current measured at the output terminals of the power supply. If no output identifier is specified, the current of the currently selected output is returned.

Parameter	Typical return
(none)	<current>
Returns the current measured for channel 1: `MEAS:CURR? CH1`	

MEASure[:SCALar]:VOLTage[:DC]? [CH1 | CH2,] (@<chanlist>)

The query returns the voltage measured at the output terminals of the power supply. If no output identifier is specified, the voltage of the currently selected output is returned.

Parameter	Typical return
(none)	<voltage>
Returns the voltage measured for channel 1: `MEAS:VOLT? CH1`	

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"	

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

This command enables or disables the dc source 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 | 1 | OFF | 0	1 or 0
*RST 0	
Disables channel 1: OUTP 0, (@1)	

OUTPut[:STATe]COUPle:CHANNel ALL | NONE | <list>
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 | <list>	<coupled channels>
*RST NONE	
Couples channels 1 and 2: OUTP:COUP:CHAN CH1, CH2	

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

The command sets the delay in seconds that the power supply 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]. The command does not affect transitions to off caused by protection functions. 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 – 3600.000 | MIN | MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s for channel 2 before turning the output off: OUTP:DEL:FALL 0.5, (@2)	

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

The command sets the delay in seconds that the power supply 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 – 3600.000 | MIN | MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s for channel 2 before turning the output on:: OUTP:DEL:RISE 0.5, (@2)	

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

This 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 of channel 1 to CURRent: OUTP:PMOD CURR, (@1)	

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

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

The query returns LATC, LIVE, or OFF.

Mode	Description
LATChing	A transition to True on the Inhibit input disables all outputs, 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
LATChing | LIVE | OFF	LATC, LIVE, or OFF
*RST OFF	
Sets the Inhibit input to Live mode: OUTP:INH:MODE LIVE	

OUTPut:PAIR OFF | PARallel | SERies
OUTPut:PAIR?

The command specifies the power supply's operation mode. Coupling and tracking must not be used in when the instrument is operating in parallel or series.

The query returns the power supply's operation mode.

Parameter	Typical return
OFF | PAR | SER *RST OFF	OFF, PAR, or SER
Specifies the power supply's operation mode to parallel: `OUTP:PAIR PAR`	

OUTPut:PON:STATe RST | RCL0 | RCL1 | RCL2 | . . . . | RCL8 | RCL9, (@<chanlist>)
OUTPut:PON:STATe?

The command specifies whether the power supply's power-on state is the *RST state (default) or the state stored in one of ten memory locations.

The query returns either RST or RCL#, where # is a digit from 0 to 9.

Parameter	Typical return
RST | RCL0 | RCL1 | RCL2 | RCL3 | RCL4 | RCL5 | RCL6 | RCL7 | RCL8 | RCL9 *RST RST	RST, RCL0, RCL1, RCL2, RCL3, RCL4, RCL5, RCL6, RCL7, RCL8, or RCL9
Sets channel 1 power-on state stored in memory location 1 at power on: `OUTP:PON:STAT RCL1, (@1)`	

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 latch that disables channel 1 due to an overvoltage or overcurrent condition: `OUT:PRO:CLE (@1)`	

OUTPut:RELay ON | OFF | 1 | 0, (@<chanlist>)
OUTPut:RELay? (@<chanlist>)

When the output of the power supply is turned off, it is implemented by setting the output to 0 volts and 0.02 amps. This gives a zero-output voltage without actually disconnecting the output. To disconnect the output an external relay must be connected between the output and the load. A TTL signal of either low true or high true is provided to control an external relay. This signal can only be controlled with the remote command OUTPut:RELay OFF | ON. The TTL output is available on the Digital IO pin 1 for channel 1 and pin 2 for channel 2. When the OUTPut:RELay state is "ON" on channel 1, the TTL output of pin 1 is high if the Polarity is positive and pin 1 is low if the Polarity is negative. The levels are reversed when the OUTPut:RELay state is "OFF".

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

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST 0	
Turns on the relay state of the power supply: OUTP:REL ON	

OUTPut:TRACk[:STATe] ON | OFF | 1 | 0
OUTPut:TRACk[:STATe]?

The command enables or disables the power supply to operate in the track mode. When the track mode is first enabled, the absolute value of the -25 V supply will be set to the same voltage level as the +25 V supply. Once enabled, any change of the programmed voltage level in either +25 V supply or -25 V supply will be reflected in the other supply. The OUTPut:TRACk OFF command returns the power supply to the non-track mode. The ±25 V supplies must not be coupled to enable track mode. At *RST, the track mode is disabled.

The query returns the track mode state of the power supply. The returned value is 0 (OFF) or 1 (ON).

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST 0	
Enables the power supply to operate in the track mode: OUTP:TRAC ON	

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

The command enables or disables current data logging of the channels stated in (@<chanlist>).

The query returns the status (on or off) of the current data logging enable of the channels stated in (@<chanlist>).

Parameter	Typical return
ON | OFF | 1 | 0	(none)
Enables current data logging of channel 1 and 2: SENS:DLOG:FUNC:CURR 1, (@1,2)	

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

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)
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?

The command enables or disables voltage data logging of the channels stated in (@<chanlist>).

The query returns the status (on or off) of the voltage data logging of the channels stated in (@<chanlist>).

Parameter	Typical return
ON | OFF | 1 | 0	(none)
Enables voltage data logging of channel 1 to 2: SENS:DLOG:FUNC:VOLT 1, (@1:2)	

SENSe:DLOG:PER <time>
SENSe:DLOG:PER?

The command sets the sample period consisting of the entered value in seconds. For the sample period, the minimum is 200 ms, and it has to be in integral of 200 ms, while the maximum is 60 s. The entered value will be rounded to the nearest 200 ms integral.

The query returns the sample period in seconds.

Parameter	Typical return
0.2 to 60	(none)
Sets the sample period to 400 ms: SENS:DLOG:PER 0.4	

SENSe:DLOG:TIME <time>
SENSe:DLOG:TIME?

The command sets the sample duration, the entered value is in seconds. For sample duration, the maximum is about 21,845 hours and 20 minutes (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 to 78642000	(none)
Sets the sample duration to 2 minutes: SENS:DLOG:TIME 120	

SENSe:DLOG:TINTerval <time>
SENSe:DLOG:TINTerval?

The command sets the sample period consisting of the entered value in seconds. For the sample period, the minimum is 200 ms, and it has to be in integral of 200 ms, while the maximum is 60 s. The entered value will be rounded to the nearest 200 ms integral.

The query returns the sample period in seconds.

Parameter	Typical return
0.2 to 60	(none)
Sets the sample period to 400 ms: SENS:DLOG:TINT 0.4	

STATus:OPERation:CONDition?

Queries the condition register for the Standard Operation Register group. The query reads the condition 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>
Reads the condition register: STAT:OPER:COND?	

STATus:OPERation[:EVENt]?

Returns the sum of the bits in the event register for the Standard Operation Register group. The query returns a decimal value which corresponds to the binary-weighted sum of all bits in the register.

Parameter	Typical return
(none)	<bit value>
Reads the event register: STAT:OPER?	

STATus:OPERation:ENABle <enable value>
STATus:OPERation:ENABle?

The command enables bits in the enable register for the Standard Operation Register group. The selected bits are then reported to the Status Byte.

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	

STATus:PRESet

The command sets the questionable enable registers to their power-on values.

Parameter	Typical return
(none)	(none)
Sets the questionable enable registers to their power-on values: STAT:PRES	

STATus:QUEStionable:CONDition?

Queries the condition register for the Questionable Data Register group. The query reads the condition 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>
Reads the condition register: STAT:QUES:COND?	

STATus:QUEStionable:[:EVENt]?

Returns the sum of the bits in the event register for the Questionable Data Register group. The query returns a decimal value which corresponds to the binary-weighted sum of all bits in the register.

Parameter	Typical return
(none)	<bit value>
Reads the event register: STAT:QUES?	

STATus:QUEStionable:ENABle <enable value>
STATus:QUEStionable:ENABle?

Enables bits in the enable register for the Questionable Data Register group. The selected bits are then reported to the Status Byte.

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 enable register: STAT:QUES:ENAB 16	

STATus:QUEStionable:INSTrument[:EVENt]?

Returns the sum of the bits in the event register for the Questionable Instrument Register group. The query returns a decimal value which corresponds to the binary-weighted sum of all bits in the register.

Parameter	Typical return
(none)	(none)
Reads the event register: STAT:QUES:INST?	

STATus:QUEStionable:INSTrument:ENABle <enable value>
STATus:QUEStionable:INSTrument:ENABle?

Enables bits in the enable register for the Questionable Instrument Register group. The selected bits are then reported to the Status Byte. This register is a mask for enabling specific bits from the Questionable Instrument event register to set the Instrument Summary bit (ISUM, bit 13) of the Questionable Status register. The "ISUM" bit of the Questionable Status register is the logical OR of all the Questionable Instrument event register bits that are enabled by the Questionable Instrument 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 enable register: `STAT:QUES:INST:ENAB 16`	

STATus:QUEStionable:INSTrument:ISUMmary<n>[:EVENt]?

The query returns the value of the Questionable Instrument Isummary event register for a specific output of the power supply. The particular output must be specified by a numeric value (<n>=1 or 2). See Programming Ranges and Output Identifiers for the output number. The event register is a read-only register which holds (latches) all events. Reading the Questionable Instrument Isummary event register clears it.

Parameter	Typical return
(none)	(none)
Returns the value of the Questionable Instrument Isummary event register for output 1: `STAT:QUES:INST:ISUM1?`	

STATus:QUEStionable:INSTrument:ISUMmary<n>:CONDition?

The query returns the CV or CC condition of the specified output (<n>=1 or 2), as shown in the table below.

Returned value	Description
0	The output is off or unregulated.
1	The output is in CC (constant current) operating mode.
2	The output is in CV (constant voltage) operating mode.
3	The output has a hardware failure.

Parameter	Typical return
(none)	0, 1, 2, or 3
Returns the CV or CC condition of the instrument for output 1: `STAT:QUES:INST:ISUM1:COND?`	

STATus:QUEStionable:INSTrument:ISUMmary<n>:ENABle <enable value>
STATus:QUEStionable:INSTrument:ISUMmary<n>:ENABle?

The command sets the value of the Questionable Instrument Isummary enable register for a specific output of the E36233A and E36234A dual output power supplies. The particular output must be specified by a numeric value (<n>=1 or 2). See Programming Ranges and Output Identifiers for the output number. This register is a mask for enabling specific bits from the Questionable Instrument Isummary event register to set the Instrument Summary bit (bit 1 and 2) of the Questionable Instrument register. These bits 1 and 2 are the logical OR of all the Questionable Instrument Isummary event register bits that are enabled by the Questionable Instrument Isummary enable register.

The query returns the value of the Questionable Instrument Isummary enable register (<n>=1 or 2).

Parameter	Typical return
(none)	1 or 2
Returns the value of the Questionable Instrument Isummary enable register for output 1: `STAT:QUES:INST:ISUM1:ENAB?`	

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
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
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 power supply's real-time clock. The range of values for the year is from 2000 – 2099.

The query returns the date in the form +2017,+7,+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's 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:PERSona:MANufacturer "<string>" | DEFault
SYSTem:PERSona:MANufacturer?
SYSTem:PERSona:MANufacturer? DEFault

The command is only applicable for E36231A and E36232A single channel models. The command changes the manufacturer identity. This command is used to change the manufacturer's ID string for backward compatibility. This affects the following identity items:

*IDN? contains "Agilent" instead of "Keysight".

The DEFault command returns the instrument to its original model identification state.

The query returns the manufacturer's ID.

Parameter	Typical return
"<string>" | DEFault	Keysight Technologies
Sets the manufacturer identity to HEWLETT-PACKARD: SYST:PERS:MAN “HEWLETT-PACKARD”	

SYSTem:PERSona:MODel E3623XA | E363XA | DEFault
SYSTem:PERSona:MODel?
SYSTem:PERSona:MODel? DEFault

The command is only applicable for E36231A and E36232A single channel models. The command changes the model number. This command is used to change the mainframe model ID string for backward compatibility.

The DEFault command returns the original model identification.

The query returns the model.

Parameter	Typical return
E3623XA | E363XA | DEFault	E3623XA or E363XA
Sets the model to E363XA: SYST:PERS:MOD E363XA	

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:DLOG[:IMMediate]

The command sends an immediate trigger signal to the data logger. This will trigger the data log regardless of the selected trigger source. You must initiate (run) the data logger before you trigger it.

Parameter	Typical return
(none)	(none)
Triggers the data logger immediately: TRIG:DLOG	

TRIGger:DLOG:SOURce BUS | EXTernal | IMMediate
TRIGger:DLOG:SOURce?

The command selects the trigger source for the data logger.

The query returns BUS, EXT, or IMM.

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.

Parameter	Typical return
BUS | EXT | IMM	BUS, EXT, or IMM
Selects BUS as the data log trigger source: TRIG:DLOG:SOUR BUS	

TRIGger[:SEQuence]:DELay <seconds> | MINimum | MAXimum, (@<chanlist>)
TRIGger[: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. Select from 0 to 3600 seconds. Minimum value is 0 and maximum value is 3600 seconds.

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

Parameter	Typical return
0 – 3600.000 | MIN | MAX |	<delay value>
*RST 0	
Sets the trigger time delay to 5 seconds on channels 1 and 2: TRIG:DEL 5, (@1,2))	

TRIGger[:SEQuence]:SOURce BUS | EXTernal | IMMediate | PIN1 | PIN2 | PIN3, (@<chanlist>)
TRIGger[: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.

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 *RST BUS	BUS, EXT, IMM, PIN1, PIN2, or PIN3
Selects pin 1 as the output trigger source for channel 2: TRIG:SOUR PIN1, (@2)	

Program example

The program below sets CH1 to an output of 3 V/1 A immediately:

```
INST CH1          Selects CH1 as the output
VOLT:MODE STEP    Set the voltage mode to Step
VOLT:TRIG 3       Set triggered voltage to 3 V
CURR:MODE STEP    Set the current mode to Step
CURR:TRIG 1       Set triggered current to 1 A
TRIG:SOUR IMM     Select the immediate trigger source
INIT              Initiate the trigger system
```

Triggering Commands

The instrument's triggering system allows you to change voltage and current output when a trigger is received. The typical process is:

1. Select one or more outputs with the INSTrument:SELect command.
2. Configure the triggered output levels by using CURRent:TRIGgered and VOLTage:TRIGgered.
3. Configure the voltage and current mode to Step by using VOLTage:MODE and CURRent:MODE.
4. Specify the trigger source, either BUS or IMMediate.
5. 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.
6. 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.

[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 output voltage level of the output in volts. The immediate level is the voltage limit value of the selected output with the INSTrument command.

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 output voltage level of the output in volts. MAX and MIN return the highest and lowest programmable voltage levels respectively for the selected range.

Model	Range of values
E36232A/E36234A (60 V) models	MAX = 61.8 V MIN = 0 V DEF = 0 V
E36231A/E36233A (30 V) models	MAX = 30.9 V MIN = 0 V DEF = 0 V

Parameter	Typical return
0 - maximum | MIN | MAX | DEF | UP | DOWN (The maximum value is dependent on the voltage rating of the power module)	<voltage level>
*RST MIN	
Sets the output voltage level to 20 V: VOLT 20, (@1)	

[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.

Model	Range of values
E36232A/E36234A (60 V) models	MAX = 61.8 V MIN = 0 V DEF = 0 V
E36231A/E36233A (30 V) models	MAX = 30.9 V MIN = 0 V DEF = 0 V

Parameter	Typical return
0 - maximum | DEF (The maximum value is dependent on the voltage rating of the power module) *RST <DEF in Default value>	<voltage level>
Sets the output voltage step size to 3 A: VOLT:STEP 3, (@1)	

[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 specified output channel in volts. The triggered level is a stored value that is transferred to the output when an output step is triggered.

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

Parameter	Typical return
0 - maximum | MIN | MAX (The maximum value is dependent on the voltage rating of the power module) *RST MIN	<voltage level>
Sets the triggered voltage level to 5 V: VOLT:TRIG 5, (@1)	

[SOURce:]VOLTage:MODE FIXed | STEP | LIST, (@<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 list values when a trigger occurs.

Parameter	Typical return
FIX | STEP | LIST	FIX, STEP, or LIST
*RST FIX	
Sets the voltage mode of channel 2 to Step: VOLT:MODE STEP,(@2)	

[SOURce:]VOLTage:PROTection[:LEVel][:AMPLitude] <voltage> | MINimum | MAXimum, (@<chanlist>)
[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)	<voltage level>
*RST <MAX in Maximum value>	
Sets the level at which overvoltage protection trips to 20 V: VOLT:PROT 20, (@2)	

[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, (@1)	

[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:RANGe P8V | P25V | P20V | P50V | LOW | HIGH, (@<chanlist>)
[SOURce:]VOLTage:RANGe? (@<chanlist>)

The command sets the output voltage range on models that have multiple ranges. This command is only applicable during persona mode. The power supply only support 1 range for voltage, accepting and returning a range that corresponds to the single range.

The query returns the presently selected range.

Parameter	Typical return
P8V | P25V | P20V | P50V | LOW | HIGH	<voltage range>
Queries the output voltage range for channel 1: SOUR:VOLT:RANG? (@1)	

[SOURce:]VOLTage:SLEW:RISing[:IMMediate] <slew> | MINimum | MAXimum | INFinity, (@<chanlist>)
[SOURce:]VOLTage:SLEW:RISing[:IMMediate]? MINimum | MAXimum (@<chanlist>)

The command sets the rising voltage slew rate in volts per second. The slew rate setting affects the rising programmed voltage changes, including those due to the output state turning on. The slew rate can be set from 0.003 (for E36232A and E36234A models) or 0.002 (E36231A and E36233A models) up to any value, however, if the value is more than the max slew rate, the DUT will slew based on the max slew rate. For very large values, the slew rate will be limited by the analog performance of the output circuit. The keywords MAXimum or INFinity sets the slew rate to the maximum value. The query returns the value that was sent, unless the value was more than the maximum slew rate, in which case the maximum value is returned.

The query returns the programmed rising slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of +9.90000030E+37 is returned, it means that the maximum or fastest slew rate has been set.

Parameter	Typical return
0 - +9.90000030E+37 | MIN | MAX | INF	<slew rate>
*RST +9.90000030E+37	
Sets the rising voltage slew rate to 5 volts per second on channel 1: VOLT:SLEW:RIS 5, (@1)	

[SOURce:]VOLTage:SLEW:RISing:MAXimum <bool>, (@<chanlist>)
[SOURce:]VOLTage:SLEW:RISing:MAXimum? (@<chanlist>)

The command sets the voltage rising slew rate maximum override. When enabled (ON), the power module's rising slew rate is set to its maximum value. When disabled (OFF), the power module's rising slew rate is set to the immediate value set by the [SOURce:]VOLTage:SLEW:RISing command. The [SOURce:]VOLTage:SLEW:RISing:MAX command does not affect the [SOURce:]VOLTage:SLEW:RISing setting.

The query returns 0 (OFF) if the rising slew rate override is disabled, and 1 (ON) if the rising slew rate override is enabled.

NOTE **The [SOURce:]VOLTage:SLEW:RISing:MAX command is coupled to the [SOURce:]VOLTage:SLEW:RISing command. If [SOURce:]VOLTage:SLEW:RISing sets the rate to MAX or INFinity, [SOURce:]VOLTage:SLEW:RISing:MAXimum is enabled. If the slew rate is set to any other value, [SOURce:]VOLTage:SLEW:RISing:MAXimum is disabled.**

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST ON	
Enables the rising slew rate maximum override on channel 1. VOLT:SLEW:RIS:MAX ON, (@1)	

[SOURce:]VOLTage:SLEW:FALLing[:IMMediate] <slew> | MINimum | MAXimum | INFinity, (@<chanlist>)
[SOURce:]VOLTage:SLEW:FALLing[:IMMediate]? MINimum | MAXimum (@<chanlist>)

The command sets the falling voltage slew rate in volts per second. The slew rate setting affects the falling programmed voltage changes, including those due to the output state turning on. The slew rate can be set from 0.003 (for E36232A/E26234A models) or 0.002 (for E36231A/E36233A models) up to any value, however, if the value is more than the max slew rate, the DUT will slew based on the max slew rate. For very large values, the slew rate will be limited by the analog performance of the output circuit. The keywords MAXimum or INFinity sets the slew rate to the maximum value. The query returns the value that was sent, unless the value was more than the maximum slew rate, in which case the maximum value is returned.

The query returns the programmed falling slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of +9.90000030E+37 is returned, it means that the maximum or fastest slew rate has been set.

Parameter	Typical return
0 - +9.90000030E+37 | MIN | MAX | INF	<slew rate>
*RST +9.90000030E+37	
Sets the falling voltage slew rate to 5 volts per second on channel 1: VOLT:SLEW:FALL 5, (@1)	

[SOURce:]VOLTage:SLEW:FALLing:MAXimum <bool>, (@<chanlist>)
[SOURce:]VOLTage:SLEW:FALLing:MAXimum? (@<chanlist>)

The command sets the voltage falling slew rate maximum override. When enabled (ON), the power module's falling slew rate is set to its maximum value. When disabled (OFF), the power module's falling slew rate is set to the immediate value set by the [SOURce:]VOLTage:SLEW:FALLing command. The [SOURce:]VOLTage:SLEW:FALLing:MAX command does not affect the [SOURce:]VOLTage:SLEW:FALLing setting.

The query returns 0 (OFF) if the falling slew rate override is disabled, and 1 (ON) if the falling slew rate override is enabled.

NOTE The [SOURce:]VOLTage:SLEW:FALLing:MAX command is coupled to the [SOURce:]VOLTage:SLEW:FALLing command. If [SOURce:]VOLTage:SLEW:FALLing sets the rate to MAX or INFinity, [SOURce:]VOLTage:SLEW:FALLing:MAXimum is enabled. If the slew rate is set to any other value, [SOURce:]VOLTage:SLEW:FALLing:MAXimum is disabled.

Parameter	Typical return
ON | 1 | OFF | 0	1 or 0
*RST ON	
Enables the falling slew rate maximum override on channel 1. VOLT:SLEW:FALL:MAX ON, (@1)	

[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, (@1)	

This information is subject to change without notice.

© Keysight Technologies 2019, 2021
Edition 1, September 2021
Printed in Malaysia

E36200-90008
www.keysight.com