

Programming  
Guide

# EL30000 Series DC Electronic Loads



<b>Notices</b> .....	<b>11</b>
<b>Copyright Notice</b> .....	<b>11</b>
<b>Manual Part Number</b> .....	<b>11</b>
<b>Edition</b> .....	<b>11</b>
<b>Published by</b> .....	<b>11</b>
<b>Warranty</b> .....	<b>11</b>
<b>Technology Licenses</b> .....	<b>11</b>
<b>U.S. Government Rights</b> .....	<b>12</b>
<b>Third Party Licenses</b> .....	<b>12</b>
<b>Waste Electrical and Electronic Equipment (WEEE)</b> .....	<b>12</b>
<b>Declarations of Conformity</b> .....	<b>12</b>
<b>Safety Information</b> .....	<b>13</b>
<b>1 Remote Operation</b> .....	<b>14</b>
<b>Introduction to the SCPI Language</b> .....	<b>15</b>
Command Format Used in this Manual .....	16
Command Separators .....	16
Querying Parameter Settings .....	17
SCPI Command Terminators .....	17
IEEE-488.2 Common Commands .....	17
SCPI Parameter Types .....	18
Halting an Input in Progress .....	19
<b>Programming Ranges</b> .....	<b>20</b>
<b>SCPI Error Messages</b> .....	<b>22</b>
Execution error codes .....	22
Calibration error codes .....	29
Self-test error codes .....	30
<b>Reset and Non-volatile Settings</b> .....	<b>31</b>
Reset (*RST) settings .....	31
Non-volatile settings .....	34
Non-volatile LAN settings .....	34
<b>SCPI Status Registers</b> .....	<b>35</b>
Status Registers .....	35
Operation Status Group .....	35
Questionable Status Group .....	36
Standard Event Status Group .....	37
Status Byte Register .....	37
Error and Output Queues .....	38
Status Diagram .....	39
<b>2 SCPI Programming</b> .....	<b>40</b>
<b>ABORt Subsystem</b> .....	<b>41</b>
ABORt:ACQuire [(@<chanlist>)] .....	41
ABORt:DLOG .....	41
ABORt:TRANsient [(@<chanlist>)] .....	41
<b>CALibration Subsystem</b> .....	<b>42</b>
CALibration:ASAVE ON   OFF   1   0 .....	42
CALibration:ASAVE? .....	42
CALibration:COUNt? .....	42
CALibration:CURRent[:DATA]:[HIGH] <current>[, (@<chanlist>)] .....	42
CALibration:CURRent[:DATA]:MID <current>[, (@<chanlist>)] .....	42
CALibration:CURRent[:DATA]:LOW <current>[, (@<chanlist>)] .....	42

CALibration:CURRent:LEVel[:HIGH] POINT1   POINT2   POINT3   POINT4   POINT5[, (@<chanlist>)]	43
CALibration:CURRent:LEVel:MID MINimum   MAXimum[, (@<chanlist>)]	43
CALibration:CURRent:LEVel:LOW MINimum   MAXimum[, (@<chanlist>)]	43
CALibration:CURLimit[:DATA] <current>[, (@<chanlist>)]	43
CALibration:CURLimit:LEVel MINimum   MAXimum[, (@<chanlist>)]	43
CALibration:DATE "<string>"	44
CALibration:DATE?	44
CALibration:SAVE	44
CALibration:SECure:CODE <new passcode>	44
CALibration:SECure:STATe ON   OFF   1   0, <passcode>	44
CALibration:SECure:STATe?	44
CALibration:STRing "<string>"	45
CALibration:STRing?	45
CALibration:VOLTage[:DATA][:HIGH] <numeric value>[, (@<chanlist>)]	45
CALibration:VOLTage[:DATA]:LOW <numeric value>[, (@<chanlist>)]	45
CALibration:VOLTage:LEVel[:HIGH] MINimum   MAXimum[, (@<chanlist>)]	45
CALibration:VOLTage:LEVel:LOW MINimum   MAXimum[, (@<chanlist>)]	45
<b>CHANnel   INSTRument Subsystem</b>	<b>46</b>
CHANnel[:SElect] [CH1   CH2]	46
CHANnel[:SElect]?	46
INSTRument[:SElect] [CH1   CH2]	46
INSTRument[:SElect]?	46
CHANnel:NSElect [1   2]	46
CHANnel:NSElect?	46
INSTRument:NSElect [1   2]	46
INSTRument:NSElect?	46
CHANnel[:LOAD] [1   2]	47
CHANnel[:LOAD]? MAX	47
INSTRument[:LOAD] [1   2]	47
INSTRument[:LOAD]? MAX	47
<b>CURRent Subsystem</b>	<b>48</b>
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <current>   MINimum   MAXimum   DEFault[, (@<chanlist>)]	48
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MINimum   MAXimum   DEFault,] [(@<chanlist>)]	48
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <current>   MINimum   MAXimum[, (@<chanlist>)]	48
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MINimum   MAXimum,] [(@<chanlist>)]	48
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <current>   MINimum   MAXimum[, (@<chanlist>)]	49
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum   MAXimum,] [(@<chanlist>)]	49
[SOURce:]CURRent:MODE FIXed   STEP   LIST[, (@<chanlist>)]	49
[SOURce:]CURRent:MODE? [(@<chanlist>)]	49
[SOURce:]CURRent:PROTEction:DELay[:TIME] <time>   MINimum   MAXimum[, (@<chanlist>)]	50
[SOURce:]CURRent:PROTEction:DELay[:TIME]? [MINimum   MAXimum,] [(@<chanlist>)]	50
[SOURce:]CURRent:PROTEction:DELay:STARt SCHange   CCTRans[, (@<chanlist>)]	50
[SOURce:]CURRent:PROTEction:DELay:STARt? [(@<chanlist>)]	50
[SOURce:]CURRent:PROTEction:STATe ON   OFF   1   0[, (@<chanlist>)]	51
[SOURce:]CURRent:PROTEction:STATe? [(@<chanlist>)]	51
[SOURce:]CURRent:PROTEction:TRIPped?	51
[SOURce:]CURRent:RANGe <current>   MINimum   MAXimum[, (@<chanlist>)]	52
[SOURce:]CURRent:RANGe? [MINimum   MAXimum,] [(@<chanlist>)]	52
[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <slew rate>   MINimum   MAXimum   INFINity[, (@<chanlist>)]	53

[SOURce:]CURRent:SLEW[:POSitive][:IMMEDIATE]? [MINimum   MAXimum,] [ (@<chanlist>)]	53
[SOURce:]CURRent:SLEW:NEGAtive[:IMMEDIATE] <slew rate>   MINimum   MAXimum   INFinity[, (@<chanlist>)]	53
[SOURce:]CURRent:SLEW:NEGAtive[:IMMEDIATE]? [MINimum   MAXimum,] [ (@<chanlist>)]	53
[SOURce:]CURRent:SLEW:COUPle ON   OFF   1   0[, (@<chanlist>)]	53
[SOURce:]CURRent:SLEW:COUPle? [(@<chanlist>)]	53
[SOURce:]CURRent:SLEW[:POSitive]:MAXimum ON   OFF   1   0[, (@<chanlist>)]	54
[SOURce:]CURRent:SLEW[:POSitive]:MAXimum? (@<chanlist>)	54
[SOURce:]CURRent:SLEW:NEGAtive:MAXimum ON   OFF   1   0[, (@<chanlist>)]	54
[SOURce:]CURRent:SLEW:NEGAtive:MAXimum? [(@<chanlist>)]	54
[SOURce:]CURRent:TLEVel <transient level>   MINimum   MAXimum[, (@<chanlist>)]	54
[SOURce:]CURRent:TLEVel? [MINimum   MAXimum,] [ (@<chanlist>)]	54
<b>DIGital Subsystem</b>	<b>55</b>
[SOURce:]DIGital:INPut:DATA?	55
[SOURce:]DIGital:OUTPut:DATA <value>	55
[SOURce:]DIGital:OUTPut:DATA?	55
[SOURce:]DIGital:PIN<1-3>:FUNctIon DIO   DINPut   TOUTput   TINPut   FAULt   INHibit   ONCouple   OFFCouple	56
[SOURce:]DIGital:PIN<1-3>:FUNctIon?	56
[SOURce:]DIGital:PIN<1-3>:POLarity POSitive   NEGAtive	57
[SOURce:]DIGital:PIN<1-3>:POLarity?	57
[SOURce:]DIGital:TOUTput:BUS[:ENABle] ON   OFF   1   0	57
[SOURce:]DIGital:TOUTput:BUS[:ENABle]?	57
<b>DISPlay Subsystem</b>	<b>58</b>
DISPlay[:WINDow][:STATe] ON   OFF   1   0	58
DISPlay[:WINDow][:STATe]?	58
DISPlay[:WINDow]:TEXT[:DATA] "<string>"	58
DISPlay[:WINDow]:TEXT[:DATA]?	58
DISPlay[:WINDow]:TEXT:CLEar	58
DISPlay[:WINDow]:VIEW METER1   METER2	59
DISPlay[:WINDow]:VIEW?	59
<b>FETCh Subsystem</b>	<b>60</b>
FETCh[:SCALar]:CURRent[:DC]? [CH1   CH2] [(@<chanlist>)]	60
FETCh[:SCALar]:VOLTage[:DC]? [CH1   CH2] [(@<chanlist>)]	60
FETCh[:SCALar]:POWer[:DC]? [CH1   CH2] [(@<chanlist>)]	60
FETCh[:SCALar]:CURRent:ACDC? [(@<chanlist>)]	60
FETCh[:SCALar]:VOLTage:ACDC? [(@<chanlist>)]	60
FETCh[:SCALar]:CURRent:MAXimum? [(@<chanlist>)]	61
FETCh[:SCALar]:VOLTage:MAXimum? [(@<chanlist>)]	61
FETCh[:SCALar]:POWer:MAXimum? [(@<chanlist>)]	61
FETCh[:SCALar]:CURRent:MINimum? [(@<chanlist>)]	61
FETCh[:SCALar]:VOLTage:MINimum? [(@<chanlist>)]	61
FETCh[:SCALar]:POWer:MINimum? [(@<chanlist>)]	61
FETCh:ARRay:CURRent[:DC]? [(@<chanlist>)]	62
FETCh:ARRay:VOLTage[:DC]? [(@<chanlist>)]	62
FETCh:ARRay:POWer[:DC]? [(@<chanlist>)]	62
FETCh[:SCALar]:DLOG? <number>[, (@<chanlist>)]	62
<b>FUNctIon   MODe Subsystem</b>	<b>63</b>
[SOURce:]FUNctIon CURRent   VOLTage   POWer   RESistance[, (@<chanlist>)]	63
[SOURce:]FUNctIon? [(@<chanlist>)]	63
[SOURce:]MODe CURRent   VOLTage   POWer   RESistance[, (@<chanlist>)]	63
[SOURce:]MODe? [(@<chanlist>)]	63

<b>IEEE-488.2 Common Commands</b> .....	<b>64</b>
*CLS .....	64
*ESE <enable value> .....	64
*ESE? .....	64
*ESR? .....	65
*IDN? .....	65
*OPC .....	65
*OPC? .....	65
*PSC 0   1 .....	66
*PSC? .....	66
*RCL 0   1   2   ...   8   9 .....	66
*RST .....	66
*SAV 0   1   2   ...   8   9 .....	67
*SRE <enable value> .....	67
*SRE? .....	67
*STB? .....	67
*TRG .....	68
*TST? .....	68
*WAI .....	68
<b>INITiate Subsystem</b> .....	<b>69</b>
INITiate[:IMMEDIATE]:ACQUIRE [(@<chanlist>)] .....	69
INITiate[:IMMEDIATE]:DLOG <"filename"> .....	69
INITiate[:IMMEDIATE]:TRANSIENT [(@<chanlist>)] .....	69
INITiate:CONTinuous:TRANSient ON   OFF   1   0[, (@<chanlist>)] .....	70
INITiate:CONTinuous:TRANSient ? [(@<chanlist>)] .....	70
<b>INPut   OUTPut Subsystem</b> .....	<b>71</b>
INPut[:STATe] ON   OFF   1   0[, (@<chanlist>)] .....	71
INPut[:STATe]? [(@<chanlist>)] .....	71
OUTPut[:STATe] ON   OFF   1   0[, (@<chanlist>)] .....	71
OUTPut[:STATe]? [(@<chanlist>)] .....	71
INPut[:STATe]:COUple:CHANNeL ALL   NONE   <list> .....	71
INPut[:STATe]:COUple:CHANNeL? .....	71
OUTPut[:STATe]:COUple:CHANNeL ALL   NONE   <list> .....	71
OUTPut[:STATe]:COUple:CHANNeL? .....	71
INPut[:STATe]:DElay:FALL <delay>   MINimum   MAXimum[, (@<chanlist>)] .....	72
INPut[:STATe]:DElay:FALL? [MINimum MAXimum,] [(@<chanlist>)] .....	72
OUTPut[:STATe]:DElay:FALL <delay>   MINimum   MAXimum[, (@<chanlist>)] .....	72
OUTPut[:STATe]:DElay:FALL? [MINimum MAXimum,] [(@<chanlist>)] .....	72
INPut[:STATe]:DElay:RISE <delay>   MINimum   MAXimum[, (@<chanlist>)] .....	73
INPut[:STATe]:DElay:RISE? [MINimum MAXimum,] [(@<chanlist>)] .....	73
OUTPut[:STATe]:DElay:RISE <delay>   MINimum   MAXimum[, (@<chanlist>)] .....	73
OUTPut[:STATe]:DElay:RISE? [MINimum MAXimum,] [(@<chanlist>)] .....	73
INPut:INHibit:MODE LATChing   LIVE   OFF .....	74
INPut:INHibit:MODE? .....	74
OUTPut:INHibit:MODE LATChing   LIVE   OFF .....	74
OUTPut:INHibit:MODE? .....	74
INPut:PAIR OFF   PARAllel .....	74
INPut:PAIR? .....	74
OUTPut:PAIR OFF   PARAllel .....	74
OUTPut:PAIR? .....	74
INPut:PROTection:CLEar [(@<chanlist>)] .....	75
OUTPut:PROTection:CLEar [(@<chanlist>)] .....	75

INPut:PON:STATe RST   RCL0   RCL1   RCL2   ....   RCL8   RCL9	75
INPut:PON:STATe?	75
OUTPut:PON:STATe RST   RCL0   RCL1   RCL2   ....   RCL8   RCL9	75
OUTPut:PON:STATe?	75
INPut:SHORT[:STATe] ON   OFF   1   0	76
INPut:SHORT[:STATe]?	76
OUTPut:SHORT[:STATe] ON   OFF   1   0	76
OUTPut:SHORT[:STATe]?	76
<b>LIST Subsystem</b>	<b>77</b>
[SOURce:]LIST:COUNt <count>   MINimum   MAXimum   INFinity[, (@<chanlist>)]	77
[SOURce:]LIST:COUNt? [MINimum   MAXimum   INFinity,] [(@<chanlist>)]	77
[SOURce:]LIST:CURRent[:LEVel] <value>{,<value >}, (@<chanlist>)]	77
[SOURce:]LIST:CURRent[:LEVel] ]? [(@<chanlist>)]	77
[SOURce:]LIST:CURRent:POINts? [(@<chanlist>)]	78
[SOURce:]LIST:DWELL <value>{,<value >}, (@<chanlist>)]	78
[SOURce:]LIST:DWELL? [(@<chanlist>)]	78
[SOURce:]LIST:DWELL:POINts? [(@<chanlist>)]	79
[SOURce:]LIST:POWer[:LEVel] <value>{,<value >}, (@<chanlist>)]	79
[SOURce:]LIST:POWer[:LEVel] ]? [(@<chanlist>)]	79
[SOURce:]LIST:POWer:POINts? [(@<chanlist>)]	79
[SOURce:]LIST:RESistance[:LEVel] <value>{,<value >}, (@<chanlist>)]	80
[SOURce:]LIST:RESistance[:LEVel] ]? [(@<chanlist>)]	80
[SOURce:]LIST:RESistance:POINts? [(@<chanlist>)]	80
[SOURce:]LIST:STEP AUTO   ONCE[, (@<chanlist>)]	81
[SOURce:]LIST:STEP? [(@<chanlist>)]	81
[SOURce:]LIST:TERMinate:LAST ON   OFF   1   0[, (@<chanlist>)]	81
[SOURce:]LIST:TERMinate:LAST? [(@<chanlist>)]	81
[SOURce:]LIST:TOUTput:BOSTep[:DATA] ON   OFF   1   0{,ON   OFF   1   0}, (@<chanlist>)]	82
SOURce:]LIST:TOUTput:BOSTep[:DATA]? [(@<chanlist>)]	82
[SOURce:]LIST:TOUTput:BOSTep:POINts? [(@<chanlist>)]	82
[SOURce:]LIST:TOUTput:EOSTep[:DATA] ON   OFF   1   0{,ON   OFF   1   0}, (@<chanlist>)]	83
[SOURce:]LIST:TOUTput:EOSTep[:DATA]? [(@<chanlist>)]	83
[SOURce:]LIST:TOUTput:EOSTep:POINts? [(@<chanlist>)]	83
[SOURce:]LIST:VOLTage[:LEVel] <value> {,<value >}, (@<chanlist>)]	84
[SOURce:]LIST:VOLTage[:LEVel]? [(@<chanlist>)]	84
[SOURce:]LIST:VOLTage:POINts? [(@<chanlist>)]	84
<b>LXI Subsystem</b>	<b>85</b>
LXI:IDENTify[:STATe] ON   OFF   1   0	85
LXI:IDENTify[:STATe?] ON   OFF   1   0	85
LXI:MDNS[:STATe] ON   OFF   1   0	85
LXI:MDNS[:STATe?] ON   OFF   1   0	85
<b>MEASure Subsystem</b>	<b>86</b>
MEASure[:SCALar]:CURRent[:DC]? [ CH1   CH2] [(@<chanlist>)]	86
MEASure[:SCALar]:VOLTage[:DC]? [ CH1   CH2] [(@<chanlist>)]	86
MEASure[:SCALar]:POWer[:DC]? [ CH1   CH2] [(@<chanlist>)]	86
MEASure[:SCALar]:CURRent:ACDC? [(@<chanlist>)]	86
MEASure[:SCALar]:VOLTage:ACDC? [(@<chanlist>)]	86
MEASure[:SCALar]:CURRent:MAXimum? [(@<chanlist>)]	87
MEASure[:SCALar]:VOLTage:MAXimum? [(@<chanlist>)]	87
MEASure[:SCALar]:POWer:MAXimum? [(@<chanlist>)]	87
MEASure[:SCALar]:CURRent:MINimum? [(@<chanlist>)]	87

MEASure[:SCALar]:VOLTage:MINimum? [(@<chanlist>)]	87
MEASure[:SCALar]:POWer:MINimum? [(@<chanlist>)]	87
MEASure:ARRAy:CURRent[:DC]? [(@<chanlist>)]	87
MEASure:ARRAy:VOLTage[:DC]? [(@<chanlist>)]	87
MEASure:ARRAy:POWer[:DC]? [(@<chanlist>)]	87
<b>MMEMory Subsystem</b>	<b>88</b>
MMEMory:EXPort:DLOG <"filename">	88
<b>POWer Subsystem</b>	<b>89</b>
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <power>   MINimum   MAXimum   DEFault[, (@<chanlist>)]	89
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]? [MINimum   MAXimum   DEFault,] [(@<chanlist>)]	89
[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude] <power>   MINimum   MAXimum[, (@<chanlist>)]	89
[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude]? [MINimum   MAXimum,] [(@<chanlist>)]	89
[SOURce:]POWer:MODE FIXed   STEP   LIST[, (@<chanlist>)]	90
[SOURce:]POWer:MODE? [(@<chanlist>)]	90
[SOURce:]POWer:PROTection:DELAy[:TIME] <time>   MINimum   MAXimum[, (@<chanlist>)]	90
[SOURce:]POWer:PROTection:DELAy[:TIME]? [MINimum   MAXimum,] [(@<chanlist>)]	90
[SOURce:]POWer:PROTection:STATe ON   OFF   1   0[, (@<chanlist>)]	90
[SOURce:]POWer:PROTection:STATe? [MINimum   MAXimum,] [(@<chanlist>)]	90
[SOURce:]POWer:RANGe <power>   MINimum   MAXimum[, (@<chanlist>)]	91
[SOURce:]POWer:RANGe? [MINimum   MAXimum,] [(@<chanlist>)]	91
[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>   MINimum   MAXimum   INFInity[, (@<chanlist>)]	92
[SOURce:]POWer:SLEW[:POSitive][:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)]	92
[SOURce:]POWer:SLEW:NEGAtive[:IMMediate] <value>   MINimum   MAXimum   INFInity[, (@<chanlist>)]	92
[SOURce:]POWer:SLEW:NEGAtive[:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)]	92
[SOURce:]POWer:SLEW:COUPlE ON   OFF   1   0[, (@<chanlist>)]	92
[SOURce:]POWer:SLEW:COUPlE? [(@<chanlist>)]	92
[SOURce:]POWer:SLEW[:POSitive]:MAXimum ON   OFF   1   0[, (@<chanlist>)]	93
[SOURce:]POWer:SLEW[:POSitive]:MAXimum? [(@<chanlist>)]	93
[SOURce:]POWer:SLEW:NEGAtive:MAXimum ON   OFF   1   0[, (@<chanlist>)]	93
[SOURce:]POWer:SLEW:NEGAtive:MAXimum? [(@<chanlist>)]	93
[SOURce:]POWer:TLEVel <value>   MINimum   MAXimum[, (@<chanlist>)]	93
[SOURce:]POWer:TLEVel? [MINimum   MAXimum,] [(@<chanlist>)]	93
<b>RESistance Subsystem</b>	<b>94</b>
[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <resistance>   MINimum   MAXimum   DEFault[, (@<chanlist>)]	94
[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MINimum   MAXimum   DEFault,] [(@<chanlist>)]	94
[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <resistance>   MINimum   MAXimum[, (@<chanlist>)]	94
[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]? [MINimum   MAXimum,] [(@<chanlist>)]	94
[SOURce:]RESistance:MODE FIXed   STEP   LIST[, (@<chanlist>)]	95
[SOURce:]VOLTage:MODE? [(@<chanlist>)]	95
[SOURce:]RESistance:RANGe <resistance>   MINimum   MAXimum[, (@<chanlist>)]	96
[SOURce:]RESistance:RANGe? [MINimum   MAXimum,] [(@<chanlist>)]	96
[SOURce:]RESistance:SLEW[:POSitive][:IMMediate] <value>   MINimum   MAXimum   INFInity[, (@<chanlist>)]	97
[SOURce:]RESistance:SLEW[:POSitive][:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)]	97
[SOURce:]RESistance:SLEW:NEGAtive[:IMMediate] <value>   MINimum   MAXimum   INFInity[, (@<chanlist>)]	97
[SOURce:]RESistance:SLEW:NEGAtive[:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)]	97
[SOURce:]RESistance:SLEW:COUPlE ON   OFF   1   0[, (@<chanlist>)]	97
[SOURce:]RESistance:SLEW:COUPlE? [(@<chanlist>)]	97
[SOURce:]RESistance:SLEW[:POSitive]:MAXimum ON   OFF   1   0[, (@<chanlist>)]	98
[SOURce:]RESistance:SLEW[:POSitive]:MAXimum? [(@<chanlist>)]	98
[SOURce:]RESistance:SLEW:NEGAtive:MAXimum ON   OFF   1   0[, (@<chanlist>)]	98
[SOURce:]RESistance:SLEW:NEGAtive:MAXimum? [(@<chanlist>)]	98

[SOURce:]RESistance:TLEVel <value>   MINimum   MAXimum[, (@<chanlist>)]	98
[SOURce:]RESistance:TLEVel? [MINimum   MAXimum,][(@<chanlist>)]	98
<b>SENSe Subsystem</b>	<b>99</b>
SENSe:DLOG:FUNCTion:CURRent ON   OFF   1   0[, (@<chanlist>)]	99
SENSe:DLOG:FUNCTion:CURRent? [(@<chanlist>)]	99
SENSe:DLOG:FUNCTion:MINMax ON   OFF   1   0	99
SENSe:DLOG:FUNCTion:MINMax?	99
SENSe:DLOG:FUNCTion:VOLTage ON   OFF   1   0[, (@<chanlist>)]	99
SENSe:DLOG:FUNCTion:VOLTage? [(@<chanlist>)]	99
SENSe:DLOG:OFFSet <offset percent>	100
SENSe:DLOG:OFFSet?	100
SENSe:DLOG:PERiod <time>   MINimum   MAXimum	100
SENSe:DLOG:PERiod? [MINimum   MAXimum]	100
SENSe:DLOG:TIME <time>   MINimum   MAXimum	100
SENSe:DLOG:TIME? [MINimum   MAXimum]	100
SENSe:DLOG:TINTerval <time>   MINimum   MAXimum	101
SENSe:DLOG:TINTerval? [MINimum   MAXimum]	101
SENSe:FUNCTion:CURRent ON   OFF   1   0[, (@<chanlist>)]	101
SENSe:FUNCTion:CURRent? [(@<chanlist>)]	101
SENSe:FUNCTion:VOLTage ON   OFF   1   0[, (@<chanlist>)]	101
SENSe:FUNCTion:VOLTage? [(@<chanlist>)]	101
SENSe:SWEep:POINts <data point>   MINimum   MAXimum[, (@<chanlist>)]	102
SENSe:SWEep:POINts? [MINimum   MAXimum,][(@<chanlist>)]	102
SENSe:SWEep:OFFSet:POINt <offset points>   MINimum   MAXimum[, (@<chanlist>)]	102
SENSe:SWEep:OFFSet:POINts? [MINimum   MAXimum,][(@<chanlist>)]	102
SENSe:SWEep:TINTerval <time>   MINimum   MAXimum[, (@<chanlist>)]	103
SENSe:SWEep:TINTerval? [MINimum   MAXimum,][(@<chanlist>)]	103
<b>SOURce Subsystem</b>	<b>104</b>
Subsystems Using the Optional SOURce Keyword	104
<b>STATus Subsystem</b>	<b>105</b>
STATus:OPERation[:EVENT]?	105
STATus:OPERation:CONDition?	105
STATus:OPERation:ENABle <enable value>	106
STATus:OPERation:ENABle?	106
STATus:OPERation:NTRansition <value>[, (<@chanlist>)]	106
STATus:OPERation:NTRansition? [(<@chanlist>)]	106
STATus:OPERation:PTRansition <value>[, (<@chanlist>)]	107
STATus:OPERation:PTRansition? [(<@chanlist>)]	107
STATus:PRESet	107
STATus:QUESTionable[:EVENT]? [(<@chanlist>)]	108
STATus:QUESTionable:CONDition? [(<@chanlist>)]	108
STATus:QUESTionable:ENABle <value>[, (<@chanlist>)]	109
STATus:QUESTionable:ENABle? [(<@chanlist>)]	109
STATus:QUESTionable:NTRansition <value>[, (<@chanlist>)]	109
STATus:QUESTionable:NTRansition? [(<@chanlist>)]	109
STATus:QUESTionable:PTRansition <value>[, (<@chanlist>)]	110
STATus:QUESTionable:PTRansition? [(<@chanlist>)]	110
<b>SYSTEM Subsystem</b>	<b>111</b>
SYSTEM:BEEPPer[:IMMEDIATE]	111
SYSTEM:BEEPPer:STATe ON   OFF   1   0	111
SYSTEM:BEEPPer:STATe?	111
SYSTEM:COMMunicate:RLSTate LOCAL   REMote   RWLock	111



SYSTem:COMMunicate:RLState?	111
SYSTem:COMMunicate:TCPIP:CONTRol?	112
SYSTem:DATE <yyyy>,<mm>,<dd>	112
SYSTem:DATE?	112
SYSTem:ERRor[:NEXT]?	113
SYSTem:LOCal	113
SYSTem:REMote	113
SYSTem:RWLock	114
SYSTem:SECurity:IMMediate	114
SYSTem:TIME <hh>,<mm>,<ss>	114
SYSTem:TIME?	114
SYSTem:VERSion?	114
<b>TRANSient Subsystem</b>	<b>115</b>
[SOURce:]TRANSient:COUNT <value>   MINimum   MAXimum   INFinity[, (@<chanlist>)]	115
[SOURce:]TRANSient:COUNT? [MIN   MAX   INFinity,] [(@<chanlist>)]	115
[SOURce:]TRANSient:DCYCLE <value>   MINimum   MAXimum[, (@<chanlist>)]	115
[SOURce:]TRANSient:DCYCLE? [MINimum   MAXimum,] [(@<chanlist>)]	115
[SOURce:]TRANSient:FREQuency <value>   MINimum   MAXimum[, (@<chanlist>)]	115
[SOURce:]TRANSient:FREQuency? [MINimum   MAXimum,] [(@<chanlist>)]	115
[SOURce:]TRANSient:MODE CONTinuous   PULSe   TOGGle   LIST[, (@<chanlist>)]	116
[SOURce:]TRANSient:MODE? [(@<chanlist>)]	116
[SOURce:]TRANSient:TWIDth <value>   MINimum   MAXimum[, (@<chanlist>)]	116
[SOURce:]TRANSient:TWIDth? [MINimum   MAXimum,] [(@<chanlist>)]	116
<b>TRIGger Subsystem</b>	<b>117</b>
TRIGger:ACQuire[:IMMediate] [(@<chanlist>)]	117
TRIGger:ACQuire:CURRent[:LEVel] <value>   MINimum   MAXimum[, (@<chanlist>)]	117
TRIGger:ACQuire:CURRent[:LEVel]? [MINimum   MAXimum,] [(@<chanlist>)]	117
TRIGger:ACQuire:CURRent:SLOPe POSitive   NEGative[, (@<chanlist>)]	117
TRIGger:ACQuire:CURRent:SLOPe? [(@<chanlist>)]	117
TRIGger:ACQuire:SOURce BUS   CURRent<n>   EXTernal   IMMEdiate   VOLTage<n>   PIN<n>[, (@<chanlist>)]	118
TRIGger:ACQuire:SOURce? [(@<chanlist>)]	118
TRIGger:ACQuire:VOLTage[:LEVel] <value>   MINimum   MAXimum[, (@<chanlist>)]	118
TRIGger:ACQuire:VOLTage[:LEVel]? [MINimum   MAXimum,] [(@<chanlist>)]	118
TRIGger:ACQuire:VOLTage:SLOPe POSitive   NEGative[, (@<chanlist>)]	119
TRIGger:ACQuire:VOLTage:SLOPe? [(@<chanlist>)]	119
TRIGger:DLOG[:IMMediate]	119
TRIGger:DLOG:CURRent[:LEVel] <value>   MINimum   MAXimum[, (@<chanlist>)]	119
TRIGger:DLOG:CURRent[:LEVel]? [MINimum   MAXimum,] [(@<chanlist>)]	119
TRIGger:DLOG:CURRent:SLOPe POSitive   NEGative[, (@<chanlist>)]	120
TRIGger:DLOG:CURRent:SLOPe? [(@<chanlist>)]	120
TRIGger:DLOG:VOLTage[:LEVel] <value>   MINimum   MAXimum[, (@<chanlist>)]	120
TRIGger:DLOG:VOLTage[:LEVel]? [MINimum   MAXimum,] [(@<chanlist>)]	120
TRIGger:DLOG:VOLTage:SLOPe POSitive   NEGative[, (@<chanlist>)]	120
TRIGger:DLOG:VOLTage:SLOPe? [(@<chanlist>)]	120
TRIGger:DLOG:SOURce BUS   EXTernal   IMMEdiate   CURRent<n>   VOLTage<n>   LRSKey   OOOKey	121
TRIGger:DLOG:SOURce?	121
TRIGger:TRANSient[:IMMediate] [(@<chanlist>)]	121
TRIGger:TRANSient:DELay <value>   MINimum   MAXimum[, (@<chanlist>)]	122
TRIGger:TRANSient:DELay? [MINimum   MAXimum,] [(@<chanlist>)]	122
TRIGger:TRANSient:SOURce BUS   EXTernal   IMMEdiate   PIN1   PIN2   PIN3[, (@<chanlist>)]	122
TRIGger[:SEQUence]:SOURce? [(@<chanlist>)]	122
<b>Triggering Commands</b>	<b>123</b>

<b>VOLTage Subsystem</b> .....	<b>124</b>
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>   MINimum   MAXimum   DEFault[, (@<chanlist>)] .....	124
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum   MAXimum   DEFault,] [(@<chanlist>)] .....	124
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <voltage>   MINimum   MAXimum[, (@<chanlist>)] .....	124
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MINimum   MAXimum,] [(@<chanlist>)] .....	124
[SOURce:]VOLTage:INHibit:VON[:LEVel] <voltage>   MAXimum   MINimum   DEFault[, (@<chanlist>)] .....	125
[SOURce:]VOLTage:INHibit:VON[:LEVel]? [MINimum   MAXimum   DEFault,] [(@<chanlist>)] .....	125
[SOURce:]VOLTage:INHibit:VON:MODE LATCHing   LIVE   OFF[, (@<chanlist>)] .....	125
[SOURce:]VOLTage:INHibit:VON:MODE? [(@<chanlist>)] .....	125
[SOURce:]VOLTage:MODE FIXed   STEP   LIST[, (@<chanlist>)] .....	126
[SOURce:]VOLTage:MODE? [(@<chanlist>)] .....	126
[SOURce:]VOLTage:RANGe <voltage>   MINimum   MAXimum[, (@<chanlist>)] .....	127
[SOURce:]VOLTage:RANGe? [MINimum   MAXimum,] [(@<chanlist>)] .....	127
[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <slew rate>   MINimum   MAXimum   INFInity[, (@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <slew rate>   MINimum   MAXimum   INFInity[, (@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW:NEGative[:IMMediate]? [MINimum   MAXimum,] [(@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW:COUPle ON   OFF   1   0[, (@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW:COUPle? [(@<chanlist>)] .....	128
[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum ON   OFF   1   0[, (@<chanlist>)] .....	129
[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum? [(@<chanlist>)] .....	129
[SOURce:]VOLTage:SLEW:NEGative:MAXimum ON   OFF   1   0[, (@<chanlist>)] .....	129
[SOURce:]VOLTage:SLEW:NEGative:MAXimum? [(@<chanlist>)] .....	129
[SOURce:]VOLTage:SENSe[:SOURce] INTernal   EXTernal[, (@<chanlist>)] .....	130
[SOURce:]VOLTage:SENSe[:SOURce]? [(@<chanlist>)] .....	130
[SOURce:]VOLTage:TLEVel <value>   MINimum   MAXimum[, (@<chanlist>)] .....	130
[SOURce:]VOLTage:TLEVel? [MINimum   MAXimum,] [(@<chanlist>)] .....	130

# Notices

## Copyright Notice

© Keysight Technologies 2020, 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

EL34243-90007

## 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](http://about.keysight.com/en/companyinfo/environment/takeback.shtml) for more information.



## Declarations of Conformity

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

## Safety Information

### CAUTION

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

---

### WARNING

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

---

# 1 Remote Operation

Introduction to the SCPI Language

Programming Ranges

SCPI Error Messages

Reset and Non-volatile Settings

SCPI Status Registers

This chapter describes the remote operation for the EL30000 Series DC electronic loads

## Introduction to the SCPI Language

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

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

[SOURCE:]

CURRENT <current> | MIN | MAX

CURRENT? MIN | MAX

CURRENT:

    TRIGGERED <current> | MIN | MAX

    TRIGGERED? MIN | MAX

VOLTAGE <voltage> | MIN | MAX

VOLTAGE? MIN|MAX

VOLTAGE:

    TRIGGERED <voltage> | MIN | MAX

    TRIGGERED? MIN | MAX

SOURCE is the root keyword of the command, CURRENT and VOLTAGE are second-level keywords, and TRIGGERED is the third-level keyword. A colon (:) separates a command keyword from a lower-level keyword.

## Command Format Used in this Manual

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

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

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

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

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

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

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

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

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

```
APPLY 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 input current to 5 A:

```
CURR 5
```

You can query the value by executing:

```
CURR?
```

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

```
CURR? MAX; CURR? MIN
```

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

---

## SCPI Command Terminators

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

## IEEE-488.2 Common Commands

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

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

## SCPI Parameter Types

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

### Channel parameters

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

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

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

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

For example, (@2) specifies channel 2 and (@1:3) specifies channels 1 through 3. The channel list, shown as <chanlist> throughout this document, must be preceded with the @ symbol and must be enclosed in parentheses (). A maximum of 4 channels may be specified through a combination of single channels and ranges. Query results are channel list order-sensitive. Results are returned in the order they are specified in the list.

#### NOTE

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

---

### Numeric parameters

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

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

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

### Discrete parameters

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

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

### Boolean parameters

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

```
DISP OFF | ON
```

## String parameters

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

```
DISP:TEXT <quoted string>
```

## Halting an Input in Progress

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

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

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

## Programming Ranges

The following table shows the input ranges (voltage, current, power, and resistance) and its default value that can be programmed for each model in EL30000 Series.

Operating Mode	Range		EL33133A	EL34143A	EL34243A	
CV	High	MAXimum	153 V	153 V	Single/Parallel: 153 V	
		MINimum	0.02 V	0.015 V	Single/Parallel: 0.015 V	
	Low	MAXimum	15.3 V	15.3 V	Single/Parallel: 15.3 V	
		MINimum	0.005 V	0.003 V	Single/Parallel: 0.003 V	
	DEFAULT (*RST)		0.02 V	0.015 V	Single/Parallel: 0.015 V	
CC	High	MAXimum	40.8 A	61.2 A	Single: 61.2 A Parallel: 122.4 A	
		MINimum	0.01 A	0.012 A	Single: 0.012 A Parallel: 0.024 A	
	Medium	MAXimum	-	6.12 A	Single: 6.12 A Parallel: 12.24 A	
		MINimum	-	0.002 A	Single: 0.002 A Parallel: 0.004 A	
	Low	MAXimum	4.08 A	0.612 A	Single: 0.612 A Parallel: 1.224 A	
		MINimum	0.001 A	0.0002 A	Single: 0.0002 A Parallel: 0.0004 A	
	DEFAULT (*RST)		0.01 A	0.012 A	0.012 A	
	CP	High	MAXimum	255 W	357 W	Single: 306 W Parallel: 612 W
			MINimum	1.5 W	2 W	Single: 2 W Parallel: 4 W
		Medium	MAXimum	25.5 W	35.7 W	Single: 30.6 W Parallel: 61.2 W
MINimum			0.15 W	0.3 W	Single: 0.2 W Parallel: 0.4 W	
Low		MAXimum	5.1 W	8.16 W	Single: 7.14 W Parallel: 14.28 W	
		MINimum	0.02 W	0.01 W	Single: 0.01 W Parallel: 0.02 W	
DEFAULT (*RST)			1.5 W	2 W	2 W	

Operating Mode	Range		EL33133A	EL34143A	EL34243A
CR	Ultra-High	MAXimum	-	100 k $\Omega$	Single: 100 k $\Omega$ Parallel: 50 k $\Omega$
		MINimum	-	250 $\Omega$	Single: 250 $\Omega$ Parallel: 125 $\Omega$
High	High	MAXimum	4 k $\Omega$	4000 $\Omega$	Single: 4000 $\Omega$ Parallel: 2000 $\Omega$
		MINimum	100 $\Omega$	100 $\Omega$	Single: 100 $\Omega$ Parallel: 50 $\Omega$
Medium	Medium	MAXimum	1.25 k $\Omega$	1250 $\Omega$	Single: 1250 $\Omega$ Parallel: 625 $\Omega$
		MINimum	10 $\Omega$	10 $\Omega$	Single: 10 $\Omega$ Parallel: 5 $\Omega$
Low	Low	MAXimum	30 $\Omega$	30 $\Omega$	Single: 30 $\Omega$ Parallel: 15 $\Omega$
		MINimum	0.08 $\Omega$	0.05 $\Omega$	Single : 0.05 $\Omega$ Parallel: 0.025 $\Omega$
Default (*RST)			4 k $\Omega$	100 k $\Omega$	100 k $\Omega$

## 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: <code>OUTP:TRAC #ON</code>
-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: <code>VOLT:LEV,1</code>
-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: <code>TRIG:SOUR,BUS</code> or <code>APPL CH1 1.0 1.0</code>
-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 electronic load. 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	Exponent too large A numeric parameter was found whose exponent was larger than 32,000.
-124	Too many digits A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.
-128	Numeric data not allowed A numeric parameter was received but a character string was expected. Example: DISP:TEXT 123
-130	Suffix error A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix or the numeric parameter does not accept a suffix. Example: TRIG:DEL 0.5 SECS
-131	Invalid suffix A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix. Example: TRIG:DEL 0.5 SECS
-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.
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error
-200	Execution error A generic syntax error.
-201	Invalid while in local



Code	Text
-202	Settings lost due to rtl
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error A data element related error occurred.
-221	Settings conflict A data element could not be executed because of the present instrument state.
-222	Data out of range A numeric parameter value is outside the valid range for the command. Example: TRIG:DEL -3
-223	Too much data A character string was received but could not be executed because the string length was more than 40 characters. This error can be generated by the CALibration:STRing command.
-224	Illegal parameter value A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice. Example: DISP:STAT XYZ (XYZ is not a valid choice).
-225	Out of memory The device has insufficient memory to perform the requested operation.
-230	Data corrupt or stale Possible invalid data. A new reading was started but not completed.
-231	Data questionable The measurement accuracy is suspect.
-240	Hardware error The command could not be executed because of a hardware problem with the instrument.
-241	Hardware missing The command could not be executed because of missing hardware, such as an option.
-250	Mass storage error
-251	Missing mass storage
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full
-256	File name not found
-257	File name error
-258	Media protected

Code	Text
-260	Expression error An expression program data element related error occurred.
-261	Math error in expression An expression program data element could not be executed due to a math error.
-270	Macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label
-274	Macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory error
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-321	Out of memory
-330	Self-test failed
-350	Queue overflow
-363	Input buffer overrun
-400	Query error A generic error query.
-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 <b>*RST</b> (reset) command has been executed.

Code	Text
-420	Query UNTERMINATED The electronic load was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed an APPLY command (which does not generate data) and then attempted an ENTER statement to read data from the remote interface.
-430	Query DEADLOCKED A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.
-440	Query UNTERMINATED after indefinite response A query was received in the same program message after a query indicating an indefinite response was executed.
304	Volt and curr in incompatible transient modes Voltage and current cannot be in Step and List mode at the same time.
308	This command is not allow while list is running
309	BOSTep list length does not match dwell list
310	EOSTep list length does not match dwell list
311	Voltage list length does not match dwell list
312	Current list length does not match dwell list
313	Power list length does not match dwell list
314	Resistance list length does not match dwell list
315	LIST and STEP transient modes are not compatible
316	Channel can have only one transient mode set
317	No transient has been configured
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	Settings conflict error; chan must be in CC mode
562	Settings conflict error; chan must be in CV mode
563	Settings conflict error; chan must be in CP mode
564	Settings conflict error; chan must be in CR mode
565	Settings conflict error; chan must be in CV or CR or CP mode
566	Settings conflict error; chan must be in CC or CR or CP mode
567	CH1 over temperature

Code	Text
568	CH2 over temperature
571	This instrument is currently operating in high temperature and may impact the accuracy
601	FPGA transmit timeout
602	Front panel failed to enter boot loader
603	CH1 FPGA failed to communicate
604	CH2 FPGA failed to communicate
605	Primary FPGA failed to communicate
606	Front panel does not respond
610	Fan test failed
611	EEPROM load failed
612	EEPROM checksum failed
613	EEPROM save failed
614	Invalid serial number
615	Invalid MAC address
720	CH2 is not allowed when parallel is enabled CH2 is not allowed to be used/disabled when parallel is enabled.
721	Calibration is not allowed when parallel is enabled Calibration is not allowed to be used/disabled when parallel 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 or acquisition is running Data can't be alter as trigger has already been initiated or acquisition is running. Abort the trigger or acquisition to change data.
736	Trig init disallow as channel is coupled with another channel that has trig initiated
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 mode Change the mode before doing this operation.
744	There is not a valid acquisition to fetch from
745	Initiate with no sense functions enabled
750	USB not connected This may be due to data logger requiring a USB thumb drive to be connected.
751	USB host access failed This may be due to DUT fails to access the USB thumb drive.
752	Insufficient space in USB drive This may be due to the memory size required by data logger is larger than the available free space.
753	Data logger is running Data logger setting can't be changed while it's running.

Code	Text
754	Data logger do not have valid data Fetch is not allowed as data logger doesn't have valid data.
755	Measurement is not allowed while data logger running
756	Sample period cannot be more than logging duration
770	Too many measurement point
800	CH1 and CH2 coupled by track system The OUTP:TRAC should be off when coupling between the CH1 input and CH2 input.
801	CH1 and CH2 coupled by trigger subsystem The CH1 input and CH2 input should be uncoupled to enable the tracking operation for those inputs.
900	Firmware update failed

## Calibration error codes

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

Code	Text
577	CH1 Invalid Cal data
578	CH2 Invalid Cal data
581	Invalid state. Cal secured
582	Invalid secure code
583	Secure code too long
584	Invalid Calibration sequence
585	Cal program constants out of range
586	Cal AUX IMON constants out of range
587	Cal measure constants out of range
588	Cal AUX VLOC constants out of range
592	This action is not allowed as calibration has not completed
594	Cal trim dac not centered (prog0)
595	Cal trim dac not centered (prog1)

## Self-test error codes

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

Code	Text
533	(CH1) Chan 1 Precision DAC test failed
534	(CH2) Chan 2 Precision DAC test failed
536	(CH1) Chan 1 Auxiliary DAC test failed
537	(CH2) Chan 2 Auxiliary DAC test failed
539	(CH1) Incorrect model number
540	(CH2) Incorrect model number
542	(Front panel) Incorrect model number
550	3.3V power lost
551	5.0V power lost
552	12V power lost

## 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
CHANnel   INSTRument:SElect	CH1
CHANel   INSTRument:NSElect	1
CHANel   INSTRument:LOAD	1
CURRent	MIN * If parallel, value is twice the default value
CURRent:TRIGgered	MIN
CURRent:LIMit	MAX
CURRent:PROTection:STATe	OFF
CURRent:PROTection:DELay	0.02 s
CURRent:PROTection:DELay:START	SCH
CURRent:MODE	FIX
CURRent:RANGe	MAX
CURRent:SLEW	MAX
CURRent:SLEW:COUPle	OFF
CURRent:SLEW:MAX	ON
CURRent:SLEW:NEGative	MAX
CURRent:SLEW:NEGative:MAX	ON
CURRent:TLevel	MIN
DIGital:OUTPut:DATA	0
DIGital:TOUTput:BUS	OFF
DISPlay	ON
DISPlay:TEXT	"" (empty)
DISPlay:VIEW	METER2
FUNction   MODE	CURR
INITiate:DLOG	- (none)
INITiate:CONTInuous:TRANSient	OFF
INPut   OUTPut:STATe	OFF
INPut   OUTPut:STATe:DELay:FALL	0
INPut   OUTPut:STATe:DELay:RISE	0
INPut   OUTPut:INHibit:MODE	OFF

SCPI command	Default value
INPut   OUTPut:PAIR	OFF
INPut   OUTPut:SHORT	OFF
LIST:VOLTage	1 step set to the minimum programmable value
LIST:CURRent	1 step set to the minimum programmable value
LIST:RESistance	1 step set to the minimum programmable value
LIST:POWer	1 step set to the minimum programmable value
LIST:DWELL	0.001 s
LIST:COUNT	1
LIST:TERMinate:LAST	OFF
LIST:STEP	AUTO
LIST:TOUTput:BOSTep	OFF
LIST:TOUTput:EOSTep	OFF
MMEMory:EXPort:DLOG	- (none)
POWer	MIN * If parallel, value is twice the default value
POWer:TRIGgered	MIN
POWer:MODE	FIX
POWer:RANGe	MAX
POWer:SLEW	MAX
POWer:SLEW:COUPlE	OFF
POWer:SLEW:MAX	ON
POWer:SLEW:NEGative	MAX
POWer:SLEW:NEGative:MAX	ON
POWer:PROTection:DELay	0.02 s
POWer:PROTection:STATe	OFF
POWer:TLEVel	MIN
RESistance	MAX * If parallel, value is twice the default value
RESistance:TRIGgered	MAX
RESistance:MODE	FIX
RESistance:RANGe	MAX
RESistance:SLEW	MAX
RESistance:SLEW:COUPlE	OFF
RESistance:SLEW:MAX	ON
RESistance:SLEW:NEGative	MAX
RESistance:SLEW:NEGative:MAX	ON
RESistance:TLEVel	MAX
SENSe:DLOG:FUNCTion:CURRent	OFF



SCPI command	Default value
SENSe:DLOG:FUNcTION:MINM	OFF
SENSe:DLOG:FUNcTION:VOLTage	ON
SENSe:DLOG:PERiod	0.2 s
SENSe:DLOG:TIME	30 s
SENSe:DLOG:TINterval	0.2 s
SENSe:FUNcTION:CURRent	ON
SENSe:FUNcTION:VOLTage	ON
SENSe:SWEp:POINts	2440
SENSe:SWEp:OFFSet:POINts	0
SENSe:SWEp:TINterval	0.00004096 s
SYSTem:COMMunicate:RLState	LOC
TRANsient:DCYCLE	50%
TRANsient:FREQuency	MAX
TRANsient:MODE	LIST
TRANsient:TWIDth	0.0005 s
TRANsient:COUNt	1
TRIGger:ACQuire:SOURce	BUS
TRIGger:ACQuire:CURRent	MIN
TRIGger:ACQuire:CURRent:SLOPe	POS
TRIGger:ACQuire:VOLTage	MIN
TRIGger:ACQuire:VOLTage:SLOPe	POS
TRIGger:TRANsient:DELay	0
TRIGger:TRANsient:SOURce	BUS
TRIGger:DLOG:SOURce	IMM
VOLTage	MIN
VOLTage:TRIGgered	MIN
VOLTage:MODE	FIX
VOLTage:RANGe	MAX
VOLTage:SENSe	INT
VOLTage:SLEW	MAX
VOLTage:SLEW:COUPlE	OFF
VOLTage:SLEW:MAX	ON
VOLTage:SLEW:NEGative	MAX
VOLTage:SLEW:NEGative:MAX	ON
VOLTage:TLEVel	MIN
VOLTage:INHibit:VON	MIN

SCPI command	Default value
VOLTage:INHibit:VON:MODE	OFF

## Non-volatile settings

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

SCPI command	Default value
INPut   OUTPut:COUPle:CHANnel (channel grouping)	NONE
INPut   OUTPut: INHibit:MODE (Output Inhibit mode)	OFF
INPut   OUTPut:PON:STATE	*RST
DIGital:PIN<1-3>:FUNCTion (digital port function)	DINP
DIGital:PIN<1-3>:POLarity (digital port polarity)	POS
Calibration password	0 (zero)
Calibration date	-
GPIB Address	5
SYSTem:BEEPer:STATe	ON
LXI:IDENtify:STATe	0
LXI:MDNS:STATe	0
DIAGnostic:LCD:BRIGhtness	100
SYSTem:DATE	-
SYSTem:TIME	-

## Non-volatile LAN settings

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

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

## SCPI Status Registers

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

### Status Registers

#### Operation Status Group

#### Questionable Status Group

#### Standard Event Status Group

#### Status Byte Register

#### Error and Input Queues

#### Status Diagram

## Status Registers

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

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

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

## Operation Status Group

These registers record signals that occur during normal operation. The group consists of a Condition, PTR/NTR, Event, and Enable register. The outputs of the Operation Status register group are logically-ORed into the OPERation summary bit (7) of the Status Byte register. Refer to **Status Registers** for a description of each register. The following table describes the bit assignments.

Bit	Bit Name	Decimal Value	Definition
0	CV	1	Input is in constant voltage.
1	CC	2	Input is in constant current.
2	CR	4	Input is in constant resistance mode.
3	CP	8	Input is in constant power mode.
4	OFF	not used	Input is programmed off.
5	SH	32	Load input terminals are shorted.
6	WTG MEAS	64	Measurement system is waiting for a trigger.
7	WTG Tran	128	List and transient system is waiting for a trigger.
8	WTG Dlog	256	Data logging system is waiting for trigger.
9	MEAS Active	512	Measurement system is initiated.
10	TRAN Active	1024	Transient system is initiated.
11	DLOG Active	2048	Data logging system is initiated.
12-15	not used	not used	0 is returned.

## Questionable Status Group

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

Bit	Bit Name	Decimal Value	Definition
0	OV	1	Input is disabled by the over-voltage protection.
1	OC	2	Input is disabled by the over-current protection.
2	not used	not used	0 is returned
3	CP+	8	Input is limited (or disabled) by the positive power limit.
4	OT	16	Over-temperature protection has tripped.
5	OV-	32	Negative over-voltage protection has tripped.
6	LIM-	54	In CV mode, the input is in negative current limit. In CC mode, the input is in negative voltage limit.  In CR and CP modes, the input is in positive current limit.
7	UNR	128	Input is unregulated.
8	INH	256	Input is inhibited by an external INHhibit signal.
9	UVI	512	Input is disabled by the under-voltage protection.
10-15	not used	not used	0 is returned

## Standard Event Status Group

These registers are programmed by Common commands. The group consists of an Event and Enable register. The Standard Event event register latches events relating to communication status. It is a read-only register that is cleared when read. The Standard Event enable register functions similarly to the enable registers of the Operation and Questionable status groups. Refer to **Status Registers** for a description of each register. The following table describes the bit assignments.

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

## Status Byte Register

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

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

## MSS and RQS Bits

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

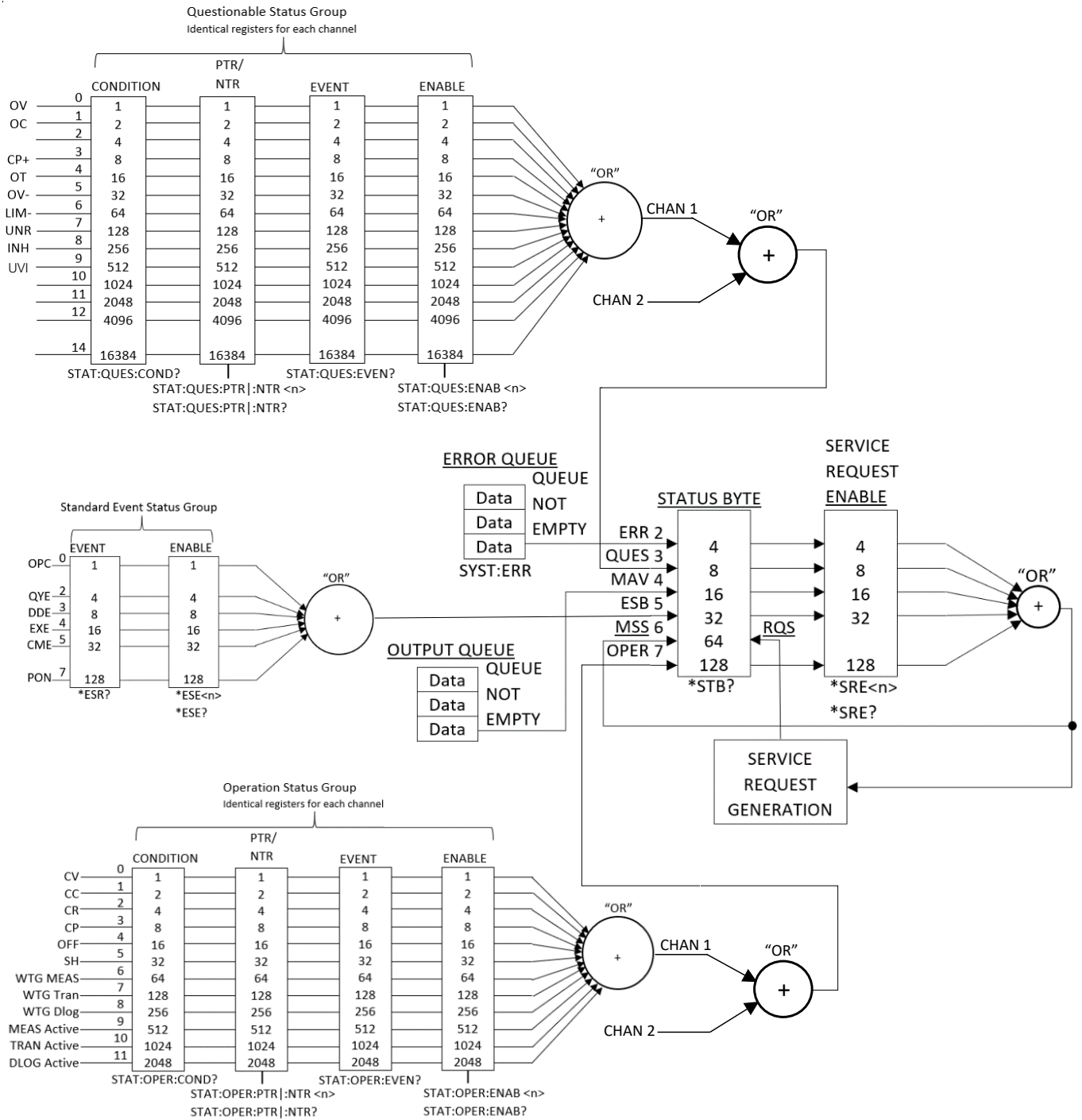
The RQS bit is a latched version of the MSS bit. Whenever the instrument requests service, it sets the SRQ interrupt line true and latches RQS into bit 6 of the Status Byte register. When the controller does a serial poll, RQS is cleared inside the register and returned in bit position 6 of the response.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

## Error and Output Queues

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

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

# Status Diagram



# 2 SCPI Programming

- ABORt Subsystem
- CALibration Subsystem
- CHANnel | INSTRument Subsystem
- CURRent Subsystem
- DIGital Subsystem
- DISPlay Subsystem
- FETCh Subsystem
- FUNCTION | MODE Subsystem
- IEEE-488.2 Common Commands
- INITiate Subsystem
- INPut | OUTPut Subsystem
- LIST Subsystem
- LXI Subsystem
- MEASure Subsystem
- MMEMory Subsystem
- POWer Subsystem
- RESistance Subsystem
- SENSe Subsystem
- STATus Subsystem
- SYSTem Subsystem
- TRANsient Subsystem
- TRIGger Subsystem
- Triggering Commands
- VOLTage Subsystem

This chapter describes the subsystem commands available to the EL30000 Series DC electronic loads.



## ABORt Subsystem

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

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

Parameter	Typical return
(@<chanlist>)	(none)
Cancels the triggered measurement of channel 1: ABOR:ACQ (@1)	

#### Remarks

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

### ABORt:DLOG

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

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

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

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

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

#### Remarks

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

## CALibration Subsystem

**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 when leaving the CAL state, the calibration data will automatically move the CAL data from volatile memory to non-volatile memory and the CALibration:COUNT value is increased by 1.

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

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

CALibration:COUNt?

The query returns the number of times the electronic load has been calibrated. Your electronic load was calibrated before it left the factory. When you receive your electronic load, read the count to determine its initial value. Since the value increases by one for each calibration point, a complete calibration for two inputs will increase the value by two counts.

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

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

CALibration:CURREnt[:DATA]:MID <current>[, (@<chanlist>)]

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

The command allows you to enter an input current value that you obtained by reading an external meter. However you must first select a calibration level (CAL:CURREnt:LEV) for the value entered. This command can only be used when calibration is unsecured and input is ON.

Parameter	Typical return
<current>	(none)
Sets the input current value (high range) obtained by reading a DMM for example 5 A: CAL:CURREnt 5	
Sets the input current value (middle range) obtained by reading a DMM for example 1 A: CAL:CURREnt:MID 1	
Sets the input current value (low range) obtained by reading a DMM for example 0.5 A: CAL:CURREnt:LOW 0.5	

CALibration:CURRent:LEVel[:HIGH] POINT1 | POINT2 | POINT3 | POINT4 | POINT5[, (@<chanlist>)]

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

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

Before using this command, you must select the input which is to be calibrated by using the **INSTrument** command. This command can only be used after calibration is unsecured. It sets the electronic load 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 or POINT1) must be selected and entered first.

Parameter	Typical return
HIGH range: POINT1   POINT2   POINT3   POINT4   POINT5	(none)
MID and LOW range: MIN   MAX	
Sets the input current (high range) to POINT1: CAL:CURR:LEV POINT1, (@1)	
Sets the input current (low range) to minimum: CAL:CURR:LEV:LOW MIN, (@1)	

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

This command can only be used after calibration is unsecured. It enters a current limit value of a selected input that you obtained by reading an external meter. You must first select a calibration level (CAL:CURL:LEV) for the value being entered. Two successive values (one for each end of the calibration range) must be selected and entered. The electronic load then computes new current limit calibration constants. These constants are then stored in non-volatile memory.

Parameter	Typical return
<current>	(none)
Sets the input current limit value, obtained by reading an external meter for example 5 A: CAL:CURL 5, (@1)	

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

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

Parameter	Typical return
MIN   MAX	(none)
Sets the electronic load to a calibration point that is entered with CALibration:CURLimit [:DATA]: CAL:CURL LEV 5, (@1)	

CALibration:DATE "<string>"

CALibration:DATE?

The command stores the date that the electronic load 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 passcode>

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

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

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

CALibration:SECure:STATe?

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

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

Parameter	Typical return
ON   OFF   1   0, <passcode>	0 or 1
Secures the electronic load 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 electronic load'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]:HIGH <numeric value>[, (@<chanlist>)]  
CALibration:VOLTage[:DATA]:LOW <numeric value>[, (@<chanlist>)]

The command enters a voltage value of a selected input 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 electronic load then computes the new calibration constants and stores them in non-volatile memory.

Parameter	Typical return
<numeric value>	(none)
Sets the high input voltage value (for channel 1), obtained by reading a DMM, for example 10 V: CAL:VOLT 10, (@1)	
Sets the low input voltage value (for channel 1), obtained by reading a DMM, for example 0.5 V: CAL:VOLT 0.5, (@1)	

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

The command sets the electronic load 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 input 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	

## CHANnel | INSTRument Subsystem

These commands program the channel selection capability of the electronic load. The CHANnel and INSTRument commands are equivalent.

**NOTE** For the EL33133A and EL34143A single channel models, only CH1 parameter is applicable. The CH1 | CH2 parameter is only applicable for the EL34243A dual channel model. You can identify each input by the input 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.

CHANnel[:SElect] [CH1 | CH2]

CHANnel[:SElect]?

INSTRument[:SElect] [CH1 | CH2]

INSTRument[:SElect]?

The command selects the input to be programmed among two inputs by the input identifier. The inputs of the EL34243A electronic load are considered as two logical instruments. The CHANnel | INSTRument command provides a mechanism to identify and select an input. When one input is selected, the other input is unavailable for programming until it is selected.

The query returns the currently selected input by the CHANnel | INSTRument[:SElect] or CHANnel | INSTRument:NSElect command.

Parameter	Typical return
CH1   CH2	CH1 or CH2
*RST CH1	
Selects channel 1 to be programmed: INST CH1 or CHAN CH1	

CHANnel:NSElect [1 | 2]

CHANnel:NSElect?

INSTRument:NSElect [1 | 2]

INSTRument:NSElect?

The command selects the input to be programmed among two inputs by a numeric value instead of the input identifier used in the INSTRument[:SElect] command. “1” selects CH1 input and “2” selects CH2 input.

The query returns the currently selected input by the INSTRument:NSElect or INSTRument[:SElect] command. The returned parameter is “1” for CH1 input, “2” for CH2 input.

Parameter	Typical return
1   2	1 or 2
*RST 1	
Selects channel 1 to be programmed: INST:NSEL 1 or CHAN:NSEL 1	

CHANnel[:LOAD] [1 | 2]

CHANnel[:LOAD]? MAX

INSTrument[:LOAD] [1 | 2]

INSTrument[:LOAD]? MAX

The command selects multiple inputs from the electronic load to which all subsequent input-specific commands will be directed.

The query returns the number of inputs in the electronic load. The returned parameter is "1" for one input, "2" for 2 inputs.

Parameter	Typical return
1   2	1 or 2
Selects one input to be programmed: INST 1 or CHAN 1	
Returns the number of inputs in the electronic load: INST? MAX or CHAN? MAX	

## CURRent Subsystem

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

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

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

The command sets the immediate current level when the input is in constant current mode. Units are in amperes.

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

Parameter	Typical return
minimum - maximum   MIN   MAX   DEF (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<current level>
*RST MIN	
<hr/>	
Sets the input current level to 3 A: CURR 3, (@1)	

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

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

The command sets the triggered current level when the input is in constant current mode. The triggered level is a stored value that will be programmed when a Step transient is triggered. Units are in amperes.

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

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<current level>
*RST MIN	
<hr/>	
Sets the triggered current to 1 A on channels 1 and 2: CURR:TRIG 1, (@1,2)	



[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <current> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum | MAXimum,] [ (@<chanlist>)]

The command sets the current limit when in CV mode. Units are in amperes. The minimum value is the most negative value.

Parameter	Typical Return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<+ current limit>
*RST MAX	
Sets the current limit to 2 A: CURR:LIM 2 (@1)	

### Remarks

- MIN is the most negative value of the range. MAX is the least negative value of the range.
- The query returns the programmed current limit in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

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

The command determines what happens to the input 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 input current remains at its immediate value.
STEP	The input goes to the triggered level when a trigger occurs.
LIST	The input 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 1 to Step: CURR:MODE STEP, (@1)	

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

The command sets the over-current protection delay time in milliseconds. The over-current protection function will not be triggered during the delay time. After the delay time has expired, the over-current protection function will be active. This prevents momentary changes in output status from triggering the over-current protection function. Values up to 255 milliseconds can be programmed, with a resolution of 1 millisecond.

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

Parameter	Typical return
0 – 0.255   MIN   MAX	<delay value>
*RST 0.02	
Sets the protection delay to 0.2 seconds for channel 1: CURR:PROT:DEL 0.2, (@1)	

### Remarks

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

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

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

Mode	Description
SCHange	Transitions into constant current mode are automatically ignored during a programmed settings change in voltage, current, or input 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 input 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 input status is in constant current mode for a time longer than the OCP delay. Input will be OFF after OCP is tripped.

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

Parameter	Typical return
ON   OFF   1   0	0 or 1
*RST OFF	
Enables the current protection state: CURR:PROT:STAT ON, (@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:]CURRent:RANGe <current> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]CURRent:RANGe? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the current range. The value that you enter must be the highest value in amperes that you expect to input. The instrument selects the range with the best resolution for the value entered. The following current ranges are available for the electronic load:

	EL33133A	EL34143A	EL34243A
High range	from 0.01 A to 40.8 A	from 0.012 A to 61.2 A	from 0.012 A to 61.2 A
Medium range	-	from 0.002 A to 6.12 A	from 0.002 A to 6.12 A
Low range	from 0.001 A to 4.08 A	from 0.0002 A to 0.612 A	from 0.0002 A to 0.612 A

The query returns the maximum current that can be programmed on the range that is presently set. The current is in the form +n.nnnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <b>Programming Range</b> )	<current range>
*RST MAX	
Sets the medium current range (from 0.001 A to 6.12 A): CURR:RANG 3, (@1)	

## Remarks

- This command is coupled with [SOURce:]CURRent. This means that if a range command places an input on a range with a lower maximum current than the present current level, an out of range error occurs. This also occurs if a current is programmed that is too large for the present range. Errors can be avoided by sending both level and range commands in the same SCPI message. When the range and level information is received as a set, no range/setting conflict occurs.
- If a programmed range change occurs while the input is enabled, the input will be temporarily disabled while the range switch occurs. The transition from on-to-off and then from off-to-on will also be delayed by the settings of **INPut:DElay:FALL** and **INPut:DElay:RISE**.

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <slew rate> | MINimum | MAXimum | INFinity[, (@<chanlist>)]

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate]? [MINimum | MAXimum,] [ (@<chanlist>)]

[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <slew rate> | MINimum | MAXimum | INFinity[, (@<chanlist>)]

[SOURce:]CURRent:SLEW:NEGative[:IMMediate]? [MINimum | MAXimum,] [ (@<chanlist>)]

The command sets the current slew rate. The slew rate is set in amps per second and affects all programmed current changes, including those due to the input state turning on or off. The slew rate can be set to any value between the minimum value and 9.9E+37. For very large values, the slew rate will be limited by the analog performance of the input circuit. The keywords MAX or INFinity set the slew rate to maximum.

Parameter	Typical Return
0 - 9.9E+37   MIN   MAX   INF	<slew rate>
*RST MAX	
Sets the positive input slew rate to 1 A per second: CURR:SLEW 1, (@1)	
Sets the negative input slew rate to 1 A per second: CURR:SLEW:NEG 1, (@1)	

#### Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using CURRent:SLEW? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E+37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]CURRent:SLEW:COUPle ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]CURRent:SLEW:COUPle? [(@<chanlist>)]

The command sets the current slew tracking state.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Enables the current slew tracking state: CURR:SLEW:COUP ON, (@1)	

[SOURce:]CURRent:SLEW[:POSitive]:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]  
[SOURce:]CURRent:SLEW[:POSitive]:MAXimum? (@<chanlist>)  
[SOURce:]CURRent:SLEW:NEGative:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]  
[SOURce:]CURRent:SLEW:NEGative:MAXimum? [(@<chanlist>)]

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by the CURRent:SLEW command. Use CURRent:SLEW? MAX to query the maximum slew rate.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST ON	
Disables the positive maximum slew rate: CURR:SLEW:MAX OFF, (@1)	
Disables the negative maximum slew rate: CURR:SLEW:NEG:MAX OFF, (@1)	

### Remark

- The CURRent:SLEW:MAX command is coupled to the CURRent:SLEW command. If the CURRent:SLEW command sets the slew rate to MAX or INFINITY, CURRent:SLEW:MAX is enabled. If the slew rate is set to any other value, CURRent:SLEW:MAX is disabled.

[SOURce:]CURRent:TLEVel <transient level> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]CURRent:TLEVel? [MINimum | MAXimum,] [ (@<chanlist>)]

The command specifies the transient of the input current. The transient function switches between the immediate setting and the transient level. Units are in amperes.

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

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <b>Programming Range</b> )	<transient level>
*RST MIN	
Sets the transient input current to 5 A: CURR:TLEV 5, (@1)	

## DIGital Subsystem

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

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

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

### [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: <code>DIG:OUTP:DATA 5</code>	

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

[SOURce:]DIGital:PIN<1-3>:FUNCtion?

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

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

Function	Description
DIO	The pin is a general-purpose ground-referenced digital input/output. The output can be set with [SOURce:]DIGital:OUTPut:DATA.
DINPut	The pin is in digital input-only mode. The digital output data of the corresponding pin is ignored.
TOUTput	The pin is configured as a trigger output. When configured as a trigger output, the pin will only generate output triggers if the List transient system has been configured to generated trigger signals. See [SOURce:]STEP:TOUTput, [SOURce:]LIST:TOUTput:BOSTep, and [SOURce:]LIST:TOUTput:EOSTep.
TINPut	The pin is configured as a trigger input. When configured as a trigger input, the pin can be selected as the source for transient trigger signals. See TRIGger[:SEQuence]:SOURce.
FAULt	Applies only to pin 1. Setting FAULt means that pin 1 functions as an isolated fault output. The fault signal is true when any output is in a protected state (from OC, OV, OT, INH). Note also that Pin 2 serves as the isolated common for pin 1. When pin 1 is set to the FAULt function, the instrument ignores any commands to program pin 2. Queries of pin 2 will return FAULt. If pin 1 is changed from FAULt to another function, pin 2 is set to DINPut.
INHhibit	Applies only to pin 3. When pin 3 is configured as an inhibit input, a true signal at the pin will disable all input channels.
ONCouple	When configured as an On control, the pin will synchronize the output On state between electronic loads. 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 electronic loads. Only one pin can be configured as an Off control. The pin will function as both an input and an output. The polarity of the pin is fixed and cannot be programmed.

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



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

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

The query returns the polarity, POS or NEG.

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

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

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

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

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

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

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

## DISPlay Subsystem

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, inputs are not sent to the display and all annunciators except ERROR are disabled. The display state is automatically turned on when you return to the local mode. Press **Lock/Unlock** key to return to the local mode.

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

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

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

DISPlay[:WINDow]:TEXT[:DATA]?

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

The query returns the quoted string.

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

DISPlay[:WINDow]:TEXT:CLEAr

The command clears the message displayed on the front panel.

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

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

The command selects 1-channel or 2-channel meter view for EL34243A models. METER1 displays one large input channel with one small less detailed channel. METER2 displays both input channels.

The query returns METER1 or METER2.

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

## FETCh Subsystem

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

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

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

FETCh[:SCALar]:POWER[:DC]? [CH1 | CH2] [(@<chanlist>)]

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

Parameter	Typical Return
(none)	<DC value>
Returns the averaged current	FETC:CURR? (@1)
Returns the averaged voltage	FETC:VOLT? (@1)
Returns the averaged power	FETC:POW? (@1)

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

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

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

Parameter	Typical Return
(none)	<ACDC value>
Returns the measured RMS current	FETC:CURR:ACDC? (@1)
Returns the measured RMS voltage	FETC:VOLT:ACDC? (@1)

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

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

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

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

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

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

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

Parameter	Typical Return
(none)	<MIN value> <MAX value>
Returns the measured minimum current	FETC:CURR:MIN? (@1)
Returns the measured minimum voltage	FETC:VOLT:MIN? (@1)
Returns the measured minimum power	FETC:POW:MIN? (@1)
Returns the measured maximum current	FETC:CURR:MAX? (@1)
Returns the measured maximum voltage	FETC:VOLT:MAX? (@1)
Returns the measured maximum power	FETC:POW:MAX? (@1)

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

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

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

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

Parameter	Typical Return
(none)	<value> [,<value>] or <Block>
Returns the measured current array	FETC:ARR:CURR? (@1)
Returns the measured voltage array	FETC:ARR:VOLT? (@1)
Returns the measured power array	FETC:ARR:POW? (@1)

### Remarks

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

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

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

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

## FUNcTion | MODE Subsystem

These commands set the input priority mode. Function and Mode command are equivalent.

```
[SOURce:]FUNcTion CURRent | VOLTage | POWer | RESistance[, (@<chanlist>)]
```

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

```
[SOURce:]MODE CURRent | VOLTage | POWer | RESistance[, (@<chanlist>)]
```

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

The command sets the input regulation mode.

Mode	Description
CURRent	Constant current (CC) mode
VOLTage	Constant voltage (CV) mode
POWer	Constant power (CP) mode
RESistance	Constant resistance (CR) mode

In CV mode, the input is controlled by a bi-polar constant voltage feedback loop, which maintains the input voltage at its positive or negative programmed setting.

In CC mode, the input is controlled by a bi-polar constant current feedback loop, which maintains the input sourcing or sinking current at its programmed setting.

In CP and CR modes, the constant current along with the constant voltage feedback loops maintain the input power and resistance at their programmed settings.

Parameter	Typical return
CURR   VOLT   POW   RES	CURR, VOLT, POW, or RES
*RST CURR	
Selects the constant current mode: FUNC CURR, (@1)	

## IEEE-488.2 Common Commands

### \*CLS

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

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

### \*ESE <enable value>

### \*ESE?

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

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

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

### Standard event status enable register

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



## \*ESR?

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

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

## \*IDN?

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

Keysight Technologies,EL34243A,MY00000001,X.X.X-X.X.X-X-X

The four comma-separated fields are the manufacturer's name, the model number, the serial number, and the revision code. The first "X.X.X" in the revision codes is the firmware revision number for the controller firmware; the second is for the front panel firmware; the third is for the primary FPGA firmware; and the fourth is for the secondary FPGA 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 electronic load 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, resistance, power, range, slew, mode, sense, short and current limit
- OCP delay, OCP state, OCP delay start, OPP state, OPP delay and UVI
- Input state and selected input
- Operation mode (Independent or Parallel) (applicable for EL34243A only)
- Input on/off sequencing
- Sequencer settings
- Trigger settings
- Digital I/O input data and bus setting (applicable for EL34143A and EL34243A)
- Scope and 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. Use **SYSTem:ERRor?** to read the error queue. See **SCPI Error Messages** for more information.

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

## \*WAI

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

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

## INITiate Subsystem

### INITiate[:IMMEDIATE]:ACQUIRE [(@<chanlist>)]

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

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

#### Remarks

- It takes a few milliseconds for the electronic load to be ready to receive a trigger signal after receiving the INITiate:ACQUIRE command, and it can take longer if the value of SENSE:SWEEP:OFFSET is negative.
- Use the appropriate **ABORT** command to return the instrument to the idle state.

### INITiate[:IMMEDIATE]:DLOG <"filename">

The command initiates the 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 or Internal:/log2.dlog. Any other filename is not accepted for the internal data logging session.

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

### INITiate[:IMMEDIATE]:TRANSient [(@<chanlist>)]

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

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

INITiate:CONTInuous:TRANSient ON | OFF | 1 | 0[, (@<chanlist>)]

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

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

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

Parameter	Typical return
ON   OFF   1   0	1 or 0
*RST OFF	
Initiates the trigger system on channel 1 continuously: INIT:CONT:TRAN ON, (@1)	

### Remark

- **ABORT:TRANSient** does not abort continuous triggers if INITiate:CONTInuous:TRANSient ON has been programmed. In this case, the trigger system will automatically re-initiate.

## INPut | OUTPut Subsystem

These commands control the input, power-on and protection clear function. The INPut and OUTPut commands are equivalent.

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

INPut[:STATe]? [(@<chanlist>)]

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

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

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

The query returns the input state of the electronic load. The returned value is “0” (OFF) or “1” (ON).

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Turns the input off: INP OFF, (@1)	

INPut[:STATe]:COUPle:CHANNeL ALL | NONE | <list>

INPut[:STATe]:COUPle:CHANNeL?

OUTPut[:STATe]:COUPle:CHANNeL ALL | NONE | <list>

OUTPut[:STATe]:COUPle:CHANNeL?

The command specifies which input channels are controlled by the input synchronization function. The input 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 input channels go to the input OFF state. This parameter is non-volatile and not affected by **\*RST**.

Parameter	Typical return
ALL   NONE   <list>	<coupled channels>
<list> = CH1   CH2	
*RST NONE	
Couples channels 1 and 2: INP:COUP:CHAN CH1, CH2	

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

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

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

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

This command sets the delay in seconds that the instrument waits before disabling the specified input. This allows multiple input channels to turn off in sequence. Each input will not turn off until its delay time has elapsed. This command effects on-to-off transitions including changes in the OUTPut[:STATe]. It does NOT affect transitions to off caused by protection functions. Delay times can be programmed from 0 to 1023 seconds with the resolution of one millisecond.

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

Parameter	Typical Return
0 - 1023   MIN   MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s before turning off the input of Channel 2: <code>OUTP:DEL:FALL 0.5, (@2)</code>	



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

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

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

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

This command sets the delay in seconds that the instrument waits before enabling the specified input. This allows multiple input channels to turn on in sequence. Each input 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 1023 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 - 1023   MIN   MAX	<delay value>
*RST 0	
Sets a delay of 0.5 s before turning on the input of Channel 2: <code>OUTP:DEL:RISE 0.5, (@2)</code>	

INPut:INHibit:MODE LATChing | LIVE | OFF

INPut:INHibit:MODE?

OUTPut:INHibit:MODE LATChing | LIVE | OFF

OUTPut:INHibit:MODE?

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

If an input channel has been turned off by INPut[:STATe], the inhibit function does not affect the load input 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 load inputs, 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 load inputs follow the state of the Inhibit input. Load inputs are disabled if Inhibit is true and enabled if Inhibit is false.
OFF	The Inhibit input is ignored.

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

INPut:PAIR OFF | PARAllel

INPut:PAIR?

OUTPut:PAIR OFF | PARAllel

OUTPut:PAIR?

**NOTE** This command is applicable for EL34243A only.

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

The query returns the electronic load operation mode.

Parameter	Typical Return
OFF   PAR	OFF or PAR
*RST OFF	
Specifies the electronic load operation mode to parallel: INP:PAIR PAR	

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

This clears the latch that disables the input when a protection condition occurs. The input is restored to the state it was in before the protection condition occurred.

Parameter	Typical return
(none)	(none)
Clears the latched protection status: INP:PRO:CLE (@1)	

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

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

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

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

INPut:SHORt[:STATe] ON | OFF | 1 | 0

INPut:SHORt[:STATe]?

OUTPut:SHORt[:STATe] ON | OFF | 1 | 0

OUTPut:SHORt[:STATe]?

The command simulates a short circuit on the input of the electronic load. It works in all operating modes and temporarily overrides input and slew settings. The actual value of the short depends on the mode and range that are active when the short is programmed.

In CV mode, a short is equivalent to programming zero volts.

In CC mode, a short is equivalent to programming full-scale current for the present current range.

In CR mode, a short is equivalent to programming the minimum resistance for the present resistance range.

Turning the short circuit on does not affect the programmed settings, and the electronic load input will return to the previously programmed values when the short is turned off.

The query returns the input short state of the electronic load. The returned value is “0” (OFF) or “1” (ON).

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Simulate a short on the electronic load input terminals: INP:SHOR ON	

## LIST Subsystem

List commands program an input sequence of multiple current, voltage, power, or resistance settings. A comma-delimited list of up to 100 steps may be programmed.

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

```
[SOURce:]LIST:COUNT? [MINimum | MAXimum | INFINITY,] [(@<chanlist>)]
```

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

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

**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 input returns to the settings that were in effect before the list started.

---

Parameter	Typical return
1 - 9999   MIN   MAX   INF	<list count>
*RST 1	
Sets the list count for channel 1 to 10: LIST:COUNT 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, Power, Resistance, 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
minimum - maximum (Values are range and product-model dependent. See <b>Programming Range</b> )	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step set to the minimum programmable value.	
Programs a current list containing 5 steps 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 input will remain at a specific step. Dwell times can be programmed from 0 through 268,435 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 input 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, Power, Resistance, 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 - 268,435	<list value 1>, <list value 2>, <list value 3>...
*RST 0.001	
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:DWELL:POIN? (@1)	

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

[SOURce:]LIST:POWer[:LEVel] ]? [(@<chanlist>)]

The command specifies the power setting for each list step in watts. A comma-delimited list of up to 100 steps may be programmed.

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

**NOTE** The order in which the power values are entered determines the sequence when the list executes. To create a valid list, the Voltage, Current, Power, Resistance, 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 power list; it does not append to the previous list.

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

[SOURce:]LIST:POWer:POINTs? [(@<chanlist>)]

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

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

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

The command specifies the resistance setting for each list step in ohms. A comma-delimited list of up to 100 steps may be programmed.

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

**NOTE** The order in which the resistance values are entered determines the sequence when the list executes. To create a valid list, the Voltage, Current, Power, Resistance, 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 resistance list; it does not append to the previous list.

---

Parameter	Typical return
minimum - maximum (Values are range and product-model dependent. See <b>Programming Range</b> )	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step set to the minimum programmable value.	
Programs a resistance list containing 3 steps for channel 1: LIST:RES 20,50,100 (@1)	

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

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

Parameter	Typical return
(none)	<points>
Returns the number of points in the resistance list for channel 1: LIST:RES: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 input 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 input. This specifies a dwell-paced list.
ONCE	The input 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>)]

The command specifies the input value when the list terminates. The state is either 1 (ON) or 0 (OFF). When ON, the input 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 input returns to the settings that were in effect before the list started. The query returns 0 (OFF) or 1 (ON).

Parameter	Typical return
ON   OFF   1   0	0 or 1
*RST OFF	
Sets the list to terminate input at the last step value: LIST:TERM LAST ON, (@1)	

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

The command specifies which list steps generate a trigger signal at the beginning of the step (BOSTep). A comma-delimited list of up to 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 inputs.

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, Power, Resistance, 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
ON   OFF   1   0	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step with a value of OFF	
Specifies that triggers will be generated at the beginning of the second step of a 5-step list for channel 1: LIST:TOUT:BOST 1,1,1,1,1, (@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. 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 measurements and transients of other units, and for digital port pins configured as trigger inputs.

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

The command specifies which list steps generate a trigger signal at the end of the step (EOSTep). A comma-delimited list of up to 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 inputs.

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, Power, Resistance, 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
ON   OFF   1   0	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step with a value of OFF	
Specifies that triggers will be generated at the end of the second step of a 5-step list for channel 1: LIST:TOUT:EOST 1,1,1,1,1, (@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, Power, Resistance, 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
minimum - maximum (Values are range and product-model dependent. See <b>Programming Range</b> )	<list value 1>, <list value 2>, <list value 3>...
*RST 1 step set to the minimum programmable value.	
Programs a voltage list containing 3 steps for channel 1: LIST:VOLT 20,10,5, (@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 Subsystem

LXI:IDENTify[:STATe] ON | OFF | 1 | 0

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

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

The query returns the state of the LXI status indicator.

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

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

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

The command allows you to control the mDNS service.

The query returns the control of the mDNS service.

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

## MEASure Subsystem

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

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

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

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

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

The query returns the averaged input measurement. Values are either in amperes, volts, or watts. The reading is in the form +n.nnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Parameter	Typical Return
(none)	<DC value>
Returns the averaged current: MEAS:CURR? (@1)	
Returns the averaged power: MEAS:POW? (@1)	
Returns the averaged voltage: MEAS:VOLT? (@1)	

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

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

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

Parameter	Typical Return
(none)	<ACDC value>
Returns the measured RMS current: MEAS:CURR:ACDC? (@1)	
Returns the measured RMS voltage: MEAS:VOLT:ACDC? (@1)	

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

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

Parameter	Typical Return
(none)	<MIN value> <MAX value>
Returns the measured minimum current: MEAS:CURR:MIN? (@1)	
Returns the measured minimum voltage: MEAS:VOLT:MIN? (@1)	
Returns the measured minimum power: MEAS:POW:MIN? (@1)	
Returns the measured maximum current: MEAS:CURR:MAX? (@1)	
Returns the measured maximum voltage: MEAS:VOLT:MAX? (@1)	
Returns the measured maximum power: MEAS:POW:MAX? (@1)	

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

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

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

Parameter	Typical Return
(none)	<value> [, <value>] or <Block>
Returns the measured current array MEAS:ARR:CURR? (@1)	
Returns the measured power array: MEAS:ARR:POW? (@1)	
Returns the measured voltage array: MEAS:ARR:VOLT? (@1)	

## MMEMory Subsystem

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"	



## POWer Subsystem

POWer commands program the input power and power protection function. The SOURce keyword is optional in the following commands.

```
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <power> | MINimum | MAXimum | DEFault[, (@<chanlist>)]
```

```
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]? [MINimum | MAXimum | DEFault,] [(@<chanlist>)]
```

The command sets the immediate power level when the input is in constant power mode. Units are in watts.

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

Parameter	Typical return
minimum - maximum   MIN   MAX   DEF (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<power level>
*RST MIN	
Sets the input power level to 50 watts: POW 50, (@1)	

```
[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude] <power> | MINimum | MAXimum[, (@<chanlist>)]
```

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

The command sets the triggered power level when the input is in constant power mode. The triggered level is a stored value that is transferred to the input when an input step is triggered. Units are in watts.

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

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<power level>
*RST MIN	
Sets the triggered power level to 75 watts: POW:TRIG 75, (@1)	

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

The command sets the transient mode. This determines what happens to the power when the transient system is initiated and triggered.

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

Parameter	Typical return
FIXed   STEP   LIST	FIXed, STEP, or LIST
*RST FIXed	
Sets the input power mode to Step: POW:MODE STEP, (@1)	

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

**NOTE** This command is the same as OUTPut:PROTection:DELAy.

The command sets the over-power protection delay. The over-power protection function will not be triggered during the delay time. This prevents momentary changes in input status from triggering the over-power protection function. If the input is regulating in power limit longer than the programmable delay time, the OPP will trip, and the input will be turned off. Programmed values can range from 0 to 0.255 seconds with a resolution of 1 millisecond.

Parameter	Typical Return
0 - 0.255   MIN   MAX	<delay value>
*RST 0.020	
Sets the protection delay to 0.2 seconds: POW:PROT:DEL 0.2, (@1)	

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

The command enables/disables the over-power protection. When the input power exceeds the power rating of the electronic load by 10%, the input is disabled and the Questionable Condition status register CP+ bit is set.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Enables the power protection state: POW:PROT:STAT ON, (@1)	

[SOURce:]POWer:RANGe <power> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]POWer:RANGe? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the power range. The value that you enter must be the highest value in watts that you expect to input. The instrument selects the range with the best resolution for the value entered. The following ranges are available for the electronic load.

	EL33133A	EL34143A	EL34243A
High range	from 1.5 W to 255 W	from 2 W to 357 W	from 2 W to 306 W
Medium range	from 0.15 to 25.5 W	from 0.3 W to 35.7 W	from 0.3 W to 30.6 W
Low range	from 0.02 W to 5.1 W	from 0.01 W to 8.16 W	from 0.01 W to 7.14 W

The query returns the maximum power that can be programmed on the range that is presently set. The power is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Parameter	Typical Return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<power range>
*RST MAX	
Sets the medium power range (from 0.2 W to 30.6 W): POW:RANG 30, (@1)	

## Remarks

- This command is coupled with [SOURce:]POWer. This means that if a range command places an input on a range with a lower maximum power than the present power level, an out of range error occurs. This also occurs if a power level is programmed that is too large for the present range. Errors can be avoided by sending both level and range commands in the same SCPI message. When the range and level information is received as a set, no range/setting conflict occurs.
- If a programmed range change occurs while the input is enabled, the input will be temporarily disabled while the range switch occurs. The transition from on-to-off and then from off-to-on will also be delayed by the settings of [INPut:DElay:FALL](#) and [INPut:DElay:RISE](#).

[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value> | MINimum | MAXimum | INFinity[, (@<chanlist>)]

[SOURce:]POWer:SLEW[:POSitive][:IMMediate]? [MINimum | MAXimum,] [(@<chanlist>)]

[SOURce:]POWer:SLEW:NEGative[:IMMediate] <value> | MINimum | MAXimum | INFinity[, (@<chanlist>)]

[SOURce:]POWer:SLEW:NEGative[:IMMediate]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the input power slew rate. The slew rate is set in watts per second and affects all programmed power changes, including those due to the input state turning on or off. The slew rate can be set to any value between 0 and 9.9E+37. For very large values, the slew rate will be limited by the analog performance of the input circuit. The keywords MAX or INFinity set the slew rate to maximum.

Parameter	Typical Return
minimum – 9.9E+37   MIN   MAX   INFinity	<slew rate>
*RST MAX	
Sets the slew rate to 10 watts per second: POW:SLEW 10, (@1)	
Sets the negative slew rate to 10 watts per second: POW:SLEW:NEG 10, (@1)	

## Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using POWer:SLEW? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]POWer:SLEW:COUPlE ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]POWer:SLEW:COUPlE? [(@<chanlist>)]

The command sets the input power slew tracking state.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Enables the input power slew tracking state: RES:SLEW COUP ON, (@1)	

```
[SOURce:]POWer:SLEW[:POSitive]:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]POWer:SLEW[:POSitive]:MAXimum? [(@<chanlist>)]
[SOURce:]POWer:SLEW:NEGative:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]
[SOURce:]POWer:SLEW:NEGative:MAXimum? [(@<chanlist>)]
```

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by POWER:SLEW. Use POWER:SLEW? MAX to query the maximum slew rate.

Parameter	Typical Return
ON   OFF   1   0,	0 or 1
*RST ON	
Enables the maximum slew rate override: POW:SLEW:MAX ON, (@1)	
Enables the negative maximum slew rate: POW:SLEW:NEG:MAX ON, (@1)	

### Remarks

- The POWER:SLEW:MAX command is coupled to the POWER:SLEW command. If the POWER:SLEW command sets the slew rate to MAX or INFINITY, POWER:SLEW:MAX is enabled. If the slew rate is set to any other value, POWER:SLEW:MAX is disabled.

```
[SOURce:]POWer:TLEVel <value> | MINimum | MAXimum[, (@<chanlist>)]
[SOURce:]POWer:TLEVel? [MINimum | MAXimum,] [(@<chanlist>)]
```

The command specifies the transient level of the input power. The transient function switches between the immediate setting and the transient level. Units are in watts.

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <b>Programming Range</b> )	<transient level>
*RST MIN	
Sets the transient input power to 5 W: POW:TLEV 5, (@1)	

## RESistance Subsystem

RESistance commands program the input resistance. The SOURce keyword is optional in the following commands.

```
[SOURce:]RESistance[:LEVel][:IMMEDIATE][:AMPLitude] <resistance> | MINimum | MAXimum | DEFault  
[, (@<chanlist>)]
```

```
[SOURce:]RESistance[:LEVel][:IMMEDIATE][:AMPLitude]? [MINimum | MAXimum | DEFault,] [(@<chan-  
list>)]
```

The command sets the immediate resistance level when the input is in constant resistance mode. Units are in ohms.

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

Parameter	Typical return
minimum - maximum   MIN   MAX   DEF (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<resistance level>
*RST MAX	
Sets the input resistance level to 0.5 ohms: RES 0.5, (@1)	

```
[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <resistance> | MINimum | MAXimum[, (@<chan-  
list>)]
```

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

The command sets the triggered resistance level when the input is in constant resistance mode. The triggered level is a stored value that will be programmed when a Step transient is triggered. Units are in ohms.

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

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<resistance level>
*RST MAX	
Sets the triggered resistance level to 1 ohms: RES:TRIG 1, (@1)	

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

The command sets the transient mode. This determines what happens to the resistance when the transient system is initiated and triggered.

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

Parameter	Typical return
FIXed   STEP   LIST	FIXed, STEP, or LIST
*RST FIXed	
Sets the resistance mode to Step: RES:MODE STEP, (@1)	

[SOURce:]RESistance:RANGe <resistance> | MINimum | MAXimum[, (@<chanlist>)]  
 [SOURce:]RESistance:RANGe? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the resistance range. The value that you enter must be the highest value in ohms that you expect to input. The instrument selects the range with the best resolution for the value entered. The following overlapping resistance ranges are available for the electronic load:

	EL33133A	EL34143A	EL34243A
High range	from 100 Ω to 4 kΩ	from 100 Ω to 4 kΩ	
Medium range	from 10 Ω to 1.25 kΩ	from 10 Ω to 1.25 kΩ	
Low range	from 0.08 Ω to 30 Ω	from 0.05 Ω to 30 Ω	

The query returns the maximum resistance that can be programmed on the range that is presently set. The resistance is in the form +n.nnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Parameter	Typical Return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <b>Programming Range</b> )	<resistance range>
*RST MAX	
Sets the medium resistance range (from 10 to 1.25 kΩ): RES:RANG 100, (@1)	

## Remarks

- This command is coupled with [SOURce:]RESistance. This means that if a resistance value is programmed that does not fall within the presently selected range, the range command automatically selects the correct range for the new resistance value. An out of range error occurs if a resistance value is less than the low-range minimum, or greater than the high-range maximum.
- The resistance level value always takes precedence over the resistance range value. Therefore, an out of range error occurs if a resistance range is changed when the presently selected resistance value does not fall within the new range. Errors can be avoided by sending both level and range commands in the same SCPI message. When the range and level information is received as a set, no range/setting conflict occurs.
- If a programmed range change occurs while the input is enabled, the input will be temporarily disabled while the range switch occurs. The transition from on-to-off and then from off-to-on will also be delayed by the settings of **INPut:DELaY:FALL** and **INPut:DELaY:RISE**.



[SOURce:]RESistance:SLEW[:POSitive][:IMMEDIATE] <value> | MINimum | MAXimum | INFInity[, (@<chanlist>)]

[SOURce:]RESistance:SLEW[:POSitive][:IMMEDIATE]? [MINimum | MAXimum,] [(@<chanlist>)]

[SOURce:]RESistance:SLEW:NEGative[:IMMEDIATE] <value> | MINimum | MAXimum | INFInity[, (@<chanlist>)]

[SOURce:]RESistance:SLEW:NEGative[:IMMEDIATE]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the resistance slew rate. The slew rate is set in ohms per second and affects all programmed resistance changes, including those due to the input state turning on or off. The slew rate can be set to any value between the minimum value and 9.9E+37. For very large values, the slew rate will be limited by the analog performance of the input circuit. The keywords MAX or INFInity set the slew rate to maximum.

Parameter	Typical Return
minimum - 9.9E+37   MIN   MAX   INFInity	<slew rate>
*RST MAX	
Sets the slew rate to 50 ohms per second: RES:SLEW 50, (@1)	
Sets the negative slew rate to 25 ohms per second: RES:SLEW:NEG 25, (@1)	

## Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using RESistance:SLEW? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]RESistance:SLEW:COUPlE ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]RESistance:SLEW:COUPlE? [(@<chanlist>)]

The command sets the resistance slew tracking state.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST OFF	
Enables the resistance slew tracking state: RES:SLEW COUP ON, (@1)	

[SOURce:]RESistance:SLEW[:POSitive]:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]RESistance:SLEW[:POSitive]:MAXimum? [(@<chanlist>)]

[SOURce:]RESistance:SLEW:NEGative:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]RESistance:SLEW:NEGative:MAXimum? [(@<chanlist>)]

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by RESistance:SLEW. Use RESistance:SLEW? MAX to query the maximum slew rate.

Parameter	Typical Return
ON   OFF   1   0, *RST ON	0 or 1
Disables the maximum slew rate override: RES:SLEW:MAX OFF, (@1)	
Disables the negative maximum slew rate: RES:SLEW:NEG:MAX OFF, (@1)	

### Remarks

- The RESistance:SLEW:MAX command is coupled to the RESistance:SLEW command. If the RESistance:SLEW command sets the slew rate to MAX or INFINITY, RESistance:SLEW:MAX is enabled. If the slew rate is set to any other value, RESistance:SLEW:MAX is disabled.

[SOURce:]RESistance:TLEVel <value> | MINimum | MAXimum[, (@<chanlist>)]

[SOURce:]RESistance:TLEVel? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the transient level of the resistance. The transient function switches between the immediate setting and the transient level. Units are in ohms.

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<transient level>
*RST MAX	
Sets the transient input resistance to 5 Ω: RES:TLEV 5, (@1)	

## SENSe Subsystem

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

SENSe:DLOG:FUNction:CURRent? [(@<chanlist>)]

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)
*RST OFF	
Enables current data logging of channel 1 and 2: SENS:DLOG:FUNC:CURR 1, (@1,2)	

SENSe:DLOG:FUNction:MINMax ON | OFF | 1 | 0

SENSe:DLOG:FUNction:MINMax?

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

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

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

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

SENSe:DLOG:FUNction:VOLTage? [(@<chanlist>)]

The command enables or disables voltage data logging 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)
*RST ON	
Enables voltage data logging of channel 1 to 2: SENS:DLOG:FUNC:VOLT 1, (@1:2)	

SENSe:DLOG:OFFSet <offset percent>

SENSe:DLOG:OFFSet?

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

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

The query returns the trigger offset in percent.

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

SENSe:DLOG:PERiod <time> | MINimum | MAXimum

SENSe:DLOG:PERiod? [MINimum | MAXimum]

The command sets the sample period, the entered value is in seconds. For sample period, minimum is 20.48  $\mu$ s, and it has to be in integral of 20.48  $\mu$ s, maximum is 60 seconds. The entered value will be rounded to the nearest 20.48  $\mu$ s integral.

The query returns the sample period in seconds.

Parameter	Typical return
0.00002048 - 60   MIN   MAX	<time>
*RST 0.2	
Sets the sample period to 400 ms: <code>SENS:DLOG:PER 0.4</code>	

**NOTE**

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

SENSe:DLOG:TIME <time> | MINimum | MAXimum

SENSe:DLOG:TIME? [MINimum | MAXimum]

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

The query returns the sample duration in seconds.

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

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

**NOTE**

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

This command sets the sample period, the entered value is in seconds. For sample period, minimum is 24.08  $\mu$ s, and it has to be in integral of 20.48  $\mu$ s, maximum is 60 seconds. The entered value will be rounded to the nearest 20.48  $\mu$ s integral.

The query returns the sample period in seconds.

Parameter	Typical return
0.00002048 - 60   MIN   MAX	<time>
*RST 0.2	
Sets the sample period to 400 ms: SENSe:DLOG:TINT 0.4	

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

SENSe:FUNCTion:CURRent? [(@<chanlist>)]

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

SENSe:FUNCTion:VOLTage? [(@<chanlist>)]

The command enables or disables the current or voltage measurement function.

The query returns the status (on or off) of the current or voltage measurement function.

Parameter	Typical return
ON   OFF   1   0	0 or 1
*RST ON	
Enables current measurement on channel 1: SENSe:FUNCT:CURR 1, (@1)	
Enables voltage measurement on channel 1: SENSe:FUNCT:VOLT 1, (@1)	

SENSe:SWEEp:POINts <data point> | MINimum | MAXimum[, (@<chanlist>)]

SENSe:SWEEp:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the number of points in a measurement. The number of points can be specified from 1 to 131,072. This command applies to both voltage and current measurement.

The number of samples (points) that can be specified depends on the number of measurement parameter selected. You can measure up to four parameters (voltage and current x 2 inputs).

1 parameters : up to 128K points

2 parameters: up to 64K points

3 parameters: up to 32K points

4 parameters; up to 16K points (K = 1024)

Parameter	Typical return
1 - 131,072   MIN   MAX	<data point>
*RST 2440	
Specifies 2048 points: SENSe:SWEEp:POIN 2048, (@1)	

SENSe:SWEEp:OFFSet:POINt <offset points> | MINimum | MAXimum[, (@<chanlist>)]

SENSe:SWEEp:OFFSet:POINts? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the offset in a data sweep for triggered measurements. Positive values represent the delay after the trigger occurs but before the samples are acquired. Negative values represent data samples taken prior to the trigger.

Parameter	Typical return
-131,071 - 2,000,000,000   MIN   MAX	<offset points>
*RST 0	
Specifies -2048 offset points: SENSe:SWEEp:OFFS:POIN -2048, (@1)	

SENSe:SWEep:TINterval <time> | MINimum | MAXimum[, (@<chanlist>)]

SENSe:SWEep:TINterval? [MINimum | MAXimum,] [(@<chanlist>)]

The command defines the time period between samples in seconds. Programmed values can range from 5.12  $\mu$ s to 40,000 s. Note that the shortest time interval (fastest speed) that can be specified depends on the number of parameters that are being measured and the model that is doing the measuring. Values above 10.24 ms are rounded to the nearest 10.24 ms increment. Values above 20.48 ms are rounded to the nearest 20.48 ms increment.

5.12 microseconds: 1 parameter

10.24 microseconds: 1 or 2 parameters

20.48 microseconds: 3 or 4 parameters

Parameter	Typical return
0.00000512 - 40,000   MIN   MAX	<time interval>
*RST 0.00004096	
Specifies an interval of 1 ms between points: SENS:SWE:TINT 0.001, (@1)	

## SOURce Subsystem

The Source subsystem programs the current, digital, list, power, resistance, transient, and voltage functions.

### Subsystems Using the Optional SOURce Keyword

Because SOURce subsystem commands are often used without the SOURce keyword, these commands are listed by their individual subsystems, below:

**CURRent**

**DIGital**

**FUNcTion | MODE**

**LIST**

**POWer**

**RESistance**

**TRANsient**

**VOLTage**



## STATus Subsystem

Status commands let you determine the operating condition of the instrument at any time. The instrument has three groups of status registers; Operation, Questionable, and Standard Event. The Operation and Questionable status groups each consist of the Condition, Enable, and Event registers as well as NTR and PTR filters. Refer to [SCPI Status Registers](#) for more information

Instrument status is also programmed using the IEEE 488.2 Common commands: \*CLS, \*ESE, \*ESR?, \*OPC, \*OPC?, \*SRE, \*STB? and \*WAI discussed under [IEEE 488.2 Common Commands](#).

### STATus:OPERation[:EVENT]?

The query returns the value of the **event register** for the **Operation Status** group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and PTR filter. Reading the Operation Status Event register clears it.

Parameter	Typical return
(none)	<bit value>
Reads the operation status event register: STAT:OPER? (@1)	

#### Remarks

- \*RST has no effect on this register.
- The value returned is the binary-weighted sum of all bits set in the register.

### STATus:OPERation:CONDition?

The query returns the value of the **condition register** for the **Operation Status** group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Operation Status Condition register does not clear it.

Parameter	Typical return
(none)	<bit value>
Reads the operation status condition register: STAT:OPER:COND? (@1)	

#### Remarks

- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:OPERation:ENABLE <enable value>

STATus:OPERation:ENABLE?

The command sets bits in the **enable register** for the **Operation Status** group. The enable register is a mask for enabling specific bits from the Operation Event register to set the OPER (operation summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register.

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

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 9 in the enable register: STAT:OPER:ENAB 512, (@1)	

### Remarks

- \*CLS does not clear the enable register, but does clear the **event register**.

STATus:OPERation:NTRansition <value>[, (<@chanlist>)]

STATus:OPERation:NTRansition? [(<@chanlist>)]

The command sets and queries the value of the **NTR** (Negative-Transition) register.

This register serves as a polarity filter between the Operation Condition and Operation Event registers.

When a bit in the NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the NTR register: STAT:OPER:NTR 24, (@1)	

### Remarks

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation Condition register sets the corresponding bit in the Operation Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation Condition register can set the corresponding bit in the Operation Event register .
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:OPERation:PTRansition <value>[, (<@chanlist>)]  
 STATus:OPERation:PTRansition? [((<@chanlist>))]

The command sets and queries the value of the **PTR** (Positive-Transition) register.

This registers serves as a polarity filter between the Operation Condition and Operation Event registers.

When a bit in the PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set.

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the PTR register::	STAT:OPER:PTR 24, (@1)

### Remarks

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation Condition register sets the corresponding bit in the Operation Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation Condition register can set the corresponding bit in the Operation Event register .
- The value returned is the binary-weighted sum of all bits set in the register.

### STATus:PRESet

The command presets the OPERation and QUESTionable ENABle, PTRansition and NTRansition registers.

Operation register	Questionable register	Preset setting
STAT:OPER:ENAB	STAT:QUES:ENAB	all defined bits are disabled
STAT:OPER:NTR	STAT:QUES:NTR	all defined bits are disabled
STAT:OPER:PTR	STAT:QUES:PTR	all defined bits are disabled

Parameter	Typical return
(none)	(none)
Presets the Operation and Questionable registers:	STAT:PRES

STATus:QUESTIONable[:EVENT]? [(<@chanlist>)]

The query returns the value of the **event register** for the Questionable Status group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and PTR filter. Reading the Questionable Status Event register clears it.

Parameter	Typical return
(none)	<bit value>
Reads the questionable status event register: STAT:QUES? (@1)	

#### Remarks

- \*RST has no effect on this register.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:QUESTIONable:CONDition? [(<@chanlist>)]

The query returns the value of the **condition register** for the **Questionable Status** group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Questionable Status Condition register does not clear it.

Parameter	Typical return
(none)	<bit value>
Reads the condition register: STAT:QUES:COND? (@1)	

#### Remarks

- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- \*RST clears this register, other than those bits where the condition still exists after \*RST.
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:QUESTionable:ENABle <value>[, (<@chanlist>)]  
 STATus:QUESTionable:ENABle? [((<@chanlist>)]

The command sets bits in the **enable register** for the **Questionable Status** group. The enable register is a mask for enabling specific bits from the Operation Event register to set the QUES (questionable summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register.

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

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
Enable bit 4 in the questionable enable register: STAT:QUES:ENAB 16, (@1)	

STATus:QUESTionable:NTRansition <value>[, (<@chanlist>)]  
 STATus:QUESTionable:NTRansition? [((<@chanlist>)]

The command sets and queries the value of the **NTR** (Negative-Transition) register.

This register serves as a polarity filter between the Questionable Condition and Questionable Event registers.

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

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the NTR register: STAT:QUES:NTR 24, (@1)	

## Remarks

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register .
- The value returned is the binary-weighted sum of all bits set in the register.

STATus:QUESTionable:PTRansition <value>[, (<@chanlist>)]

STATus:QUESTionable:PTRansition? [(<@chanlist>)]

The command sets and queries the value of the **PTR** (Positive-Transition) register.

This registers serves as a polarity filter between the Questionable Condition and Questionable Event registers.

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

STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Parameter	Typical return
A decimal value that corresponds to the binary-weighted sum of the bits in the register.	<bit value>
*RST 0	
Enable bit 3 and 4 in the PTR register:: STAT:QUES:PTR 24, (@1)	

### Remarks

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register .
- The value returned is the binary-weighted sum of all bits set in the register.

## SYSTem Subsystem

### SYSTem:BEEPer[:IMMediate]

The command issues a single beep immediately.

Parameter	Typical return
(none)	(none)
Issues a single beep immediately: SYST:BEEP	

### SYSTem:BEEPer:STATe ON | OFF | 1 | 0

### SYSTem:BEEPer:STATe?

The command enables or disables the beeper.

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

Parameter	Typical return
ON   OFF   1   0	0 or 1
*RST ON	
Turns on the beeper: SYST:BEEP:STAT ON	

### SYSTem:COMMunicate:RLSTate LOCal | REMote | RWLock

### SYSTem:COMMunicate:RLSTate?

The command sets the electronic load to remote or local mode. The LOCal parameter is the same as SYSTem:LOCAl, the REMote parameter is the same as SYSTem:REMote, and the RWLock parameter is the same as SYSTemRWLock.

The query returns LOC, REM, or RWL.

Parameter	Typical return
LOC   REM   RWL	LOC, REM, or RWL
*RST LOC	
Sets the electronic load 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 supported)
Queries the control connection port number: SYST:COMM:TCP:CONT?	

**NOTE** The control socket connection can only be used by a client to send a device clear to the instrument or to detect Service Request (SRQ) events.

Refer to "Using Sockets" in the *User's Guide* for more information.

## SYSTem:DATE <yyyy>,<mm>,<dd>

### SYSTem:DATE?

The command sets the date of the real time clock in year (yyyy), month (mm), and day (dd). The range of values for the year is from 2000 to 2099.

The query returns comma-separated values that correspond to the year, month, and day. For example:  
+2020,+1,+26.

Parameter	Typical return
<yyyy>,<mm>,<dd>	<+yyyy,+mm,+dd>
Sets the date to November 27, 2019: SYST:DATE 2019,11,27	



## SYSTem:ERRor[:NEXT]?

The query returns the electronic load error queue of up to 20 errors. The electronic load 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 electronic load 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 electronic load in the local mode. All front-panel keys are fully functional.

Parameter	Typical return
(none)	(none)
Sets the electronic load in the local mode: SYST:LOC	

## SYSTem:REMOte

The command places the electronic load 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 electronic load 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 electronic load: SYST:SEC:IMM	

## 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 electronic load. 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?	

## TRANSient Subsystem

TRANSient commands program the transient generator of the electronic load. The transient generator programs a second (transient) level at which the electronic load can operate without changing the original programmed settings.

See also [\[SOURce:\]CURRENT:TLEVel](#), [\[SOURce:\]RESistance:TLEVel](#), [\[SOURce:\]POWer:TLEVel](#), and [\[SOURce:\]VOLTage:TLEVel](#).

```
[SOURce:]TRANSient:COUNT <value> | MINimum | MAXimum | INFinity[, (@<chanlist>)]  
[SOURce:]TRANSient:COUNT? [MIN | MAX | INFinity,] [(@<chanlist>)]
```

The command sets the number of times that the continuous mode is executed before it is completed. The transient count can be set to any value between 1 and 9999.

Parameter	Typical Return
1 - 9999   MIN   MAX   INFinity	<transient count>
*RST 1	
Sets the transient count to 50: <code>TRAN:COUN 50, (@1)</code>	

```
[SOURce:]TRANSient:DCYClE <value> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]TRANSient:DCYClE? [MINimum | MAXimum,] [(@<chanlist>)]
```

The command sets the duty cycle for each of the transients when the generator is in CONTinuous mode. Units are in percentage (%)

Parameter	Typical return
1.8 - 98.2   MIN   MAX	<duty cycle>
*RST 50	
Sets the duty cycle to 10.5%: <code>TRAN:DCYC 10.5, (@1)</code>	

```
[SOURce:]TRANSient:FREQuency <value> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]TRANSient:FREQuency? [MINimum | MAXimum,] [(@<chanlist>)]
```

**NOTE** This command is the same as `INPut:PROtection:DELay`.

The command sets the frequency of the transient when the generator is in CONTinuous mode. Units are in Hertz (Hz)

Parameter	Typical Return
0.25 - 10000   MIN   MAX	<delay value>
*RST MAX	
Sets the frequency of transient to 50 Hz: <code>TRAN:FREQ 50, (@1)</code>	

[SOURce:]TRANSient:MODE CONTInuous | PULSe | TOGGle | LIST[, (@<chanlist>)]  
[SOURce:]TRANSient:MODE? [(@<chanlist>)]

The command sets the operating mode of the transient generator.

Mode	Description
CONTInuous	The transient generator puts out a continuous pulse stream.
PULSe	The transient generator puts out a single pulse upon receipt of a trigger.
TOGGle	The transient generator toggles between two levels upon receipt of a trigger.
LIST	The transient generator switch to list mode.

Parameter	Typical Return
CONT   PULS   TOGG   LIST	CONT, PULS, TOGG, or LIST
*RST LIST	
Sets the transient generator to continuous mode: TRAN:MODE CONT, (@1)	

[SOURce:]TRANSient:TWIDth <value> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]TRANSient:TWIDth? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the pulse width of the transients when the generator is in PULSe mode. Units are in seconds.

Parameter	Typical Return
0.0001 - 268,435   MIN   MAX	<pulse width>
*RST 0.0005	
Sets the pulse width to 0.5 second: TRAN:TWID 0.5, (@1)	

## TRIGger Subsystem

Trigger commands control the transient and acquisition subsystems.

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

The command triggers the measurement immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Generates an acquisition trigger immediately: TRIG:ACQ (@1)	

### Remarks

- You must initiate the measurement trigger system before you can send any trigger.
- At acquire trigger completion, the WTG-meas bit in the Status Operation Condition register is cleared.

TRIGger:ACQuire:CURRent[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]

TRIGger:ACQuire:CURRent[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the current triggered level of the input. Applies when the measurement trigger source is set to a level. Values are specified in amperes. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
minimum - maximum   MIN   MAX (The maximum values are 102% of the rating of the electronic load.)	<triggered level>
*RST MIN	
Sets the triggered current level to 3 A: TRIG:ACQ:CURR 3, (@1)	

TRIGger:ACQuire:CURRent:SLOPe POSitive | NEGative[, (@<chanlist>)]

TRIGger:ACQuire:CURRent:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the measurement trigger source is set to a level.

Slope	Description
POSitive	Specifies a rising slope of the input signal.
NEGative	Specifies a falling slope of the input signal.

Parameter	Typical return
POS   NEG	POS or NEG
*RST POS	
Sets the current slope to falling edge: TRIG:ACQ:CURR:SLOP NEG, (@1)	

TRIGger:ACQuire:SOURce BUS | CURRent<n> | EXTErnal | IMMEDIATE | VOLTage<n> | PIN<n>[, (@<chanlist>)]

TRIGger:ACQuire:SOURce? [(@<chanlist>)]

The command selects the trigger source for the acquisition system.

The query returns BUS, EXT, IMM, CURR<n>, VOLT<n>, or PIN<n>.

The following trigger sources can be selected:

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTErnal	Selects all connector pins configured as trigger sources.
IMMEDIATE	Generates a trigger as soon as the trigger system is INITiated.
CURRent<n>	Selects the input current level of a channel.
VOLTage<n>	Selects the input voltage level of a channel.
PIN<n>	Selects a digital port pin configured as a trigger input, where <n> indicates the pin number.

Parameter	Typical return
BUS   CURR<1-2>   EXT   IMM   VOLT<1-2>   PIN<1-3>	BUS, CURR<n>, EXT, IMM, VOLT<n>, or PIN<n>
*RST BUS	
Selects a current level on channel 1 as the acquire trigger:	TRIG:ACQ:SOUR CURR1, (@1)

TRIGger:ACQuire:VOLTage[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]

TRIGger:ACQuire:VOLTage[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the voltage triggered level of the input. Applies when the measurement trigger source is set to a level. Values are specified in volts. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
minimum - maximum   MIN   MAX (The maximum values are 102% of the rating of the electronic load.)	<triggered level>
*RST MIN	
Sets the triggered voltage level to 50 V:	TRIG:ACQ:VOLT 50, (@1)

TRIGger:ACQuire:VOLTage:SLOPe POSitive | NEGative[, (@<chanlist>)]  
 TRIGger:ACQuire:VOLTage:SLOPe? [ (@<chanlist>)]

The command sets the slope of the signal. Applies when the measurement trigger source is set to a level.

**Positive** specifies a rising slope of the input signal.

**NEGative** specifies a falling slope of the input signal.

Parameter	Typical return
POSitive   NEGative	POS or NEG
*RST POS	
Sets the voltage slope to rising edge: TRIG:ACQ:VOLT:SLOP POS, (@1)	

TRIGger:DLOG[:IMMEDIATE]

The command triggers the internal data logger immediately. This command overrides any selected trigger source and generate immediate triggers.

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

### Remarks

- You must initiate the data logger before you can send any trigger.

TRIGger:DLOG:CURRent[:LEVel] <value> | MINimum | MAXimum[, (@<chanlist>)]  
 TRIGger:DLOG:CURRent[:LEVel]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the current triggered level of the data logger. Applies when the data logger trigger source is set to a level. Values are specified in amperes. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
minimum - maximum   MIN   MAX (The maximum values are 102% of the rating of the electronic load.)	<triggered level>
*RST MIN	
Sets the triggered current level to 3 A: TRIG:DLOG:CURR 3, (@1)	

TRIGger:DLOG:CURRent:SLOPe POSitive | NEGative[, (@<chanlist>)]

TRIGger:DLOG:CURRent:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the data logger trigger source is set to a level.

Slope	Description
POSitive	Specifies a rising slope of the input signal.
NEGative	Specifies a falling slope of the input signal.

Parameter	Typical return
POS   NEG	POS or NEG
*RST POS	
Sets the current slope to falling edge: TRIG:DLOG:CURR:SLOP NEG, (@1)	

TRIGger:DLOG:VOLTage[:LEVe] <value> | MINimum | MAXimum[, (@<chanlist>)]

TRIGger:DLOG:VOLTage[:LEVe]? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the voltage triggered level of the data logger. Applies when the data logger trigger source is set to a level. Values are specified in volts. The minimum and maximum values depend on the ratings of the unit.

Parameter	Typical return
minimum - maximum   MIN   MAX (The maximum values are 102% of the rating of the electronic load.)	<triggered level>
*RST MIN	
Sets the triggered voltage level to 50 V: TRIG:DLOG:VOLT 50, (@1)	

TRIGger:DLOG:VOLTage:SLOPe POSitive | NEGative[, (@<chanlist>)]

TRIGger:DLOG:VOLTage:SLOPe? [(@<chanlist>)]

The command sets the slope of the signal. Applies when the data logger trigger source is set to a level.

**Positive** specifies a rising slope of the input signal.

**NEGative** specifies a falling slope of the input signal.

Parameter	Typical return
POSitive   NEGative	POS or NEG
*RST POS	
Sets the voltage slope to rising edge: TRIG:DLOG:VOLT:SLOP POS, (@1)	



TRIGger:DLOG:SOURce BUS | EXTernal | IMMEDIATE | CURRent<n> | VOLTage<n> | LRSKey | OOOKey  
 TRIGger:DLOG:SOURce?

The command selects the trigger source for the data logger.

The query returns BUS, EXT, IMM, CURR<n>, VOLT<n>, LRSK, or OOOK.

The following trigger sources can be selected:

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects all connector pins that have been configured as trigger sources
IMMEDIATE	Sets the trigger source to true. As soon as the data logger is INITiated, it will send the trigger immediately.
CURRent<n>	Selects the measured current level of a specific channel. <n> specifies a channel.
VOLTage<n>	Selects the measured voltage level of a specific channel. <n> specifies a channel.
LRSKey	Selects the List Run/Stop key.
OOOKey	Selects the Input On/Off keys and the All Inputs On/Off keys.

Parameter	Typical return
BUS   EXT   IMM   CURR<1-2>   VOLT<1-2>   LRSK   OOOK *RST IMM	BUS, EXT, IMM, CURR<n>, VOLT<n>, LRSK, or OOOK
Selects remote interface trigger command as the data log trigger source:	TRIG:DLOG:SOUR BUS

TRIGger:TRANsient[:IMMEDIATE] [(@<chanlist>)]

The command triggers the input immediately. This command overrides any selected trigger source and generate immediate triggers.

Parameter	Typical return
(none)	(none)
Generates a transient trigger immediately:	TRIG:TRAN (@1)

**Remarks**

- You must initiate the trigger system before you can send any trigger.
- At transient trigger completion, the WTG-tran bit in the Status Operation Condition register is cleared.

TRIGger:TRANSient:DElay <value> | MINimum | MAXimum[, (@<chanlist>)]

TRIGger:TRANSient: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 electronic load input. Programmed value can range from 0 to 0.255 seconds.

Parameter	Typical return
0 - 0.255   MIN   MAX	<delay value>
*RST 0	
Sets the trigger time delay to 0.1 s: TRIG:TRAN:DEL 0.1, (@1)	

TRIGger:TRANSient:SOURce BUS | EXTernal | IMMEDIATE | PIN1 | PIN2 | PIN3[, (@<chanlist>)]

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

The command selects the trigger source for the input trigger system.

The query returns BUS, EXT, IMM or PIN<n>.

PIN<n> must be configured as trigger input before it can be used as a trigger source. See [SOURce:]DIGital:PIN<n>:FUNctioN and [SOURce:]DIGital:PIN<n>:POLarity.

Trigger Source	Description
BUS	Selects a remote interface trigger command.
EXTernal	Selects ALL connector pins that have been configured as trigger sources.
IMMEDIATE	Sets the trigger source to true. As soon as the input is INITiated, it will send the trigger immediately.
PIN<n>	Selects a digital port pin configured as a trigger input where <n> indicates the pin number.

Parameter	Typical return
BUS   EXT   IMM   PIN1   PIN2   PIN3	BUS, EXT, IMM, PIN1, PIN2, or PIN3
*RST BUS	
Selects digital port pin 2 as the input trigger source: TRIG:SOUR PIN2, (@1)	

## Triggering Commands

The instrument's triggering system allows you to change current, power, resistance, or voltage input when a trigger is received. The typical process is:

1. Select one or more inputs with the `INSTRument:SElect` or `CHANnel:SElect` command.
2. Configure the triggered input levels by using `CURRent:TRIGgered`, `POWer:TRIGgered`, `RESistance:TRIGgered`, or `VOLTage:TRIGgered`.
3. Configure the current, power, resistance, or voltage mode to Step by using `CURRent:MODE`, `POWer:MODE`, `RESistance:MODE`, or `VOLTage:MODE`.
4. Specify the trigger source, either `BUS`, `EXTernal`, `IMMEDIATE`, or `PIN<n>`.
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 input change.
6. Send an `INITiate[:IMMEDIATE]` command. If the `IMMEDIATE` source is selected, the selected input is set to the triggered level immediately. If the `BUS` trigger source is selected, the input is set to the triggered level after the instrument receives the `*TRG` command.

## VOLTage Subsystem

Voltage commands program the input voltage and voltage protection functions. The SOURce keyword is optional in the following commands.

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>| MINimum | MAXimum | DEFault[, (@<chanlist>)]
```

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum | MAXimum | DEFault,] [(@<chanlist>)]
```

The command programs the immediate voltage level when the input is in constant voltage mode. Units are in volts.

The query returns the programmed voltage level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. MAX and MIN return the highest and lowest programmable voltage levels respectively for the selected range.

Parameter	Typical return
minimum - maximum   MIN   MAX   DEF (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<voltage level>
*RST MIN	
Sets the input voltage level to 20 V: <code>VOLT 20, (@1)</code>	

```
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <voltage>| MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MINimum | MAXimum,] [(@<chanlist>)]
```

The command programs the triggered voltage level when the input is in constant voltage mode. The triggered level is a stored value that is transferred to the input when an input step is triggered. Units are in volts.

The query returns the programmed voltage level in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. MAX and MIN return the highest and lowest programmable voltage levels respectively for the selected range.

Parameter	Typical return
minimum - maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<voltage level>
*RST MIN	
Sets the triggered voltage level to 5 V: <code>VOLT:TRIG 5, (@1)</code>	

[SOURce:]VOLTage:INHibit:VON[:LEVel] <voltage> | MAXimum | MINimum | DEFault[, (@<chanlist>)]  
[SOURce:]VOLTage:INHibit:VON[:LEVel]? [MINimum | MAXimum | DEFault,] [(@<chanlist>)]

When enabled, as specified by VOLTage:INHibit:VON:MODE, the electronic load will not sink current until the input voltage rises above the voltage-on setting. Units are in volts.

Parameter	Typical return
minimum - 153   MIN   MAX   DEF	<voltage-on value>
*RST MIN	
Specifies a voltage-on of 4 volts: VOLT:INH:VON 4, (@1)	

[SOURce:]VOLTage:INHibit:VON:MODE LATChing | LIVE | OFF[, (@<chanlist>)]  
[SOURce:]VOLTage:INHibit:VON:MODE? [(@<chanlist>)]

The command specifies the behavior of the under-voltage inhibit function. When enabled (not OFF), the electronic load will not sink current until the input voltage rises above the voltage-on setting specified by VOLTage:INHibit:VON <value>.

Mode	Description
LATChing	Allows the electronic load to sink current even if the voltage subsequently drops below the voltage-on setting.
LIVE	Turns the input off whenever the voltage drops below the voltage-on setting. Turns the input back on when the voltage reaches the voltage-on setting.
OFF	Turns the under-voltage inhibit function off.

Parameter	Typical return
LIVE   LATChing   OFF	LIVE, LATC, or OFF
*RST OFF	
Sets the inhibit mode to latching: VOLT:INH:VON:MODE LATC, (@1)	

### Remarks

- The under-voltage inhibit function will ignore any programmed on/off delays.
- The under-voltage inhibit function is not available when electronic loads are grouped.
- The under-voltage inhibit function is not available in CV operating mode.

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

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

The command determines what happens to the input 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 input voltage remains at the immediate value.
STEP	The input goes to the triggered level when a trigger occurs.
LIST	The input follows the list values when a trigger occurs.

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

[SOURce:]VOLTage:RANGe <voltage> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]VOLTage:RANGe? [MINimum | MAXimum,] [(@<chanlist>)]

The command sets the voltage range. The value that you enter must be the highest value in volts that you expect to input. The instrument selects the range with the best resolution for the value entered. The following ranges are available for the electronic load.

	EL33133A	EL34143A	EL34243A
High range	from 0.02 V to 153 V	from 0.015 V to 153 V	
Low range	from 0.005 V to 15.3 V	from 0.003 V to 15.3 V	

The query returns the maximum voltage that can be input on the range that is presently set. The voltage is in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas.

Parameter	Typical return
minimum – maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<voltage range>
*RST MAX	
Sets the high voltage range (from 0.015 V to 153 V):	VOLT:RANG 100, (@1)

## Remarks

- This command is coupled with [SOURce:]VOLTage. This means that if a range command places an input on a range with a lower maximum voltage than the present voltage level, an out of range error occurs. This also occurs if a voltage is programmed that is too large for the present range. Errors can be avoided by sending both level and range commands in the same SCPI message. When the range and level information is received as a set, no range/setting conflict occurs.
- If a programmed range change occurs while the input is enabled, the input will be temporarily disabled while the range switch occurs. The transition from on-to-off and then from off-to-on will also be delayed by the settings of [INPut:DELay:FALL](#) and [INPut:DELay:RISE](#).

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

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

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

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

The command sets the voltage slew rate. The slew rate is set in volts per second and affects all programmed voltage changes, including those due to the input state turning on or off. The slew rate can be set to any value between the minimum value and 9.9E+37. For very large values, the slew rate will be limited by the analog performance of the input circuit. The keywords MAX or INFinity set the slew rate to maximum.

Parameter	Typical Return
minimum – 9.9E+37   MIN   MAX   INFinity	<slew rate>
*RST MAX	
Sets the positive input slew rate to 5 V per second: VOLT:SLEW 5, (@1)	
Sets the negative input slew rate to 5 V per second: VOLT:SLEW:NEG 5, (@1)	

## Remarks

- The query returns the value that was sent. If the value is less than the minimum slew rate, only the minimum value is returned. The resolution of the slew setting is also the minimum value, which can be queried using VOLTage:SLEW? MIN. The exact value varies slightly based on calibration.
- The query returns the programmed slew rate in the form +n.nnnnnnE+nn for each channel specified. Multiple responses are separated by commas. If a slew rate of 9.9E+37 is returned, it means that the maximum or fastest slew rate has been set.

[SOURce:]VOLTage:SLEW:COUPlE ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]VOLTage:SLEW:COUPlE? [(@<chanlist>)]

Sets the voltage slew tracking state.

Parameter	Typical return
ON   1   OFF   0	1 or 0
*RST OFF	
Enables the voltage slew tracking state: VOLT:SLEW:COUP ON, (@1)	



[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum? [(@<chanlist>)]

[SOURce:]VOLTage:SLEW:NEGative:MAXimum ON | OFF | 1 | 0[, (@<chanlist>)]

[SOURce:]VOLTage:SLEW:NEGative:MAXimum? [(@<chanlist>)]

The command enables/disables the maximum slew rate override. When enabled, the slew rate is set to its maximum value. When disabled, the slew rate is set to the immediate value set by the VOLTage:SLEW command. Use VOLTage:SLEW? MAX to query the maximum slew rate.

Parameter	Typical Return
ON   OFF   1   0	0 or 1
*RST ON	
<hr/>	
Enables the positive maximum slew rate: VOLT:SLEW:MAX ON, (@1)	
Enables the negative maximum slew rate: VOLT:SLEW:NEG:MAX ON (@1)	
<hr/>	

### Remark

- The VOLTage:SLEW:MAX command is coupled to the VOLTage:SLEW command. If the VOLTage:SLEW command sets the slew rate to MAX or INFINITY, VOLTage:SLEW:MAX is enabled. If the slew rate is set to any other value, VOLTage:SLEW:MAX is disabled.

[SOURce:]VOLTage:SENSe[:SOURce] INTernal | EXTernal[, (@<chanlist>)]  
[SOURce:]VOLTage:SENSe[:SOURce]? [(@<chanlist>)]

The command sets the state of the remote sense relays.

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 input 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 input 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 electronic load to short the input 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 input 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)	

[SOURce:]VOLTage:TLEVel <value> | MINimum | MAXimum[, (@<chanlist>)]  
[SOURce:]VOLTage:TLEVel? [MINimum | MAXimum,] [(@<chanlist>)]

The command specifies the transient level of the input voltage. The transient function switches between the immediate setting and the transient level. Units are in volts.

Parameter	Typical return
minimum – maximum   MIN   MAX (Values are range and product-model dependent. See <a href="#">Programming Range</a> )	<slew rate>
*RST MIN	
Sets the transient input voltage to 5 V: VOLT:TLEV 5, (@1)	



This information is subject to change without notice.

© Keysight Technologies 2020, 2021  
Edition 1, September 2021  
Printed in Malaysia



EL34243-90007

[www.keysight.com](http://www.keysight.com)