



**Revision F**  
**August 2006**  
Copyright ©2002-2006  
By California Instruments  
All rights reserved.  
**P/N 5005-961**

**KDC Series**  
**DC Power Supply**  
**Programming Manual**  
Series I & II

**TEL: +1 (858) 404 6936**  
**FAX: +1 (858) 677-0940**  
**Email: [sales@argantix.com](mailto:sales@argantix.com)**  
**Web Site: <http://www.argantix.com>**

---

Programming Manual  
DC Power Supply  
Argantix

KDC Series I  
KDC Series II

© Copyright 2002-2006, California Instruments, Rev F

 **SAFETY SUMMARY**

**This power supply contains high voltage and current circuits that are potentially lethal. Because of its size and weight, electrical and mechanical stability must be ensured. The following safety guidelines must be followed when operating or servicing this equipment. These guidelines are not a substitute for vigilance and common sense. If this power supply is not used as specified in this manual, the protection provided by this equipment may be impaired. California Instruments assumes no liability for the customer's failure to comply with these requirements.**

**BEFORE APPLYING POWER**

1. Check the units input rating on the type label and verify the correct voltage is applied to the unit (for example 208 V L-L, 3 Phase).
2. The chassis and cabinet of this power supply must be grounded to minimize shock hazard. A chassis ground is provided at the input terminal block. This is located at the back of the cabinet on the right hand side (looking at the back of the unit). The chassis ground must be connected to an electrical ground through an insulated wire (green/yellow) of sufficient gauge.
3. The KDC Series power supplies do not have internal AC mains disconnect. Instead, protection is provided by fuses. The on/off button only engages or disengages the bias supplies, it does not disconnect the AC mains to the unit. **It is strongly recommended to use a suitably rated circuit breaker or mains disconnect device with branch protection rating per local electrical codes between point of connection and the KDC unit AC input to enable complete disconnection of AC input power.**

**FUSES**

Use only fuses of the specified current, voltage, and protection speed.

**DO NOT OPERATE IN A VOLATILE ATMOSPHERE**

Do not operate the power supply in the presence of flammable gases or fumes. This product is designed to operate in a controlled environment. Do not expose to rain or snow.

**DO NOT TOUCH ENERGIZED CIRCUITS**

Disconnect the power cable before servicing this equipment. Even with the power cable disconnected, high voltage can still exist on some circuits. Discharge these voltages before servicing. Allow at least 30 minutes for all internal circuits to discharge before removing the top cover. Only qualified service personnel may remove covers, replace components or make adjustments.

**DO NOT SERVICE ALONE**

Do not remove covers, replace components, or make adjustments unless another person, who can administer first aid, is present.

**DO NOT EXCEED INPUT RATINGS**

Do not exceed the rated input voltage or frequency. Additional hazards may be introduced because of component failure or improper operation.

**DO NOT MODIFY INSTRUMENT OR SUBSTITUTE PARTS**

Do not modify this instrument or substitute parts. Additional hazards may be introduced because of component failure or improper operation.

**MOVING THE POWER SUPPLY**

When moving the power supply, observe the following:

1. Remove all AC power to unit.
2. Use two people to prevent injury.

**SURFACE STABILITY**

Operate the power supply only on a level surface.

SAFETY SYMBOLS:



THIS SYMBOL INDICATES DIRECT CURRENT



THIS SYMBOL INDICATES ALTERNATING CURRENT



THIS SYMBOL INDICATES BOTH DIRECT AND ALTERNATING CURRENT



THIS SYMBOL INDICATES THREE-PHASE ALTERNATING CURRENT



THIS SYMBOL INDICATES EARTH (GROUND) TERMINAL



THIS SYMBOL INDICATES PROTECTIVE CONDUCTOR TERMINAL



THIS SYMBOL INDICATES FRAME OR CHASSIS TERMINAL



THIS SYMBOL INDICATES ON (SUPPLY)



THIS SYMBOL INDICATES OFF (SUPPLY)



THIS SYMBOL INDICATES CAUTION, RISK OF ELECTRIC SHOCK



THIS SYMBOL INDICATES CAUTION (REFER TO ACCOMPANYING DOCUMENTS)

**Electrical Safety Symbols Used in This Manual**

## WARRANTY INFORMATION

CALIFORNIA INSTRUMENTS CORPORATION warrants each instrument manufactured by them to be free from defects in material and workmanship for a period of five years from the date of shipment to the original purchaser. Excepted from this warranty are fuses and batteries that carry the warranty of their original manufacturer where applicable.

CALIFORNIA INSTRUMENTS will service, replace, or adjust any defective part or parts, free of charge, when the instrument is returned freight both ways prepaid, and when examination reveals that the fault has not occurred because of misuse, abnormal conditions of operation, user modification, or attempted user repair. Equipment repaired beyond the effective date of warranty or when abnormal usage has occurred will be charged at applicable rates.

CALIFORNIA INSTRUMENTS will submit an estimate for such charges before commencing repair, if so requested.

## SERVICE PROCEDURE

If a fault develops, notify CALIFORNIA INSTRUMENTS at [support@argantix.com](mailto:support@argantix.com) or its local representative, giving full details of the difficulty, including the model number and serial number. On receipt of this information, service information or a Return Material Authorization (RMA) number will be given. Add the RMA number furnished to the shipping label. Pack the instrument carefully to prevent transportation damage, affix label to shipping container, and ship freight prepaid to the factory. CALIFORNIA INSTRUMENTS shall not be responsible for repair of damage due to improper handling or packing. Instruments returned without RMA No. or freight collect may be refused at California Instruments discretion. All Argantix products repaired will be returned freight collect, Ex Works CALIFORNIA INSTRUMENTS, 9689 Towne Centre Drive, San Diego, CA 92121-1964. If requested, an estimate of repair charges will be made before work begins on repairs not covered by the Warranty.

## DAMAGE IN TRANSIT

The instrument should be tested when it is received. If it fails to operate properly, or is damaged in any way, a claim should be filed immediately with the carrier. The claim agent should obtain a full report of the damage, and a copy of this report should be forwarded to us by fax or email (Fax: 858 677 0940, Email: [support@argantix.com](mailto:support@argantix.com)). CALIFORNIA INSTRUMENTS will prepare an estimate of repair cost and repair the instrument when authorized by the claim agent. Please include model number and serial number when referring to the instrument.

## SPARE PARTS

To order spare parts, user manuals, or determine the correct replacement part for your Argantix products, please contact the Customer Service department by phone at + 1 858 404 6936, press 2 or by email [support@argantix.com](mailto:support@argantix.com).

## Table of Contents

1	Introduction .....	7
1.1	General Description .....	7
1.2	Bench Top use .....	8
1.3	Equipment Rack use.....	8
2	Remote Control .....	9
2.1	Introduction.....	9
2.2	RS232C Interface .....	10
2.3	Optional IEEE Interface .....	12
2.4	KDCGUI Installation.....	13
2.5	Trouble Shooting - RS232C.....	15
2.6	Trouble Shooting - IEEE-488 / GPIB .....	18
2.7	Software Registration .....	18
3	Introduction to SCPI.....	19
3.1	Conventions Used in This Manual .....	19
3.2	The SCPI Commands and Messages.....	19
3.3	Using Queries .....	21
3.4	Structure of a SCPI Message .....	22
3.5	SCPI Data Formats.....	24
4	SCPI Command Reference .....	25
4.1	Abort Subsystem .....	25
4.2	Calibration Subsystem .....	26
4.3	Initiate Subsystem .....	33
4.4	Measurement Subsystem .....	35
4.5	Output Subsystem .....	41
4.6	Sense Subsystem.....	44
4.7	Source Subsystem.....	45
4.8	Status Subsystem Commands.....	64
4.9	System Commands .....	70
4.10	Trigger Subsystem.....	74
4.11	Common Commands .....	76
5	Programming Examples.....	83
5.1	Introduction.....	83
5.2	Programming the Output .....	83
5.3	Basic Measurements .....	84
5.4	Setting the Power on Initialization Values .....	85
5.5	Programming Output Transients.....	86
5.6	Triggering Output Changes .....	90
5.7	Acquiring Measurement Data .....	92
5.8	Controlling the Instantaneous Voltage and Current Data Buffers .....	95
5.9	Trigger System Summary .....	96
6	Status Registers.....	97
6.1	Power-On Conditions.....	97
6.2	Standard Event Status Group.....	97
6.3	Status Byte Register .....	97
6.4	Examples.....	99
7	Option Commands .....	101
7.1	Introduction.....	101
7.2	Mil-Std 704 Option -704.....	102
	Appendix A: SCPI Command tree.....	105
	Appendix B: SCPI Conformance Information.....	109
	Appendix C: Error Messages .....	110
	Index.....	114

## List of Figures

Figure 1-1: KDC Series DC Power Supply .....	7
Figure 2-1: KDCGUI Main screen .....	9
Figure 2-2: Rear Panel View .....	12
Figure 2-3: System Properties Dialog Box .....	16
Figure 2-4: Advanced Port Settings Dialog Box .....	16
Figure 2-5: COM Port Properties Dialog Box .....	16
Figure 3-1: Partial Command Tree .....	20
Figure 3-2: Command Message Structure .....	22
Figure 5-1 : Transient Trigger System Model.....	90
Figure 5-2: Measurement Acquisition Trigger Model.....	94
Figure 5-3: Pre-event and Post-event Triggering .....	96
Figure 5-4: Trigger system block diagram .....	96
Figure 6-1: Power Supply Status System Model .....	98

## List of Tables

Table 4-1: Mode query results.....	56
Table 4-1: PULSe:HOLD = WIDTH parameters .....	58
Table 4-2: PULSe:HOLD = DCYClE parameters.....	58
Table 4-3: Bit Configuration of Standard Event Status Enable Register.....	77
Table 4-4: Bit Configuration of Standard Event Status Register.....	78
Table 4-5: *RST Default Parameter Values.....	80
Table 4-6: Status Registers - Power on Conditions.....	80
Table 4-7: Bit Configuration of Status Byte Register .....	81
Table 5-1: Factory Default Power on Settings.....	85
Table 7-1: Bus Error Messages.....	113

# 1 Introduction

## 1.1 General Description



Figure 1-1: KDC Series DC Power Supply

The Argantix KDC Series of DC Programmable Power Supplies are designed specifically for laboratory test and systems applications requiring single output, variable DC voltage and current with good ripple and regulation characteristics and sophisticated control and measurement capabilities. The KDC Series comes in a 3U chassis and can operate in constant current or constant voltage mode with auto crossover feature. Available Power levels are 5kW, 10kW and 15kW.

Simple menu driven front panel controls enable the voltage and current limit to be changed. Measurements of voltage, current, peak current and power can also be read from the front panel LCD display.

The front panel contains an output on/off button for controlling the DC supply output and an Output ON LED indicator that informs the operator of the output status at all time.

The LCD display is used to display the programmed voltage and current as well as all measurements. More advanced functions can be accessed through nested levels of menus.

A standard RS232C and optional IEEE 488 interface is available for applications that require remote control and measurements. While operated remotely, the front panel can be locked out and the remote control status is indicated by a REMOTE LED.

The power on/off toggle switch can be used to turn the DC supply on or off.

This programming manual applies to both KDC Series I and KDC Series II controllers. Series II controllers are used in KDC top assemblies with Revision E on the model/serial tag. Older revision KDC units will have KDC Series I controllers.



---

## 1.2 Bench Top use

---

The KDC Series DC Supply may be used as a bench top unit provided adequate provisions are made to protect the end-user from touching the output terminals on higher voltage models. To this end, output covers are provided on these KDC units. These covers must be installed during use and may only be removed when the unit is not in use and disconnected from AC mains.

Adequate air flow must be maintained at all times so care should be taken not to block the top and side air intakes or the rear air exhaust. Leave at least 4" of clearance at the back of the unit to allow proper airflow.

---

## 1.3 Equipment Rack use

---

The KDC Series uses a 19" cabinet with rack ears and can be installed in a standard equipment rack. Due to the weight of the unit however, it must be properly supported by either a shelf or L brackets on both sides. The rack ears are not designed to support the full weight of the KDC unit when installed in a cabinet but only to prevent it from sliding out.

Contact Argantix customer support ([support@calinst.com](mailto:support@calinst.com)) for information on rack mount kits for specific cabinets depths.

Proper airflow must be maintained in the instrument cabinet and the rear of the cabinet should not be closed off by a door but rather a perforated screen allowing adequate airflow. It is also important to leave enough clearance above each unit as some of the air intake is on the top cover.

---

**Note:**            **It is recommended to leave a 1.75" (1 U) space above each KDC unit.**

---

This space may be covered by a louvered cover plate to allow some air intake from the front of the cabinet. When planning cabinet space, allow for 4U per DC supply.

## 2 Remote Control

### 2.1 Introduction

The KDC Series comes standard with an RS232C serial interface on the rear panel. It can optionally be furnished with a combination IEEE-488 and analog control interface at the time of purchase. The GPIB interface is part of the -IF option package.

Units are shipped with the Argantix KDC Graphical User Interface program - KDCGUI. This Windows™ program provides a soft front panel to the instrument when connected to a PC through the RS232C or IEEE-488 interface. Additional benefits are obtained from using the PC as a control interface. Some of these benefits include the ability to store measurement data to disk and produce output transients to simulate various DC supply conditions.

The KDCGUI is a Windows™ program and as such requires a PC capable of running Windows 98™, or Windows NT™/2000/XP. For best performance, a Pentium based PC is recommended.

**Note:** The KDCGui was developed to work with the National Instruments Gpib-32.dll only. For use over the GPIB bus, a National Instruments brand GPIB controller is required in the PC.

Complete information on how to use the KDCGUI can be found in the on-line help supplied with the program. Use the Help menu or press on any of the many Help keys located in all program windows for an explanation of the relevant screen or function.

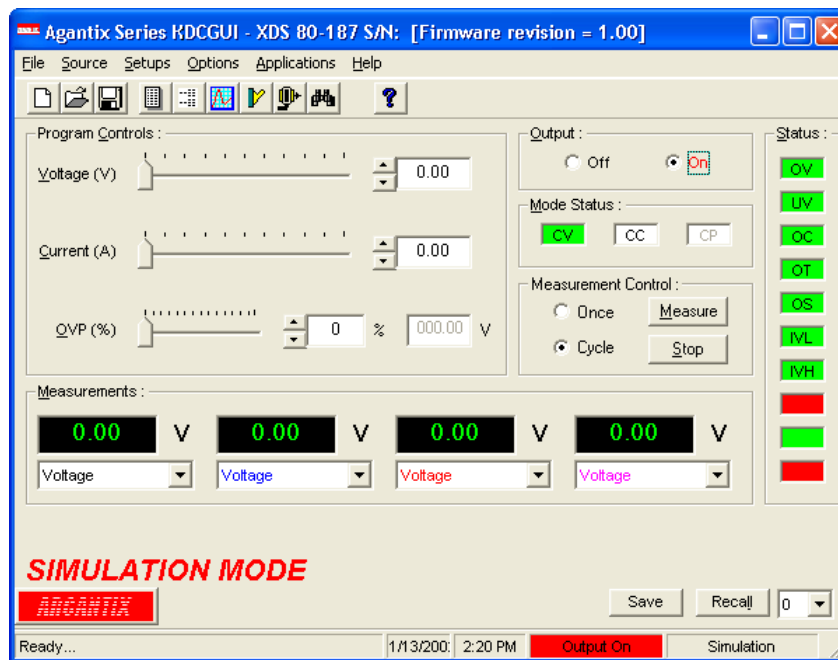


Figure 2-1: KDCGUI Main screen

## 2.2 RS232C Interface

A suitable cable to connect the power supply to a 9 pin PC-AT style serial port is supplied with the power supply.

The KDC Series expects a LF (Hex 10) terminator at the end of each string sent over the RS232C interface. If the programming environment you use to develop test programs does not append a LF terminator to each output string, the KDC Series will not respond. This may be the case for programs like LabView™ using VISA drivers.

### 2.2.1 Serial Communication Test Program

The following sample program written in Quick-BASIC can be used to check communication to the KDC Series Power supply over the RS232C serial interface. The interface is optional and must be installed for this to work.

```
'Argantix RS232C Communication Demo Program
'(c) 2002 Copyright Argantix, All Rights Reserved
'
'
'This program is for demonstration purposes only and is not to be
'used for any commercial application
'=====
'Function and Subroutine Declarations
DECLARE FUNCTION retstring$ ()
'=====
'MAIN PROGRAM CODE
'=====
'OPEN COM2. Replace with COM1, COM3 or COM4 for Com port used
'The input and output buffers are set to 2K each although
'this is not required for most operations.

OPEN "COM2:19200,n,8,1,BIN,LF,TB2048,RB2048" FOR RANDOM AS #1 LEN = 1
CLS
PRINT "**** INTERACTIVE MODE ****"
'Enter and endless loop to accept user entered commands
DO
  INPUT "Enter Command ('quit' to exit)--> ", cmd$
  IF cmd$ <> "QUIT" AND cmd$ <> "quit" THEN
    IF cmd$ <> "" THEN
      PRINT #1, cmd$ + CHR$(10);
    END IF
    IF INSTR(cmd$, "?") THEN
      PRINT "Power supply query response = "; retstring$
    END IF
    'Check for Errors after each command is issued
    PRINT "Sending non-query syntax check *ESR?"
    'On fast PC's we may have to hold off between commands
    FOR t = 0 TO 1000: NEXT t
    PRINT #1, "*ESR?" + CHR$(10);
    esr% = 0 'Clear last error
    'Mask off bits 5,4,3,2 only. Other bits are not used.
    esr% = VAL(retstring$) AND 60
    'Process esr% value for error bits
    IF esr% AND 4 THEN
      PRINT "**** Query Error Reported by power supply ****"
    END IF
    IF esr% AND 8 THEN
      PRINT "**** Instrument Dependent Error Reported by power supply ****"
    END IF
    IF esr% AND 16 THEN
      PRINT "**** Command Execution Error Reported by power supply ****"
    END IF
    IF esr% AND 32 THEN
      PRINT "**** Command Syntax Error Reported by power supply ****"
    END IF
    'Clear ERR. -XXX Message from front panel if any error occurred
```

```
    IF esr% <> 0 THEN
      PRINT #1, "*CLS" + CHR$(10);
    END IF
  END IF
LOOP UNTIL cmd$ = "QUIT" OR cmd$ = "quit"
'Close COM port on exit
CLOSE #1
END
'=====

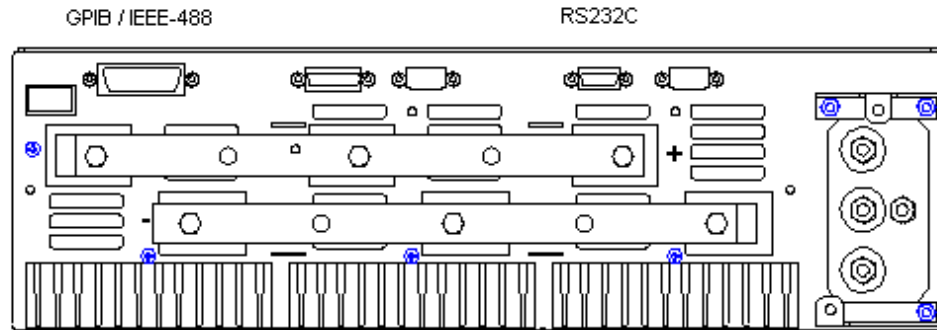
FUNCTION retstring$
  'This function returns a response string from the KDC Series
  'power supply. The QBasic statement LINE INPUT cannot be used
  'as the KDC Series does not return a CR <13> after a response
  'message. The LINE INPUT function waits for a CR before
  'returning a string. The P Series returns a LF <10> instead
  'so we need to poll each returned character for a LF to
  'assemble the response string. The COM port needs to be
  'opened AS random with a record length of 1 for it to support
  'this function. Also, the device number is assumed to be #1
  'Optionally, this value could be passed as a parameter to
  'make this program more generic.

  DIM char AS STRING * 1
  DIM resp AS STRING
  char = ""
  resp = ""
  DO
    char = INPUT$(1, #1)
    resp = resp + char
  LOOP UNTIL char = CHR$(10)
  'Return result
  retstring = LEFT$(resp, LEN(resp) - 1)
END FUNCTION
```

## 2.3 Optional IEEE Interface

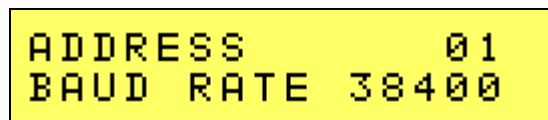
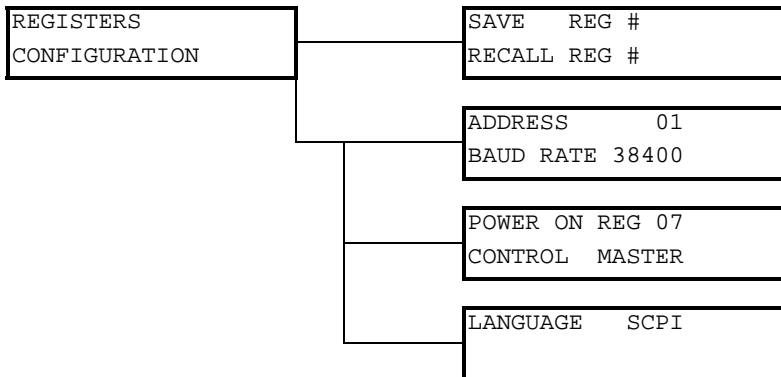
The KDC Series can optionally be equipped with an IEEE-488 interface in addition to the standard RS232C interface.

Figure 2-2: Rear Panel View



The IEEE address of the KDC DC Supply is set from the Configuration menu. Refer to the KDC Series User Manual (P/N 5005-960) for more details.

The IEEE/GPIB address selection can be accessed from the front panel Configuration screen.



Sets the selected IEEE / GPIB bus address for the optional IEEE/GPIB interface. Factory default is address 1. The left knob (VOLTAGE) can be used to scroll through the 0 through 31 address range. Do not use address 0 as this address is typically reserved for the GPIB controller.

---

## 2.4 KDCGUI Installation

---

This section covers installation of the KDCGUI from the distribution disk to the user's PC. To successfully install and operate the KDCGUI program, you will need the following equipment:

Argantix KDC Series DC power supply.

PC capable of running Windows 98™ , Windows 2000™ or Windows XP™

- RS232C communications port and straight-through RS232C serial cable (supplied with the product.)

or

- If option -IF (GPIB) is installed, a National Instruments IEEE-488 Controller Card

---

**Note:** *The KDCGUI can be run in the absence of a DC power supply. If no DC supply is available, the KDCGUI can be operated in a simulation mode. The program will detect these conditions and start up in simulation mode after notifying the operator. Measurements in this case will be simulated and should not be used for any analytical purpose.*

---

### 2.4.1 Connecting to the PC Using RS232

---

Connect the power supply to the PC using an RS232C cable. A straight-through 9 pin DB9 male to DB9 female cable can be used. This cable is supplied in the skip kit for the KDC Series supply. (P/N 250709)

The power supply RS232C port settings are set as follows:

Baud rate:	Selectable from Configuration menu. . Refer to the KDC Series User Manual (P/N 5005-960) for details. Set KDCGUI baud rate to match baud rate selected on DC Supply.
Data bits:	8
Stop bits	1
Parity bits:	none
Handshake:	Hardware

The DC supply must be configured to match the above settings.

### 2.4.2 Connecting to the PC Using the optional IEEE-488/GPIB interface

---

Connect the power supply to the PC using an IEEE-488 interface cable. A National Instruments GPIB controller card is required to use the KDCGUI program. Set the desired IEEE address from the configuration menu. See section 2.3.

---

### 2.4.3 Installing the Software from CD ROM

---

The KDCGUI software is distributed on CD-ROM. The KDCGUI must be installed from CD using the included setup program as all required files are compressed. To install the KDCGUI, proceed as follows:

1. Turn on the PC and boot up in Windows™
2. Insert the CD in your CD-ROM drive.
3. Run the Setup.exe program from the root directory of the CD.
4. A CD Browser like program will open. Select the product series for which you want to install the software and select the GUI Software tab. Select an available operating system (typically 32 bit Windows) and click on the Install button to begin the installation.
5. Follow the instructions provided by the setup program to complete the installation.
6. When the installation has completed, remove the CD ROM and store in a safe place.

If prompted to do so, reboot the PC to activate the new settings. You are now ready to start using the KDCGUI software.

---

## 2.5 Trouble Shooting - RS232C

---

This section provides guidelines for resolving communication problems that may occur when using the KDCGUI software under Windows™.

You may encounter problems when using the serial interface with the KDCGUI program that is supplied with the interface option for this source.

### Symptoms:

1. Unable to verify connection to the power supply through RS232C interface. Time-out occurs and dialog box appears that asks if you want to switch to simulation mode. An error message (Error -113, "Undefined Header") may appear on the front panel LCD of the power supply.
- Or -
2. Verification is successful but slewing of voltage or current limit results in an Error -113, "Undefined Header".

### Things to check first:

1. Is any PC com port available for communication with the power supply? On older PC's com port interrupts are often shared between com 1 and com3 as well as com2 and com 4. If any of these other com ports is in use, it may prevent you from using the com port that shares the same interrupt.
2. Did you use the RS232C cable that was supplied with the power source? If not, make sure you obtain the correct cable. (DB9 to DB9, straight through, male to female).

### Resolution for Symptom 1

The KDC Series of power supplies require hardware handshaking to control data flow from the PC to the Power supply. After receiving a command, the power supply asserts the DTR line to hold off further communication from the PC until the command just sent has been processed. Under Windows, it is possible to use a 16 byte transmit buffer if a 16550 UART is present. When enabled, this mode may result in more than one command being placed in the UART output buffer. If this happens, the handshake from the Power supply will not prevent the content of this buffer being sent to the Power supply and thus will result in more than one command being sent to the Power supply.

To resolve this problem, the UART FIFO length needs to be reduced or disabled altogether. The procedure to disabled the FIFO mode is outlined in the next paragraph.

Note: If turning off the UART FIFO conflicts with other applications that need to run on the same PC using the same COM port, this solution may not be acceptable. In this case, skip this step and proceed to the second symptom resolution, which relies on a software implemented hold-off instead of turning off the FIFO buffer.



**Procedure**

1. Make sure the KDCGUI program is closed and no device is using the com port in use.
2. From the Windows™ desktop, right click on the “My Computer” icon which is located in the top left corner of the screen.
3. From the popup menu, select the “Properties” entry.
4. This brings up the System Properties dialog box
5. From the tabs shown at the top of the dialog box, select the “Device Manager” tab.
6. The relevant screen is shown in Figure 2-3.
7. Expand the Ports(COM&LPT) tree selection by clicking on the ‘+’ sign
8. Select the COM port you plan to use to communicate with the P or KDC Series power supply.
9. With the correct COM port selected, click on the Properties button at the bottom left of the dialog box.
10. This brings up the settings dialog box for the COM port. In this example, COM2 is used.
11. From the tabs at the top of the dialog box, select the “Port Settings” tab.
12. This should bring up the dialog shown in Figure 2-5.
13. Next, select the Advanced button to bring up the advanced port settings dialog box shown in Figure 2-4.

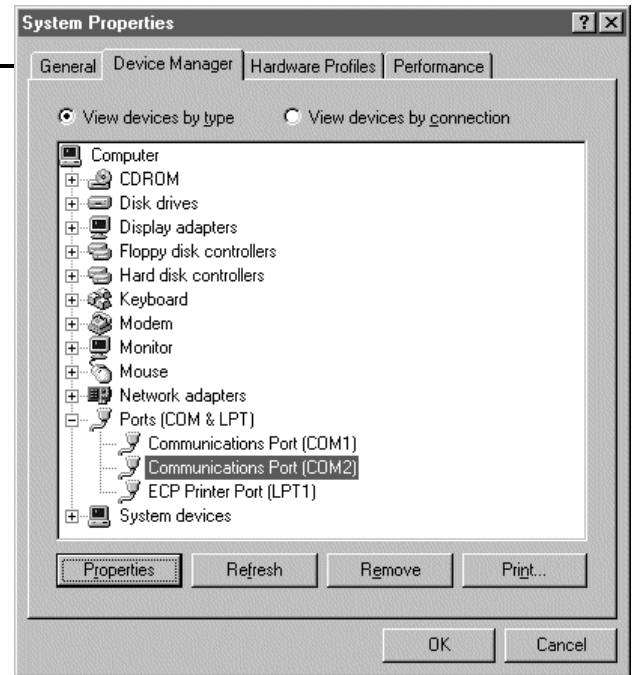


Figure 2-3: System Properties Dialog Box

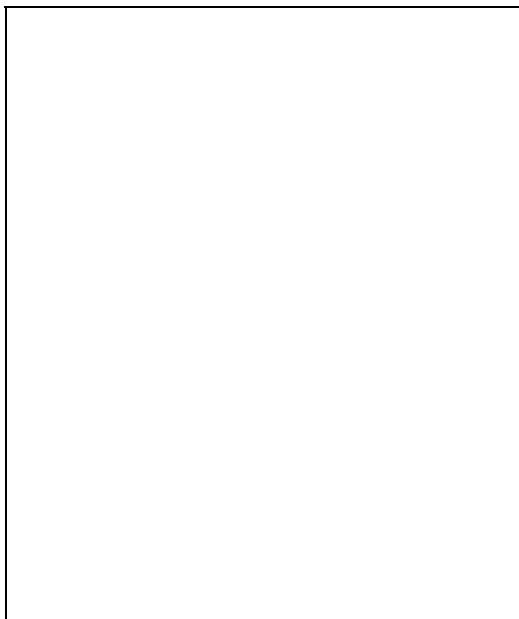


Figure 2-4: Advanced Port Settings Dialog Box

14. At the top of the dialog box, disable the “Use FIFO buffers (requires 16550 compatible UART)” checkbox. This will bypass the UART FIFO’s and enable the hardware handshake to work correctly.
15. Click on the all OK buttons to close all dialog boxes that remain open.
16. This should enable the KDCGUI to work correctly.

Figure 2-5: COM Port Properties Dialog Box

## Resolution for Symptom 2

Execute the suggested procedure to resolve symptom 1 first. If an occasional error continues to occur while slewing the voltage or frequency slider controls in the KDCGUI program, increase the command delay parameter located in the System, Interface dialog. The default value for the CmdDelay is set to 30 and is displayed below the command line of the Interface window. Change this value to 40 by entering the new value from the keyboard.

Try to connect again. You may have to cycle the power on the power supply if it still shows the Error -113, "Undefined Header" display.

Higher values than 40 may be set for the CmdDelay parameter if this fix does not resolve your problem.

---

## 2.6 Trouble Shooting - IEEE-488 / GPIB

---

This section provides guidelines for resolving communication problems that may occur when using the KDC series with or without the KDCGUI software with a IEEE-488 / GPIB interface.

### Symptoms:

1. Unable to verify connection to the power supply through IEEE-488 interface. Time-out occurs any time an attempt is made to verify connection.

### Things to check:

1. Is the interface option set to use the IEEE-488 interface instead of the RS232C interface?
2. Does the bus address selected in the configuration menu of the KDC supply match the address selected in the System, Interface dialog windows?
3. Did you connect the cable in correctly and are both ends of the GPIB cable securely tied down?

---

## 2.7 Software Registration

---

Updates of this and other Argantix programs are posted on a regular basis on the Argantix web site. You can find available programs by selecting the Software, GUI's and Drivers menu. To gain access to these downloads, you will need to register as a user on our web site. For instructions on how to register and request the required access level for software downloads, visit our web site at

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

### 3 Introduction to SCPI

---

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over the RS232 or IEEE 488 bus. The same SCPI commands and parameters control the same functions in different classes of instruments. For example, you would use the same MEAS:VOLT? command to measure the Power supply output voltage or the output voltage of a SCPI-compatible Multimeter.

#### 3.1 Conventions Used in This Manual

---

Angle brackets<>	Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.
Vertical bar	Vertical bars separate alternative parameters. For example, 0   1 indicates that either "0" or "1" can be used as a parameter.
Square Brackets [ ]	Items within square brackets are optional. The representation [SOURce:]VOLT means that SOURce: may be omitted.
Boldface font	Boldface font is used to emphasize syntax in command definitions. CURR <NRf> shows a command definition.
Upper case font	Upper case font is used to show program lines in text. OUTP 1 shows a program line.

#### 3.2 The SCPI Commands and Messages

---

This paragraph explains the syntax difference between SCPI Commands and SCPI messages.

The power supply supports two types of commands, IEEE-488-2 common and SCPI subsystem commands.

- IEEE-488-2 common commands are generally not related to specific operations but to controlling overall Power supply functions such as reset, status and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk:

\*RST      \*IDN?   \*SRE 255

- SCPI Subsystem commands perform specific Power supply functions. They are organized into an inverted tree structure with the "root" at the top. Some are single commands while others are grouped within specific subsystems. **You must include the root header in all commands sent to the Power supply.**

Refer to appendix A for the SCPI tree structure.

##### Types of SCPI Messages

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

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

## The SCPI Command Tree

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

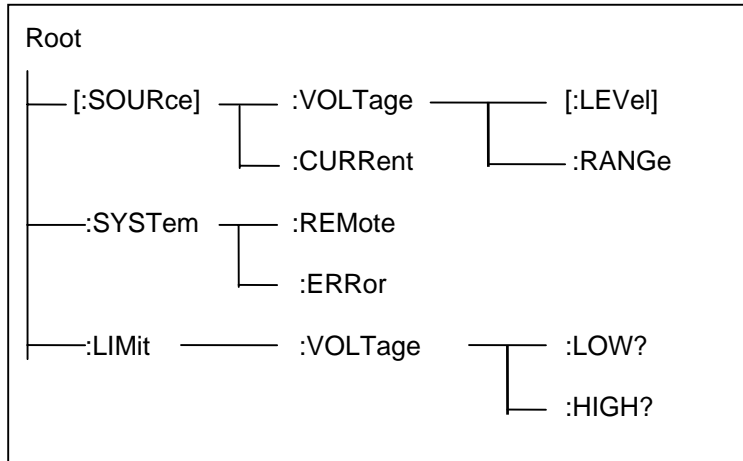


Figure 3-1: Partial Command Tree

### The Root Level

Note the location of the ROOT node at the top of the tree. Commands at the root level are at the top level of the command tree. The SCPI interface is at this location when:

- the Power supply is powered on
- a device clear (DCL) is sent to the Power supply
- the SCPI interface encounters a message terminator
- the SCPI interface encounters a root specifier

### Active Header Path

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

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

### Moving Among Subsystems

In order to combine commands from different subsystems, you need to be able to restore the active path to the root. You do this with the root specifier (:). For example, you could set the output voltage to 60 V and set the display mode to power.

```
VOLT 60  
DISP:MODE 3
```

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

```
VOLT 60::DISP:MODE 3
```

### Including Common Commands

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

```
VOLTage 115;*ESE 255  
OUTPut 0;*RCL 2
```

## 3.3 Using Queries

---

Observe the following precautions with queries:

- Set up the proper number of variables for the returned data.
- Read back all the results of a query before sending another command to the power supply . Otherwise a Query Error will occur and the non-returned data will be lost.

### 3.4 Structure of a SCPI Message

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

#### The Message Unit

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

```
CURRent?<newline>
VOLTag?<newline>
```

The message unit may include a parameter after the header. The parameter usually is numeric:

```
VOLTag 20<newline>
OUTPut 1<newline>
```

#### Combining Message Units

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

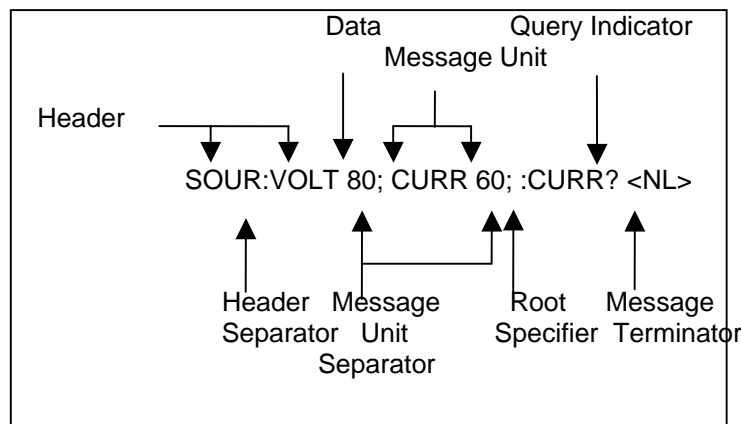


Figure 3-2: Command Message Structure

The basic parts of the above message are:

Message Component	Example
Headers	SOURC VOLT CURR CURR
Header Separator	The colon in SOUR:VOLT
Data	80 60
Data Separator	The space in VOLT 80 and CURR 60
Message Units	VOLT 80 CURR 60 CURR?
Message Unit Separator	The semicolons in VOLT 80; and CURR 60;
Root Specifier	The colon in :CURR?
Query Indicator	The question mark in CURR?
Message Terminator	The <NL> (newline) indicator. Terminators are not part of the SCPI syntax

## Headers

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

Long Form	The header is completely spelled out, such as VOLTAGE, SYSTEM, and OUTPUT. The KDC Parser will stop parsing the remainder of any command after its short form has been parsed. Effectively, this means the remainder will be a don't care and syntax errors will not be detected or reported.
Short Form	The header has only the first three or four letters, such as VOLT, SYST, and OUTP. The KDC parser only parses the short form of all SCPI commands and ignores any remainder.

The SCPI interface is not sensitive to case. It will recognize any case mixture, such as VOLTAGE, VOLTage or Voltage. Short form headers result in faster program execution.

## Header Convention

In the command descriptions in chapter 4 of this manual, headers are emphasized with boldface type. The proper short form is shown in upper-case letters, such as DELay.

Header Separator

If a command has more than one header, you must separate them with a colon (SYSTem:ERRor LIMit:VOLT:LOW).

## Optional Headers

The use of some headers is optional. Optional headers are shown in brackets, such as VOLTage[:LEVel] 100.

## Query Indicator

Following a header with a question mark turns it into a query (VOLTage?).

## Message Unit Separator

When two or more message units are combined into a compound message, separate the units with a semicolon (VOLT 100;CURR 60).

## Root Specifier

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

SOURce:VOLTage:LEVel 100	All colons are header separators
:SOURce:VOLTage:LEVel 100	Only the first colon is a root specifier
SOURce:VOLTage:LEVel 100;:CURRent 55	Only the third colon is a root specifier

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



### Message Terminator

A terminator informs SCPI that it has reached the end of a message. The only permitted message terminator is:

- newline (<NL>), which is ASCII decimal 10 or hex 0A.

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

## 3.5 SCPI Data Formats

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

### Numerical Data Formats

Symbol	Data Form
--------	-----------

#### Talking Formats

<NR1>	Digits with an implied decimal point assumed at the right of the least-significant digit. Examples: 273
<NR2>	Digits with an explicit decimal point. Example: .2.73
<NR3>	Digits with an explicit decimal point and an exponent. Example: 2.73E+2
<Bool>	Boolean Data. Example: 0   1
<BNRD>	Binary Numeric Response Data or Hexadecimal numeric response data.

#### Listening Formats

<Nrf>	Extended format that includes <NR1>, <NR2> and <NR3>. Examples: 273.2 , 2.73E2
<Bool>	Boolean Data. Example: 0   1

### Character Data

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

<CRD>	Character Response Data. Permits the return of character strings.
<AARD>	Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
<SRD>	String Response Data. Returns string parameters enclosed in double quotes.

## 4 SCPI Command Reference

---

Where appropriate, related commands or queries are included. These are listed because they are either directly related by function, or because reading about them will clarify or enhance your understanding of the original command or query.

This chapter is organized as follows:

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

Subsystem commands are specific to Power supply functions. They can be a single command or a group of commands. The groups are comprised of commands that extend one or more levels below the root. The description of common commands follows the description of the subsystem commands.

The subsystem command groups are listed in alphabetical order and the commands within each subsystem are grouped alphabetically under the subsystem. Commands followed by a question mark (?) take only the query form. When commands take both the command and query form, this is noted in the syntax descriptions.

Common commands start with a leading "\*" (star) character and are defined in the IEEE-488.2 standard. The common commands supported by the KDC Series are listed in section 4.11.

### 4.1 Abort Subsystem

---

This subsystem aborts any transient or measurement process in progress.

#### Subsystem Syntax

ABORt                      Resets the trigger system to the Idle state

#### ABORt

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

Command Syntax	ABORt
Parameters	one
Examples	ABOR
Related Commands	INIT   *RST   *TRG

## 4.2 Calibration Subsystem

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

- Enable and disable the calibration mode
- Calibrate the measured current and measured voltage and store new calibration in nonvolatile memory.
- Calibrate the current and voltage output levels, and store new calibration constants in nonvolatile memory.

### Subsystem Syntax

CALibrate	
:MEASure	
:CURRent	
[:AMBient]	Ambient temperature calibrations
[:FSCale]	Calibrate full-scale DC current measurements
:ZERO	Calibrate DC current measurements offset
:TEMPerature	Elevated temperature calibrations
[:FSCale]	Calibrate full-scale DC current measurements at higher temperature
:VOLTage	
[:AMBient]	Ambient temperature calibrations
[:FSCale]	Calibrate full-scale DC voltage measurements
:ZERO	Calibrate DC voltage measurements offset
:TEMPerature	Elevated temperature calibrations
[:FSCale]	Calibrate full-scale DC voltage measurements
[:SOURce]	
:CURRent	
:EXTernal	RPI input calibration
:5V	5V input range
:10V	10V input range
:RESistance	Resistive input range
:INTernal	Internal
:OFFSet	offset calibration
:RANGe	full scale calibration
:VOLTage	
:EXTernal	RPI input calibration
:5V	5V input range
:10V	10V input range
:RESistance	Resistive input range
:INTernal	Internal
:OFFSet	offset calibration
:RANGe	full scale calibration
:PASSword	Allows entry of calibration password required to change calibration coefficients
:SAVE	Saves new or modified calibration coefficients

## 4.2.1 Calibration - Measurement

### Subsystem Syntax

CALibrate		
:MEASure		
:CURRent		
[:AMBient]		Ambient temperature calibrations
[:FSCale]		Calibrate full-scale DC current measurements
:ZERO		Calibrate DC current measurements offset
:TEMPerature		Elevated temperature calibrations
[:FSCale]		Calibrate full-scale DC current measurements at higher temperature
:VOLTage		
[:AMBient]		Ambient temperature calibrations
[:FSCale]		Calibrate full-scale DC voltage measurements
:ZERO		Calibrate DC voltage measurements offset
:TEMPerature		Elevated temperature calibrations
[:FSCale]		Calibrate full-scale DC voltage measurements

#### **CALibrate:MEASure:CURRent[:AMBient][:FSCale] <NRf>**

This command initiates the calibration of the DC current measurement at full scale and at ambient temperature.

Command Syntax	CALibrate:MEASure:CURRent[:AMBient][:FSCale]
Parameters	<NRf> (actual load current measured with external device)
Examples	CAL:MEAS:CURR 50
Query Syntax	CALibrate:MEASure:CURRent?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:CURRent:ZERO

#### **CALibrate:MEASure:CURRent[:AMBient]:ZERO <NRf>**

This command initiates the offset adjustment of the DC current measurement at ambient temperature.

Command Syntax	CALibrate:MEASure:CURRent[:AMBient]:ZERO
Parameters	<NRf> (0 or desired offset value)
Examples	CAL:MEAS:CURR:ZERO
Query Syntax	CALibrate:MEASure:CURRent:ZERO?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:CURRent[:AMB][:FSCale]

#### **CALibrate:MEASure:CURRent:TEMPerature[:FSCale] <NRf>**

This command initiates the calibration of the DC current measurement at elevated temperature.

Command Syntax	CALibrate:MEASure:CURRent:TEMPerature[:FSCale]
Parameters	<NRf> (actual load current measured with external device)
Examples	CAL:MEAS:CURR:TEMP
Query Syntax	CALibrate:MEASure:CURRent:TEMPerature?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:CURRent[:AMB][:FSCale]

**CALibrate:MEASure:VOLTage[:AMBient][:FSCale] <NRf>**

This command initiates the calibration of the DC voltage measurement at full scale and at ambient temperature.

Command Syntax	CALibrate:MEASure:VOLTage[:AMBient][:FSCale]
Parameters	<NRf> (actual DC output voltage measured with external device)
Examples	CAL:MEAS:VOLT 120
Query Syntax	CALibrate:MEASure:VOLTage?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:VOLTage:ZERO

**CALibrate:MEASure:VOLTage[:AMBient]:ZERO <NRf>**

This command initiates the offset adjustment of the DC voltage measurement at ambient temperature.

Command Syntax	CALibrate:MEASure:VOLTage[:AMBient]:ZERO
Parameters	<NRf> (0 or desired offset value)
Examples	CAL:MEAS:VOLT:ZERO 0
Query Syntax	CALibrate:MEASure:VOLT:ZERO?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:VOLTage[:AMBient][:FSCale]

**CALibrate:MEASure:VOLTage:TEMPerature[:FSCale] <NRf>**

This command initiates the calibration of the DC voltage measurement at full scale and at elevated temperature.

Command Syntax	CALibrate:MEASure:VOLTage:TEMPerature[:FSCale]
Parameters	<NRf> (actual DC output voltage measured with external device)
Examples	CAL:MEAS:VOLT:TEMP 120
Query Syntax	CALibrate:MEASure:VOLTage:TEMPerature?
Returned Parameters	<NRf>
Related Commands	CALibrate:MEASure:VOLTage[:AMBient][:FSCale]

## 4.2.2 Calibration - Output

### Subsystem Syntax

```

CALibrate
  [:SOURce]
    :CURRent
      :EXTernal      RPI input calibration
      :5V            5V input range
      :10V           10V input range
      :RESistance    Resistive input range
    :INTernal        Internal
    :OFFSet          offset calibration
    :RANGe           full scale calibration
  :VOLTage
    :EXTernal      RPI input calibration
    :5V            5V input range
    :10V           10V input range
    :RESistance    Resistive input range
    :INTernal        Internal
    :OFFSet          offset calibration
    :RANGe           full scale calibration

```

#### **CALibrate[:SOURce]:CURRent:EXTernal:5V <NRf+>**

This command will set the calibration coefficient for the 0-5V RPC input range.

Command Syntax	CALibrate:SOURce:CURRent:EXTernal:5V <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:EXT:5V -2
Query Syntax	CALibrate:CURRent:EXTernal:5V?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:EXT:5V

#### **CALibrate[:SOURce]:CURRent:EXTernal:10V <NRf+>**

This command will set the calibration coefficient for the 0-10V RPC input range.

Command Syntax	CALibrate:SOURce:CURRent:EXTernal:10V <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:EXT:10V -2
Query Syntax	CALibrate: CURRent:EXTernal:10V?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:EXT:10V

#### **CALibrate[:SOURce]:CURRent:EXTernal:RESistance <NRf+>**

This command will set the calibration coefficient for the RPC resistive input.

Command Syntax	CALibrate:SOURce:CURRent:EXTErnal:RESistance <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:EXT:RES -2
Query Syntax	CALibrate:CURRent:EXTErnal:RESistance?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:EXT:RES

#### **CALibrate[:SOURce]:CURRent:INTernal <NRf+>**

This command will set the calibration coefficient for the internal current range.

Command Syntax	CALibrate:SOURce:CURRent:INTernal <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:INT -2
Query Syntax	CALibrate:CURRent:INTernal?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:INT

#### **CALibrate[:SOURce]:CURRent:OFFSet <NRf+>**

This command will set the calibration coefficient for the dc offset current.

Command Syntax	CALibrate:SOURce:CURRent:OFFSet <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:OFFS -2
Query Syntax	CALibrate:CURRent:OFFSet?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:RAN

#### **CALibrate[:SOURce]:CURRent:RANGe <NRf+>**

This command will set the calibration coefficient for the dc full-scale output current.

Command Syntax	CALibrate:SOURce:CURRent:RANGe <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:CURR:RANG -2
Query Syntax	CALibrate:CURRent:RANGe?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:OFFS

#### **CALibrate[:SOURce]:VOLTage:EXTErnal:5V <NRf+>**

This command will set the calibration coefficient for the 0-5V RPV input range.

Command Syntax	CALibrate:SOURce:VOLTage:EXTErnal:5V <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:EXT:5V -2
Query Syntax	CALibrate:VOLTage:EXTErnal:5V?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:EXT:5V

#### **CALibrate[:SOURce]:VOLTage:EXTErnal:10V <NRf+>**

This command will set the calibration coefficient for the 0-10V RPV input range.

Command Syntax	CALibrate:SOURce: VOLTage:EXTernal:10V <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:EXT:10V -2
Query Syntax	CALibrate:VOLTage:EXTernal:10V?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:EXT:10V

#### **CALibrate[:SOURce]:VOLTage:EXTernal:RESistance <NRf+>**

This command will set the calibration coefficient for the RPV resistive input.

Command Syntax	CALibrate:SOURce:VOLTage:EXTernal:RESistance <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:EXT:RES -2
Query Syntax	CALibrate:VOLTage:EXTernal:RESistance?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:EXT:RES

#### **CALibrate[:SOURce]:VOLTage:INTernal <NRf+>**

This command will set the calibration coefficient for the internal voltage range.

Command Syntax	CALibrate:SOURce:VOLTage:INTernal <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:INT -2
Query Syntax	CALibrate:VOLTage:INTernal?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:CURR:INT

#### **CALibrate[:SOURce]:VOLTage:OFFSet <NRf+>**

This command will set the calibration coefficient for the dc offset current.

Command Syntax	CALibrate:SOURce:VOLTage:OFFSet <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:OFFS -2
Query Syntax	CALibrate:VOLTage:OFFSet?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:RAN

#### **CALibrate[:SOURce]:VOLTage:RANGe <NRf+>**

This command will set the calibration coefficient for the dc full-scale output voltage.

Command Syntax	CALibrate:SOURce:VOLTage:RANGe <NRf+>
Parameters	<NRf> (a value between -127 and +128)
Examples	CAL:VOLT:RANG -2
Query Syntax	CALibrate:VOLTage:RANGe?
Returned Parameters	<NR1> (value range -127 to +128)
Related Commands	CAL:VOLT:OFFS



### 4.2.3 Calibration - Password

#### CALibrate:PASSword

This command allows the entry of the calibration password. The calibration password is required to use the data entry form of the calibration commands. Without the use of this password, only the query form can be used to query any calibration coefficient but no new calibration can be performed. Calibration queries always return two values. The first value is the calibration coefficient itself, the second value is the temperature associated with that coefficient. The DC power supply controller computes all temperature coefficients.

The calibration password is defined as the DC power supply voltage range. Thus a KDC 80-62 model (80 V range) will have a calibration password equal to 80. The password needs to be enclosed by single or double quotation marks. Thus, if the units voltage range is 150 VDC, the calibration password is "150" and the command syntax would be:

CAL:PASS "150"

Command Syntax	CALibrate:PASSword<SRD>
Parameters	<voltage range value> (default)
Examples	CAL:PASS '150'      CAL:PASS "150"
Related Commands	*IDN?

### 4.2.4 Calibration - Save

#### CALibrate:SAVE

This command saves all calibration coefficients to non-volatile memory. This command should be issued after all calibration adjustments have been made. If not, all changes will be lost when unit is turned off and the previous calibration values will take effect the next time the unit is powered up.

**Note:** *Saving calibration data to non-volatile memory requires more time to process by the power supply controller than other commands. As such, it is recommended to hold off on sending additional commands for about 300 msecs*

Command Syntax	CALibrate:SAVE<SRD>
Parameters	None
Examples	CAL:SAVE
Related Commands	CAL:PASS

### 4.3 Initiate Subsystem

This subsystem controls the triggering of the DC power supply. See section 5.6 under "Triggering Output Changes" for an explanation of the Trigger Subsystem. The INITiate commands control the initialization of the transient system. The trigger subsystem must first be enabled using the INITiate commands or no triggering action will occur. Refer to Figure 5-1 for a block diagram of the trigger system.

#### Subsystem Syntax

```
INITiate
  [:IMMEDIATE]           Initiates the system for one trigger
  [:TRANSient]
  :ACQuire
  :CONTinuous           sets if the initiate continuous to (ON|OFF)
```

#### INITiate[:IMMEDIATE][:TRANSient]

The INITiate command controls the initiation of the transient generation or measurement acquisition trigger system. They cause the trigger system to make a transition from the Idle state to the Waiting-for-Trigger state. If the trigger system is not in the Idle state, the initiate commands are ignored.

If both the transient generation and the measurement acquisition trigger systems are initiated at the same time, the first trigger event that occurs will trigger the measurement system. The transient system will trigger on the next trigger event. Thus, the measurement system has a higher trigger priority than the transient system. If the measurement trigger is generated by a transient event, the acquisition system should be initiated first so it is waiting for the trigger caused by the transient system.

Command Syntax	INITiate[:IMMEDIATE][:TRANSient]
Parameters	None
Examples	INIT
Related Commands	ABOR INIT:CONT *TRIG

#### INITiate:ACQuire

The INITiate:ACQuire command controls the initiation of the measurements trigger system. They cause the trigger system to make a measurement from the Idle state to the Waiting-for-Trigger state. If the trigger system is not in the Idle state, the initiate commands are ignored.

Command Syntax	INITiate:ACQuire
Parameters	None
Examples	INIT:ACQ
Related Commands	ABOR INIT:CONT *TRIG

#### INITiate:CONTinuous

These commands control the transient generator trigger system.

- 1 or ON continuously initiates the transient trigger system.
- 0 or OFF turns off continuous triggering. In this state, the trigger system must be initiated for each triggered event using INITiate:TRANSient.

Command Syntax	INITiate:CONTinuous
Parameters	0 1 OFF ON
*RST Value	OFF
Examples	INIT:CONT ON
Related Commands	ABOR INIT:CONT *TRG

## 4.4 Measurement Subsystem

This subsystem programs the measurement capability of the power supply.

### Subsystem Syntax

MEASure/FETch	
:ARRay	
:CURRent[:DC]?	Returns the DC current data points.
:MODE	BINary or ASCii data format
:POWer[:DC]?	Returns the DC power measurement data points.
:VOLTag[:DC]?	Returns the DC voltage measurement data points.
[:SCALar]	
:CURRent	
[:DC]?	Returns the average DC value of current
:AMPLitude	
:MAX?	Returns peak current measurement reading.
:RESet	Resets the peak current measurements.
:POWer[:DC]?	Returns the DC power measurement reading.
:VOLTag[:DC]?	Returns the DC voltage measurement reading.

### Fetch or Measure formats:

All measurement commands are available in either measure or fetch query formats. A measure command will always force new data to be acquired and transferred. A fetch command on the other hand will return the last data available. Sending repeated fetch commands will cause the same result to be returned unless or until a measurement trigger is generated in some way, either as the result of a bus command or a measurement trigger event caused by another source such as the transient system. For example, if it is important to retrieve both voltage and current readings from the same moment in time, use the MEAS form to query the voltage followed by the FETCh form to query the current. This will ensure that both readings are calculated from the same acquisition window (buffer).

Regardless of the measurement format used, it is required to append a ;\*WAI command to each measurement query to ensure the GPIB bus controller is held until the measurement result is available to be received by the application program. If not, subsequent query results may interfere with the response data to the query. Thus, issue MEAS:VOLT?;\*WAI to measure the voltage.

### MEASure:ARRay:CURRent[:DC]?

### FETCh:ARRay:CURRent[:DC]?

These queries return an array containing the instantaneous output current in amperes. The data returned in arbitrary block data format as follows:

```
#5<block length n><b0><b1><b2><b3>.....<bn-3><bn-2><bn-1><bn>
```

where b0,b1,b2,b3 are four hex bytes represent IEEE single precision floating number, where b0 is the most significant byte and b3 is the least significant byte.

The output voltage, current and power are digitized whenever a measure command is given or whenever an acquire trigger occurs. The time interval between samples can range from 20.833 microseconds (48 Ks/s) to 208.33 microseconds (4.8 Ks/s). The query SENSE:SWEp:TINTerval? will return the time interval, the position of the trigger relative to the beginning of the data buffer is determined by SENSE:SWEp:OFFSet.

Query Syntax	MEASure:ARRay:CURRent[:DC]? [<n>,<n>] FETCh:ARRay:CURRent[:DC]? [<n>,<n>]
Parameters	Optional block and offset parameters <n>,<n>. Where the first value <n> is the number of 256 sample blocks to transfer and the second value <n> is the first block (offset) to start with. Number of blocks is from 1 to 16, offset is from 0 to 15.
Examples	MEAS:ARR:CURR?    FETC:ARR:CURR? 4,0
Returned Parameters	<BNRD>            4096 data points in arbitrary block data format
Related Commands	MEAS:ARR:VOLT      SENS:SWE

### MEASure:ARRay:MODE

This command selects the waveform array data format to be used. The default mode is binary (BIN) which uses an IEEE floating point data format in which each data sample is transferred as a 4 byte floating point binary data word. Alternatively, an ASCII format may be selected (ASCIi) in which each data sample is sent as 8 ASCII Hex values representing the 4 byte IEEE floating point data. Note that the transfer mode only applies to measurement array queries. All other measurement queries always return ASCII data. Note that at power on, the default mode is always set to binary (BIN).

Syntax	MEASure:ARRay:MODE
Parameters	BIN   ASCIi
Examples	MEAS:ARR:MOD ASC
Related Commands	MEAS:ARR:VOLT      MEAS:ARR:CURR

**Note:** The MEAS:ARR:MOD command is provided to allow waveform data transfers in ASCII on DBCS versions of MS Windows. Examples of DBCS versions are Chinese, Korean, Japanese etc. On most Windows versions, the binary mode can be used as it reduces the amount of data transferred and thus provides better throughput.

**Note:** KDC units with firmware revision 1.06 through 1.12 have a parser error that requires different parameters to be sent with the MEAS:ARR:MOD command. For units with this firmware revision, use MAST instead of BIN and use APAR instead of ASC.

The ASCII mode will double the number of characters transferred so provisions for a larger receive buffer on the PC may have to be made. In binary mode, the maximum data size for the complete acquisition buffer is 16KB. In ASCII mode, the maximum data size for the complete acquisition buffer is 32KB. The block size and block offset parameters may be used to reduce the amount of data transferred if the entire acquisition buffer is not needed by the end-user or application program.

Acquisition data is formatted in an IEEE floating point format to reduce data size. The following VB6 sample program illustrates how to convert to native floating point data formats.

Conversion function sample VB6. Converting waveform data from either transfer mode to a single precision value can be accomplished using the following sample routine:

```
Public Function StringToIEEEFloat(ByVal sData As String, ByVal bAsciiMode As Boolean) As Single
'=====
'bAsciiMode flag is used if data is received as 8 ascii chars
'representing Hex 0-9,A-F. If bAsciiMode flag is false, then
'data is process as 4 char representing a byte each. Ascii
'mode is needed for DCBS windows
'=====
    Dim i            As Integer
    Dim j            As Integer
    Dim iChar        As Integer
    Dim expo         As Long
    Dim mantisse     As Long
```

```

Dim expo_val As Variant
Dim mant_f As Single
Dim c(3) As Long 'Must use 32 bit integers to allow for
                  'intermediate result of 24 bit shift
Dim sign As Boolean
'=====
Const MANT_MAX = &H7FFFFFFF
Const EXPO_MAX = 2 ^ 126
'=====

On Error GoTo FloatConvError
If bAsciiMode Then
'Retrieve ASC values from eight hex byte input data
sData = UCase(sData)
For i = 0 To 3
c(i) = 0
For j = 0 To 1
iChar = AscB(Mid$(sData, i * 2 + j + 1, 1)) - 48
If iChar > 9 Then iChar = iChar - 7
c(i) = c(i) * 16 * j + iChar
Next j
Next i
Else
'Retrieve ASC values from four byte input data
'Note: Don't use ASCB or ASCW functions as results will differ
'based on character sets, even on non DCBS Windows
'Retrieve ASC values from four byte input data
For i = 0 To 3
c(i) = Asc(Mid$(sData, i + 1, 1))
Next i
End If
'Get sign bit
sign = ((c(0) And &H80) = &H80)
'Get exponent value less sign bit
expo = (c(0) And &H7F) * 2
'Pick up exponent sign
If (c(1) And &H80) = &H80 Then expo = expo Or 1
'get data less exponent sign bit
c(1) = c(1) And &H7F
mantisse = c(1) * &H10000 + c(2) * &H100 + c(3)
mant_f = mantisse / MANT_MAX
'Process exponent
If (expo <> 0) And (expo <> &HFF) Then
expo = expo - 127
mant_f = mant_f + 1
expo_val = 2 ^ Abs(expo)
If (expo > 0) Then mant_f = mant_f * expo_val
If (expo < 0) Then mant_f = mant_f / expo_val
Else
If (mant_f <> 0) Then
If expo = 0 Then
mant_f = mant_f / EXPO_MAX
Else
mant_f = mant_f * EXPO_MAX
End If
End If
End If
'Append number sign and return value
If sign Then mant_f = -mant_f
StringToIEEEFloat = mant_f
Exit Function
'=====

FloatConvError:
'Conversion errors are truncated to zero
StringToIEEEFloat = 0
Exit Function

End Function

```

**MEASure:ARRay:POWer[:DC]?  
FETCh:ARRay:POWer[:DC]?**

These queries return an array containing the instantaneous output power in watts. The data returned in arbitrary block data format as follows:

#5<block length n><b0><b1><b2><b3>.....<bn-3><bn -2><bn-1><bn>

where b0,b1,b2,b3 are four hex bytes represent IEEE single precision floating number, where b0 is the most significant byte and b3 is the least significant byte.

The output voltage, current and power are digitized whenever a measure command is given or whenever an acquire trigger occurs. The time interval between samples can range from 20.833 microseconds to 208.33 microseconds. The query SENSE:SWEp:TINterval? will return the time interval, the position of the trigger relative to the beginning of the data buffer is determined by SENSE:SWEp:OFFSet.

Query Syntax	MEASure:ARRay:POWer[:DC]? [<n>, <n>] FETCh:ARRay:POWer[:DC]? [<n>, <n>]
Parameters	Optional block and offset parameters <n>,<n>. Where the first value <n> is the number of 256 sample blocks to transfer and the second value <n> is the first block (offset) to start with. Number of blocks is from 1 to 16, offset is from 0 to 15.
Examples	MEAS:ARR:VOLT?      FETC:ARR:VOLT?
Returned Parameters	<BNRD>      4096 data points in arbitrary block data format
Related Commands	MEAS:ARR:VOLT      SENS:SWE

**MEASure:ARRay:VOLTAge[:DC]?  
FETCh:ARRay:VOLTAge[:DC]?**

These queries return an array containing the instantaneous output voltage in volts. The data returned in arbitrary block data format as follows:

#5<block length n><b0><b1><b2><b3>.....<bn-3><bn -2><bn-1><bn>

where b0,b1,b2,b3 are four hex bytes represent IEEE single precision floating number, where b0 is the most significant byte and b3 is the least significant byte.

The output voltage, current and power are digitized whenever a measure command is given or whenever an acquire trigger occurs. The time interval between samples can range from 20.833 microseconds to 208.33 microseconds. The query SENSE:SWEp:TINterval? will return the time interval, the position of the trigger relative to the beginning of the data buffer is determined by SENSE:SWEp:OFFSet.

Query Syntax	MEASure:ARRay:VOLTAge[:DC]? [<n>, <n>] FETCh:ARRay:VOLTAge[:DC]? [<n>, <n>]
Parameters	Optional block and offset parameters <n>,<n>. Where the first value <n> is the number of 256 sample blocks to transfer and the second value <n> is the first block (offset) to start with. Number of blocks is from 1 to 16, offset is from 0 to 15.
Examples	MEAS:ARR:VOLT?      FETC:ARR:VOLT?
Returned Parameters	<BNRD>      4096 data points in arbitrary block data format
Related Commands	MEAS:ARR:CURR      SENS:SWE

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

This query returns the DC value of the DC current being sourced at the output terminals. Note that the output must be on to obtain current flow.

Query Syntax	MEASure:CURRent? FETCh[:SCALar]:CURRent[:DC]?
Parameters	None
Examples	MEAS:CURR?
Returned Parameters	<NR2>
Related Commands	OUTP 0   1 MEAS:CURR:AMPL:MAX?

**MEASure[:SCALar]:CURRent:AMPLitude:MAX?  
FETCh[:SCALar]:CURRent:AMPLitude:MAX?**

This query returns the peak current value of the output current being sourced at the output terminals. Note that the output must be ON to obtain current flow. The peak current measurement circuit uses a sample and hold method and latches the highest peak current value found since the last peak measurement reset command.

Query Syntax	MEASure:CURRent:AMPLitude:MAX? FETCh[:SCALar]:CURRent:AMPLitude:MAX?
Parameters	None
Examples	MEAS:CURR:AMPL:MAX?
Returned Parameters	<NR2>
Related Commands	OUTP 0   1 MEAS:CURR? MEAS:CURR:AMPL:RES

**MEASure[:SCALar]:CURRent:AMPLitude:RESet  
FETCh[:SCALar]:CURRent:AMPLitude:RESet**

This command resets the peak current measurement sample and hold circuit to zero.

Query Syntax	MEASure:CURRent:AMPLitude:RESet FETCh[:SCALar]:CURRent:AMPLitude:RESet
Parameters	None
Examples	MEAS:CURR:AMPL:RES
Related Commands	MEAS:CURR:AMPL:MAX?

**MEASure[:SCALar]:POWer[:DC]?  
FETCh[:SCALar]:POWer[:DC]?**

This query returns the DC power delivered to the unit under test by the Power supply. Note that the output must be on to obtain current flow.

Query Syntax	MEASure:POWer? FETCh[:SCALar]:POWer[:DC]?
Parameters	None
Examples	MEAS:POW?
Returned Parameters	<NR2>
Related Commands	MEAS:VOLT? MEAS:CURR?

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

This query returns the average DC voltage being sourced at the output terminals. Note that the output must be on or no voltage will be present at the output terminals.



Query Syntax	MEASure:VOLTage? FETCh[:SCALar]:VOLTage[:DC]?
Parameters	None
Examples	MEAS:VOLT?
Returned Parameters	<NR2>
Related Commands	OUTP 0   1

## 4.5 Output Subsystem

This subsystem controls the main outputs, the signal outputs and the output impedance of the DC supply.

### Subsystem Syntax

OUTPut		
:IMPedance		
[:STATe]	Enable/disable output impedance program	
:REAL	Sets resistive part of output impedance	
RI[:LEVe]	Sets Remote Inhibit input level mode (LOW   HIGH).	
[:STATe]	Enable/disable output voltage, current, power, etc.	
:TTLTrg		
[:STATe]	Enable/disable trigger out drive	
:SOURce	Selects a TTLTrg source (BOT EOT LIST)	

### OUTPut:IMPedance[:STATe]

This command enables or disables the supply's output impedance programming. The state of a disabled output impedance is equivalent to the minimum impedance. The query form returns the output state.

Command Syntax	OUTPut:IMPdance[:STATe]<bool>
Parameters	0   OFF   1   ON
*RST Value	OFF
Examples	OUTP:IMP 1   OUTP:IMP:STAT ON
Query Syntax	OUTPut:IMPdance[:STATe]?
Returned Parameters	0   1
Related Commands	OUTPut:IMPedance:REAL

### OUTPut:IMPedance:REAL

This command sets the real part of the output impedance of the AC source in milli-ohms. OUTPut:IMPedance:STATe must be enabled for the programmed impedance to affect the output. The available range is a function of voltage and current range of the DC supply and can be queried using the "OUTP:IMP:REAL? MIN" and "OUTP:IMP:REAL? MAX" commands

Command Syntax	OUTP:IMPedance:REAL<NRf>
Parameters	model specific.
*RST Value	min
Examples	OUTP:IMP:REAL 20
Query Syntax	OUTPut:IMP:REAL?
Returned Parameters	<NR2>
Related Commands	OUTP:IMP

**OUTPut:RI[:LEVel]**

This command sets the remote inhibit level mode. Factory default is LOW, which requires a contact closure to open the output relay. The level can be reversed by setting it to HIGH. Once set, the setup including the new RI level must be saved to Register 0 and REG0 must be set as the power on setup in order for the power supply to always power up with this RI state. If the power on state is set to \*RST (factory default), the RI state will default to LOW.. Note that this command is only implement with firmware revision 2.00 or higher. Lower firmware revisions only provide the default LOW setting.

Command Syntax	OUTPut:RI[:LEVel]
Parameters	LOW   HIGH
*RST Value	OFF
Examples	OUTP:RI HIGH
Query Syntax	OUTP:RI?
Returned Parameters	<CRD>
Related Commands	OUTP

**OUTPut[:STATe]**

This command enables or disables the supply output. The state of a disabled output is an output voltage amplitude set to 0 volts and the output stage turned off. The query form returns the output state.

Command Syntax	OUTPut[:STATe]<bool>
Parameters	0   OFF   1   ON
*RST Value	OFF
Examples	OUTP 1            OUTP:STAT ON
Query Syntax	OUTPut[:STATe]?
Returned Parameters	0   1
Related Commands	*RCL   *SAV

**OUTPut:TTLTrg[:STATe]**

This command enables or disables the Trigger Out (FUNCTION STROBE) signal, which is available at the rear panel of the DC Supply. Refer to the User Manual P/N 5005-960 for connector location, pin out information and signal levels for the function strobe signal.

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

**OUTPut:TTLTrg:SOURce**

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

- BOT Beginning of transient output
- EOT End of transient output
- LIST Specified by the TTLTrg list

When an event becomes true at the selected TTLTrg source, a pulse is sent to the function strobe output on the rear panel of the DC Supply.

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

## 4.6 Sense Subsystem

This subsystem controls the measurement acquisition system.

### Subsystem Syntax

SENSe  
 :SWEep  
   :OFFSet       Define trigger points relative to the start of the digitizer data record  
   :TINTerval?   Queries the digitizer sample interval

### SENSe:SWEep:OFFSet

This command defines the trigger point relative to the start of the returned data record when an acquire trigger is used. The values can range from -4096 times the sample interval (pre-trigger) to 1000 msec (post trigger).

When the value specified is negative (less than 0 msec), the values in the beginning of the data record represent samples taken prior to the actual trigger moment.

Command Syntax	SENSe:SWEep:OFFSet <NRf+>
Parameters	time in msec
*RST Value	0
Examples	SENS:SWE:OFFS -5
Query Syntax	SENSe:SWEep:OFFSet?
Returned Parameters	<NR2>
Related Commands	SENS:SWE:TINT?       MEAS:ARR

### SENSe:SWEep:TINTerval

This command sets the time period between samples when voltage and current digitization is controlled by the acquisition trigger sequence. Allowable sample interval settings range from 20.833  $\mu$ sec to 208.33  $\mu$ sec. The acquired time interval equals the buffer size (4096) times the sample interval. (85.33 msec to 853.33 msec).

The sampling interval can be programmed by sending a value from 20.833 to 208.33. Values that are not a multiple of 20.833 will be adjusted to the nearest multiple by the power supply. To determine the actual interval set, the query format can be used to query back the setting.

Command Syntax	SENSe:SWEep:TINTerval?
Parameters	20.833 to 208.33 in multiples of 20.833
*RST Value	0
Examples	SENS:SWE:OFFS -5
Query Syntax	SENSe:SWEep:OFFSet?
Returned Parameters	<NR2>
Related Commands	SENS:SWE:OFFS       MEAS:ARR

## 4.7 Source Subsystem

This subsystem programs all the output parameters of the power supply. It is the most elaborate of all sub systems and is broken down into separate paragraphs for clarity. The top level source sub system commands are shown here.

### Subsystem Syntax

[:SOURce]	
:ALC	Auto level control mode.
:CURRent	Program current setting
:LIMit	Query range limits
:LIST	Transient List system
:MODE	CVOLTage   CCURrent   CPOWer
:POWer	Programs the Constant Power mode level
:PULSE	Transient Pulse system
:VOLTage	Program voltage setting

The following paragraphs show the completed hierarchies of the various source sub system commands.

### 4.7.1 Source – ALC subsystem

#### [SOURce:]ALC

This command is used to enable or disable the auto level control mode and applies only if the KDC is equipped with the –FT option. The presence of the –FT option can be queried using the \*OPT? or the SYST:CONF? query command.

When enabled, the ALC mode adjusts the output voltage as needed to maintain the programmed voltage setting. This improves the load regulation for –FT models. On non-FT models (standard KDC models), there is no benefit from this ALC mode and setting the ALC mode on has no effect (don't care).

Note that the ALC mode will automatically be disabled when the KDC enters CC or CP mode. The ALC does not generate any faults if regulation cannot be obtained.

Command Syntax	[SOURce:]ALC <param>
Parameters	ON   OFF
Example:	ALC ON
Query Syntax	ALC?
Returned Parameters	<CRD>
Related Commands	VOLT:MODE

### 4.7.2 Source - Current subsystem

[:SOURce]	
:CURRent	
[:LEVel]	
[:IMMediate]	
[:AMPLitude]	
[:DC]	Set the current value in amps. 0 to range
:HIGH	Set the maximum current allowed. 0 to range
:LOW	Set the minimum current allowed. 0 to range
:MODE	FIXed   STEP   PULSe   LIST
:PROTection	
:DELay	0.1 to 5 sec
:STATE	n/a
:REFerence:	INTernal   EXT5V   EXT10V   EXTRES
:SLEW	
[:IMMediate]	Current slew rate in A/sec
:MODE	FIXed   STEP   PULSe   LIST
:TRIGger	0 to range
:TRIGgered	
[:AMPLitude]	0 to range

The source current sub system is used to set the programmable current capability of the DC Supply.

#### **[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude][:DC]**

This command sets the current limit of the output. If the load current exceeds the programmed current limit, the power supply will respond based on the setting of the PROT:MODE.

The current may be set between the low limit and high limit as programmed using the CURR:HIGH and CURR:LOW commands. However, the high limit cannot be higher than the model's capability (configuration limit) which can be queried with the LIM:CURR? command. The low limit cannot be less than zero. The \*RST state for high and low current is zero for low, and maximum configuration limit for high. (model specific).

Command Syntax	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude][:DC] <NRf>
Parameters	value between CURR:HIGH and CURR:LOW settings.
Unit	A (amperes)
*RST	Defined by the PON
Examples	CURR 5
Query Syntax	CURRent?
Returned Parameters	<NR2>
Related Commands	CURR:HIGH    CURR:LOW

#### **[SOURce:]CURRent[:LEVel][:IMMediate]:HIGH**

This command sets the maximum programmed current that will be accepted by the power supply. If a value above this limit value is programmed, a "-222 Data out of Range" error will be generated.

The current limit may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:CURR? command. The high limit cannot be less than the low limit however. The \*RST state for high current limit is the maximum configuration limit. (model specific).

Command Syntax	[SOURce:]CURRent:HIGH <NRf>
Parameters	0 to LIM:CURR? value.
Unit	A (amperes)
*RST	Max. Current
Examples	CURR:HIGH 5
Query Syntax	CURRent:HIGH?
Returned Parameters	<NR2>
Related Commands	CURR                      CURR:LOW

**[SOURce:]CURRent[:LEVel][:IMMediate]:LOW**

This command sets the minimum programmed current that will be accepted by the power supply. If a value below this limit value is programmed, a "-222 Data out of Range" error will be generated.

The current limit may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:CURR? command. The low limit cannot be more than the high limit however. The \*RST state for low current limit is zero.

Command Syntax	[SOURce:]CURRent:LOW <NRf>
Parameters	0 to LIM:CURR? value.
Unit	A (amperes)
*RST	0
Examples	CURR:LOW 5
Query Syntax	CURRent:LOW?
Returned Parameters	<NR2>
Related Commands	CURR                      CURR:HIGH

**[SOURce:]CURRent:MODE**

This command sets the current programming mode. Available modes are:

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

The current limit may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:CURR? command. The low limit cannot be more than the high limit however. The \*RST state for low current limit is zero.

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

**[SOURce:]CURRent:PROTection:DELay**

This command sets the programmable current limit delay when in CV mode of operation. The delay determines for how long the shutdown of the output will be delayed. The programmable range is from 0.1 (100 msec) to 5 secs.



Command Syntax	[SOURce:]CURRent:PROTection:DElAY <NRf>
Parameters	0.1 to 5
*RST Value	0.1
Examples	CURR:PROT:DEL 1.00
Query Syntax	[SOURce:]CURRent:PROT:DEL?
Returned Parameters	<NR2>
Related Commands	SOUR:MODE                      CURR

#### [SOURce:]CURRent:PROTection:STATE

This command sets the programmable current limit state. It is presently not supported and has no effect when issued.

Command Syntax	[SOURce:]CURRent:PROTection:STATE
Parameters	none
*RST Value	
Examples	CURR:PROT:STATE
Query Syntax	[SOURce:]CURRent:PROT:STATE?
Returned Parameters	<NR1>
Related Commands	SOUR:CURR:PROT:DEL              CURR

#### [SOURce:]CURRent:REfERENCE

This command sets the current programming reference. Available references are:

INTernal            The current level is programmed by the internal controller (default).  
EXT5V                The current level is determined by an external 0 - 5 V DC input signal.  
EXT10V              The current level is determined by an external 0 - 10 V DC input signal.  
EXTRES              The current level is determined by an external 0 - 10 KOhm resistor.  
For modes other than INTernal, a suitable input signal must be present at the analog I/O connector. See KDC Series User Manual P/N 5005-960.

Command Syntax	[SOURce:]CURRent:REfERENCE <param>
Parameters	INTernal   EXT5V   EXT10V   EXTRES
*RST Value	INTernal
Examples	CURR:REF INT                      CURR:REF EXT10V
Query Syntax	[SOURce:]CURRent:REF?
Returned Parameters	<CRD>
Related Commands	VOLT:REF                              CURR

#### [SOURce:]CURRent:SLEW[:IMMediate]

This command sets the current slew rate. The current slew rate determines the rate of change when a new current level is programmed. The maximum slew rate that can be programmed is a function of the available current range and will differ from model to model. The MAX parameter may be used to set or query the maximum slew rate.

Command Syntax	[SOURce:]CURRent:SLEW[:IMMediate] <NRf>
Parameters	0 to MAX
*RST Value	MAX
Examples	CURR:SLEW 100                      CURR:SLEW MAX
Query Syntax	[SOURce:]CURRent:SLEW? MAX              CURR:SLEW?
Returned Parameters	<NR2>
Related Commands	CURR:TRIG                              CURR

#### [SOURce:]CURRent:SLEW:MODE

This command sets the current slew rate programming mode. Available modes are:

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

The current slew may be set between zero and the model's maximum current slew capability (determined by model) which can be queried with the CURR:SLEW? MAX command.

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

**[SOURce:]CURRent:SLEW:TRIGgered**

This command sets the current slew rate trigger level. When triggered by a step or pulse transient, the current slew rate will be set to the programmed level. Note that the new slew rate does not take effect until the unit is armed and a trigger event occurs. A parameter of MAXimum will set the slew to its maximum possible rate.

The current slew may be set between zero and the model's maximum current slew capability (determined by model) which can be queried with the CURR:SLEW? MAX command.

Command Syntax	[SOURce:]CURRent:SLEW:TRIGgered <NRf>
Parameters	0 to MAX
*RST Value	0
Examples	CURR:SLEW:TRIG
Query Syntax	[SOURce:]CURRent:SLEW:TRIG?
Returned Parameters	<NR2>
Related Commands	CURR:SLEW:MODE    CURR:SLEW

**[SOURce:]CURRent:TRIGgered[:AMPLitude]**

This command sets the current trigger level. When triggered by a step or pulse transient, the current will be set to the programmed level. Note that the new current setting does not take effect until the unit is armed and a trigger event occurs. The maximum current that can be programmed is a function of the model and may be queried using the LIMIT subsystem. It is also further determined by the user defined CURR:HIGH and CURR:LOW settings.

The current slew may be set between zero and the model's maximum current slew capability (determined by model) which can be queried with the CURR:SLEW? MAX command.

Command Syntax	[SOURce:]CURRent:TRIGgered[:AMPLitude <NRf>
Parameters	0 to MAX
*RST Value	0
Examples	CURR:TRIG
Query Syntax	[SOURce:]CURRent: TRIG?
Returned Parameters	<NR2>
Related Commands	CURR:MODE    CURR

### 4.7.3 Source - Limit subsystem

```
[:SOURce]
  :LIMit
    :CURRent?           Current range limit: Model specific
    :VOLTage?          Voltage range limit. Model specific
```

The Source Limit sub-system may be used to query the maximum capabilities of the DC Supply.

#### LIMit:CURRent

This command will return the maximum current the power supply can deliver to the load. Only the query format of this command is available.

Command Syntax	[SOURce:]LIMit:CURRent <NRf>	[command protected]
Parameters	maximum current	
Example:	LIM:VOLT?	
Query Syntax	[SOURce:]LIMit:CURRent?	
Returned Parameters	<NR2>	
Related Commands	LIM:VOLT?	

#### LIMit:VOLTage

This command will return the maximum voltage of the power supply. Only the query format of this command is available.

Command Syntax	[SOURce:]LIMit:VOLTage <NRf>	[command protected]
Parameters	voltage range]	
Example:	LIM:VOLT?	
Query Syntax	[SOURce:]LIMit:VOLTage?	
Returned Parameters	<NR2>	
Related Commands	LIM:CURR?	

#### 4.7.4 Source - List subsystem

```

[:SOURce]
  :LIST
    :COUNT          Number of times to repeat transient list
    :CURRent
      [:LEVEL]
        [:AMPLitude]
        :POINTs
    :SLEW
      [:RATE]        0.0001 to 1e6 / sec
      :POINTs
    :DWELl
      [:TIME]        Dwell time in seconds
      :POINTs        Dwell data point value
    :REPeat
      [:TIME]        Number of time to repeat list data point
      :POINTs        Repeat data point value
    :STEP             ONCE | AUTO
    :TTLTrig
      [:SWITCh]      0.0001 to 1e6 / sec
      :POINTs
    :VOLTage
      [:LEVEL]
        [:AMPLitude]
        :POINTs
    :SLEW
      [:RATE]        0.0001 to 1e6 / sec
      :POINTs

```

The source List sub-system is used to set the programmable current capability of the DC Supply. Note that the number of maximum points for all lists is 99.

##### LIST:COUNT

This command sets the number of times that the list is executed before it is completed. The command accepts parameters in the range 1 through 2E8.

Command Syntax	[SOURce:]LIST:COUNT<NRf+>
Parameters	1 to 2E8   MINimum   MAXimum
*RST Value	1
Examples	LIST:COUN 3 LIST:COUN MAX
Query Syntax	[SOURce:]LIST:COUNT?
Returned Parameters	<NRf>
Related Commands	LIST:FREQ LIST:TTLT LIST:VOLT

##### LIST:CURRent[:LEVel][:AMPLitude]

This command specifies the output current points in a list. The current points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which the list will be output when a list transient is triggered.

Command Syntax	[SOURce:]LIST:CURRent[:LEVel][:AMPLitude] <NRf+>,<NRf+>
Parameters	0 to MAX
Unit	Volt
Examples	LIST:CURR 2.0,2.5,3.0
Query Syntax	[SOURce:]LIST:CURRent[:LEVel]?
Returned Parameters	<NR2>
Related Commands	LIST:CURR:POIN?      LIST:COUN

### LIST:CURRent[:LEVel]:POINts?

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

Query Syntax	[SOURce:]LIST:CURRent:POINts?
Returned Parameters	<NR1>
Example	LIST:CURR:POIN?
Related Commands	LIST:CURR

### LIST:CURRent:SLEW[:RATE]

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

Command Syntax	[SOURce:]LIST:CURRent:SLEW[:RATE] <NRf+>,<NRf+>
Parameters	0.1 to 1E9   MAX
Unit	A/s (Amps per second)
Example	LIST:CURR:SLEW 10, 1E2, MAX
Query Syntax	[SOURce:]LIST:CURRent:SLEW?
Returned Parameters	<NR2>
Related Commands	LIST:CURR:SLEW:POIN?      LIST:COUN LIST:DWEL      LIST:STEP

### LIST:CURRent:SLEW:POINts?

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

Query Syntax	[SOURce:]LIST:CURRent:SLEW:POINts?
Returned Parameters	<NR1>
Example	LIST:CURR:SLEW:POIN?
Related Commands	LIST:CURR:SLEW

### LIST:DWELI[:TIME]

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

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

The order in which the points are entered determines the sequence in which they are output when a list is triggered.

Command Syntax	[SOURce:]LIST:DWELI[:TIME]<NRf+> ,<NRf+>
Parameters	0.001 to 9E4 MINimum MAXimum
Unit	S (seconds)
Examples	LIST:DWEL .1, .5, 1.5
Query Syntax	[SOURce:]LIST:DWEL?
Returned Parameters	<NR2>
Related Commands	LIST:TTLT      LIST:VOLT

### LIST:DWELI:POINTs?

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

Query Syntax	[SOURce:]LIST:DWELI:POINTs?
Returned Parameters	<NR1>
Example	LIST:DWEL:POIN?
Related Commands	LIST:DWELI

### LIST:REPeat[:TIME]

This command sets the sequence of repeat values for each data list point. The repeat values are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which they are repeated when a list is triggered.

Command Syntax	[SOURce:]LIST:REPeat[:TIME] <NRf+>,<NRf+>
Parameters	0 to 99
Examples	LIST:REPeat 1,0,5
Query Syntax	[SOURce:]LIST:REPeat?
Returned Parameters	<NR1>
Related Commands	LIST:PHAS:POIN?      LIST:COUN LIST:DWEL      LIST:STEP

### LIST:REPeat:POINTs?

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

Query Syntax	SOURce:]LIST:PHASe:POINTs?
Returned Parameters	<NR1>
Example	LIST:PHAS:POIN?
Related Commands	LIST:FREQ      LIST:DWEL

### LIST:STEP

This command specifies how the list sequencing responds to triggers.

- ONCE causes the list to advance only one point after each trigger. Triggers that arrive during a dwell delay are ignored.
- AUTO causes the entire list to be output sequentially after the starting trigger, paced by its dwell delays. As each dwell delay elapses, the next point is immediately output.

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

### LIST:TTLTrg[:SWITCh]

This command sets the sequence of Trigger Out list points. Each point set to “1” will cause a pulse to be output at Trigger Out (Function strobe) output when that list step is reached. Those entries which are set “0” will not generate Trigger Out pulses. The order in which the list points are given determines the sequence in which Trigger Out pulses will be output when a list transient is triggered.

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

### LIST:TTLTrg:POINts?

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

Query Syntax	[SOURce:]LIST:TTLTrg:POINts?
Returned Parameters	<NR1>
Example	LIST:TTLT:POIN?
Related Commands	LIST:TTLT

### LIST:VOLTage[:LEVel][:AMPLitude]

This command specifies the output voltage points in a list. The voltage points are given in the command parameters, which are separated by commas. The order in which the points are entered determines the sequence in which the list will be output when a list transient is triggered.

Command Syntax	[SOURce:]LIST:VOLTage[:LEVel][:AMPLitude] <NRf+>,<NRf+>
Parameters	0 to MAX
Unit	Volt
Examples	LIST:VOLT 2.0,2.5,3.0
Query Syntax	[SOURce:]LIST:VOLTage[:LEVel]?
Returned Parameters	<NR2>
Related Commands	LIST:VOLT:POIN?    LIST:COUN    LIST:DWELLIST:STEP

### LIST:VOLTage[:LEVel]:POINts?

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

Query Syntax	[SOURce:]LIST:VOLTage:POINts?
Returned Parameters	<NR1>
Example	LIST:VOLT:POIN?
Related Commands	LIST:VOLT

### LIST:VOLTage:SLEW[:RATE]

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

Command Syntax	[SOURce:]LIST:VOLTage:SLEW[:RATE] <NRf+>,<NRf+>
Parameters	0.1 to 1E9   MAX
Unit	V/S (volts per second)
Example	LIST:VOLT:SLEW 10, 1E2, MAX
Query Syntax	[SOURce:]LIST:VOLTage:SLEW?
Returned Parameters	<NR2>
Related Commands	LIST:VOLT:SLEW:POIN?      LIST:COUN LIST:DWEL      LIST:STEP

### LIST:VOLTage:SLEW:POINts?

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

Query Syntax	[SOURce:]LIST:VOLTage:SLEW:POINts?
Returned Parameters	<NR1>
Example	LIST:VOLT:SLEW:POIN?
Related Commands	LIST:VOLT:SLEW



#### 4.7.5 Source – Mode subsystem

```
[:SOURce]
:MODE          CVOLtage | CCURrent | CPOWER
```

##### [SOURce:]MODE

This command is used to set or query the power supply's operating mode. The power supply can operate in either Constant Voltage or Constant Current mode. A <CRD> string is returned for a MODE? Query as shown in the table below:

<CRD>	Mode
CVOL	Constant Voltage (CV)
CCUR	Constant Current (CC)
CPOW	Constant Power (CP)

Table 4-1: Mode query results

Command Syntax	[SOURce:]MODE <CRD>
Parameters	CVOLtage   CCURrent   CPOWER
Example:	MODE CVOL
Query Syntax	MODE?
Returned Parameters	<CRD>
Related Commands	VOLT:PROT:DEL

#### 4.7.6 Source – Power subsystem

```
[:SOURce]
:POWER
```

##### [SOURce:]POWER

This command is used to set or query the constant power mode level. This command is only relevant if the MODE is set to constant power mode. In the CC or CV modes, this setting has no effect. The power level set cannot exceed the power level of a single KDC unit, either 5000, 10000 or 15000 depending on model configuration. Also make sure the voltage and current settings are consistent with the desired power level in CP mode.

Command Syntax	[SOURce:]POWER <NRf>
Parameters	<NRf>
Example:	POW 3400
Query Syntax	POW?
Returned Parameters	<NR2>
Related Commands	VOLT:MODE

#### 4.7.7 Source - Pulse subsystem

[:SOURce]	
:PULSE	
:COUNt	1 to 2e8
:DCYClE	0 to 100% pulse width must be >= 0.001
:HOLD	WIDTh   DCYClE
:PERiod	0.002 to 90,000
:WIDth	0.001 to < 90,000

The source pulse sub-system is used to program the pulsed output mode of operation of the DC Supply.

##### **PULSe:COUNt**

This command sets the number of pulses that are output when a triggered output transient occurs. The command accepts parameters in the range 1 through 2E8. If MAXimum is sent, the output pulse repeats indefinitely.

Command Syntax	[SOURce:]PULSe:COUNt<NRf+>
Parameters	1 to 2E8   MINimum   MAXimum
*RST Value	1
Examples	PULS:COUN 3 PULS:COUN MIN          PULS:COUN MAX
Query Syntax	[SOURce:]PULS:COUNt?
Returned Parameters	<NR2>
Related Commands	PULS:DCYC    PULS:PER    PULS:WIDT

##### **PULSe:DCYClE**

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

Command Syntax	[SOURce:]PULSe:DCYClE<NRf+>
Parameters	0 to 100% MINimum MAXimum
*RST Value	50%
Examples	PULS:DCYC 75 PULS:DCYC MAX
Query Syntax	[SOURce:]PULSe:DCYClE?
Returned Parameters	<NR2>
Related Commands	PULS:COUN    PULS:PER    PULS:WIDT

##### **PULSe:HOLD**

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

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

Parameter Set			Action
DCYClE	PERiod	WIDTh	
		√	Sets WIDTh. If WIDTh < PERiod, recalculates DCYClE; otherwise, recalculates the PERiod and DCYClE.
	√		Sets PERiod. If WIDTh < PERiod, recalculates DCYClE; otherwise, recalculates the PERiod and DCYClE.
√			Sets DCYClE and recalculates PERiod

Table 4-1: PULSe:HOLD = WIDTh parameters

Parameter Set			Action
DCYClE	PERiod	WIDTh	
		√	Sets WIDTh and recalculates the PERiod
	√		Sets PERiod and recalculates the WIDTh
√			Sets DCYClE and recalculates PERiod

Table 4-2: PULSe:HOLD = DCYClE parameters

### PULSe:PERiod

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

Command Syntax	[SOURce:]PULSe:PERiod<NRf+>
Parameters	2 msecs to 90,000 secs   MINimum   MAXimum
Unit	S (seconds)
*RST Value	1 sec
Examples	PER 0.001PER MIN
Query Syntax	[SOURce:]PERiod?
Returned Parameters	<NR2>
Related Commands	PULS:COUN PULS:DCYC PULS:WIDT

### PULSe:WIDTh

This command sets the width of a transient output pulse. The command parameters are model-dependent.

Command Syntax	[SOURce:]PULSe:WIDTh<NRf+>
Parameters	1 msecs to 90,000 secs   MINimum   MAXimum
Unit	S (seconds)
*RST Value	0.5 secs
Examples	PULS:WIDT 0.001PULS:WIDT MIN
Query Syntax	[SOURce:]PULSe:WIDTh?
Returned Parameters	<NR2>
Related Commands	PULS:COUN PULS:DCYC PULS:PER

#### 4.7.8 Source - Voltage subsystem

[:SOURce]	
:VOLTage	
[:LEVel]	
[:IMMediate]	
[:AMPlitude]	
[:DC]	Set the voltage value in volts. 0 to range
:HIGH	Set the maximum voltage allowed. 0 to range
:LOW	Set the minimum voltage allowed. 0 to range
:MODE	FIXed   STEP   PULSe   LIST
:PROTEction	
[:LEVel]	0 to 110%
:DELay	0.1 to 5 sec
:MODE	RANGe   SVOLTage   ABSolute
:REFerence:	INTernal   EXT5V   EXT10V   EXTRES
:SLEW	
[:IMMediate]	Voltage slew rate in V/sec
:MODE	FIXed   STEP   PULSe   LIST
:TRIGger	0 to range
:TRIGgered	
[:AMPlitude]	0 to range

The source voltage sub-system is used to set the programmable voltage capability of the DC Supply.

#### **[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude][:DC]**

This command programs the DC output voltage level of the power supply.

The voltage may be set between the low limit and high limit as programmed using the VOLT:HIGH and VOLT:LOW commands. However, the high limit cannot be higher than the model's capability (configuration limit) which can be queried with the LIM:VOLT? command. The low limit cannot be less than zero. The \*RST state for high and low voltage is zero for low, and maximum configuration limit voltage for high. (model specific).

Command Syntax	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPlitude][:DC]<NRf>
Parameters	value between VOLT:HIGH and VOLT:LOW settings.
Unit	V (Volt)
*RST Value	0 volt
Examples	VOLT 250      VOLT:LEV 25
Query Syntax	[SOURce:]VOLTage[:LEVel]?
Returned Parameters	<NR2>
Related Commands	VOLT:HIGH    VOLT:LOW

#### **[SOURce:]VOLTage[:LEVel][:IMMediate]:HIGH**

This command sets the maximum programmed voltage that will be accepted by the power supply. If a value above this limit value is programmed, a "-222 Data out of Range" error will be generated.

The voltage limit may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:VOLT? command. The high limit cannot be less than the low limit however. The \*RST state for high voltage limit is the maximum configuration limit. (model specific).

Command Syntax	[SOURce:]VOLTage[:LEVel][:IMMEDIATE]:HIGH <NRf>
Parameters	0 to LIM:VOLT? value.
Unit	V (Volt)
*RST	Max. Voltage
Examples	VOLT:HIGH 5
Query Syntax	VOLTage:HIGH?
Returned Parameters	<NR2>
Related Commands	VOLT                    VOLT:LOW

### **[SOURce:]VOLTage[:LEVel][:IMMEDIATE]:LOW**

This command sets the minimum programmed current that will be accepted by the power supply. If a value below this limit value is programmed, a "-222 Data out of Range" error will be generated.

The voltage limit may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:VOLT? command. The low limit cannot be more than the high limit however. The \*RST state for low voltage limit is zero.

Command Syntax	[SOURce:]VOLTage[:LEVel][:IMMEDIATE]:LOW <NRf>
Parameters	0 to LIM:VOLT? value.
Unit	V (Volt)
*RST	0
Examples	VOLT:LOW 5
Query Syntax	VOLTage:LOW?
Returned Parameters	<NR2>
Related Commands	VOLT                    VOLT:HIGH

### **[SOURce:]VOLTage:MODE**

This command sets the voltage programming mode. Available modes are:

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

The voltage may be set between zero and the model's capability (configuration limit) which can be queried with the LIM:VOLT? command. The low limit cannot be more than the high limit however. The \*RST state for low voltage limit is zero.

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

### **[SOURce:]VOLTage:PROTection[:LEVel]**

This command sets the programmable voltage protection level. If the output voltage exceeds this set level, a fault will be generated. The level set is expressed as either a percentage of programmed value (SVOLTage), a percentage of the voltage range (RANGe) or an absolute value (ABSolute). This is determined by the VOLT:PROT:MODE command.

Command Syntax	[SOURce:]VOLTage:PROTection[:LEVel] <NRf>
Parameters	0 to 110%   105 to 110%   <NR3>
*RST Value	0.1
Examples	VOLT:PROT 105
Query Syntax	[SOURce:]VOLTage:PROT[:LEVel]?
Returned Parameters	<NR2>
Related Commands	SOUR:VOLT:PROT:MODE      SOUR:VOLT:PROT:DEL

#### [SOURce:]VOLTage:PROTection:DELay

This command sets the programmable current limit delay when in CV mode of operation. The delay determines for how long the shutdown of the output will be delayed. The programmable range is from 0.1 (100 msec) to 5 secs.

Command Syntax	[SOURce:]VOLTage:PROTection:DELay <NRf>
Parameters	0.1 to 5
*RST Value	0.1
Examples	VOLT:PROT:DEL 1.00
Query Syntax	[SOURce:]VOLTage:PROT:DEL?
Returned Parameters	<NR2>
Related Commands	SOUR:MODE                  CURR

#### [SOURce:]VOLTage:PROTection:MODE

This command sets the programmable voltage limit mode. Available options are SVOLTage for percent of programmed voltage, RANGE for percent of voltage range or ABSolute for an absolute value. The mode affects the meaning of the value set with the VOLT:PROT:LEV command.

Command Syntax	[SOURce:]VOLTage:PROTection:MODE
Parameters	SVOLTage   RANGE   ABSolute
*RST Value	SVOLT
Examples	VOLT:PROT:MODE SVOLT
Query Syntax	[SOURce:]VOLTage:PROT:MODE?
Returned Parameters	<CRD>
Related Commands	SOUR:CURR:PROT      CURR

#### [SOURce:]VOLTage:REFerence

This command sets the voltage programming reference. Available references are:

- INTernal      The voltage level is programmed by the internal controller (default).
- EXT5V        The voltage level is determined by an external 0 - 5 V DC input signal.
- EXT10V      The voltage level is determined by an external 0 - 10 V DC input signal.
- EXTRES      The voltage level is determined by an external 0 - 10 KOhm resistor.

For modes other than INTernal, a suitable input signal must be present at the analog I/O connector. See KDC Series User Manual, P/N 5005-960.

Command Syntax	[SOURce:]VOLTage:REFerence <param>
Parameters	INTernal   EXT5V   EXT10V   EXTRES
*RST Value	INTernal
Examples	VOLT:REF INT            VOLT:REF EXT10V
Query Syntax	[SOURce:]VOLTage:REF?
Returned Parameters	<CRD>
Related Commands	CURR:REF            VOLT

#### [SOURce:]VOLTage:SLEW[:IMMediate]

This command sets the voltage slew rate. The voltage slew rate determines the rate of change when a new voltage level is programmed. The maximum slew rate that can be programmed is a function of the available voltage range and will differ from model to model. The MAX parameter may be used to set or query the maximum slew rate.

Command Syntax	[SOURce:]VOLTage:SLEW[:IMMediate] <NRf>
Parameters	0 to MAX
*RST Value	MAX
Examples	VOLT:SLEW 100            VOLT:SLEW MAX
Query Syntax	[SOURce:]VOLTage:SLEW? MAX            VOLT:SLEW?
Returned Parameters	<NR2>
Related Commands	VOLT:TRIG            VOLT

#### [SOURce:]VOLTage:SLEW:MODE

This command sets the voltage slew rate programming mode. Available modes are:

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

The voltage slew may be set between zero and the model's maximum voltage slew capability (determined by model) which can be queried with the VOLT:SLEW? MAX command.

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

#### [SOURce:]VOLTage:SLEW:TRIGgered

This command sets the voltage slew rate trigger level. When triggered by a step or pulse transient, the voltage slew rate will be set to the programmed level. Note that the new slew rate does not take effect until the unit is armed and a trigger event occurs. A parameter of MAXimum will set the slew to its maximum possible rate.

The voltage slew may be set between zero and the model's maximum voltage slew capability (determined by model) which can be queried with the VOLT:SLEW? MAX command.

Command Syntax	[SOURce:]VOLTage:SLEW:TRIGgered <NRf>
Parameters	0 to MAX
*RST Value	0
Examples	VOLT:SLEW:TRIG
Query Syntax	[SOURce:]VOLTage:SLEW:TRIG?
Returned Parameters	<NR2>
Related Commands	VOLT:SLEW:MODE    VOLT:SLEW

#### [SOURce:]VOLTage:TRIGgered[:AMPLitude]

This command sets the voltage trigger level. When triggered by a step or pulse transient, the voltage will be set to the programmed level. Note that the new voltage setting does not take effect until the unit is armed and a trigger event occurs. The maximum voltage that can be programmed is a function of the model and may be queried using the LIMIT subsystem. It is also further determined by the user defined VOLT:HIGH and VOLT:LOW settings.

The voltage slew may be set between zero and the model's maximum voltage slew capability (determined by model) which can be queried with the VOLT:SLEW? MAX command.

Command Syntax	[SOURce:]VOLTage:TRIGgered[:AMPLitude] <NRf>
Parameters	0 to MAX
*RST Value	0
Examples	VOLT:TRIG
Query Syntax	[SOURce:]VOLTage: TRIG?
Returned Parameters	<NR2>
Related Commands	VOLT:MODE    VOLT



## 4.8 Status Subsystem Commands

This subsystem programs the DC power supply's status registers. There are four groups of status registers;

- **Operation Status Group**
- **Questionable Status Group**
- **Questionable Instrument Isummary Group**
- **Standard Event Group**

The Standard Event group is programmed with Common commands. The Operation, Questionable, and Instrument ISummary status groups each consist of the following three registers:

Condition	Enable	Event
-----------	--------	-------

Refer to chapter 6 for more information about the status registers.

### Subsystem Syntax

STATus

:OPERation	
:CONDition	Returns the value of the condition register
:ENABle	Enables specific bits in the Event register
:EVENT	Returns the value of the event register
:QUESTionable	
:CONDition	Returns the value of the condition register
:ENABle	Enables specific bits in the Event register
:EVENT	Returns the value of the event register
:INSTRument	
:ISUMmary	
:CONDition	Returns the selected phase's condition register
:ENABle	Enable specific bits in selected phase's Event register values
:EVENT	Returns the selected phase's event register value

#### 4.8.1 Status - Operation Status Group

##### Subsystem Syntax

STATus

:OPERation

:CONDition

:ENABle

:EVENT

Returns the value of the condition register

Enables specific bits in the Event register

Returns the value of the event register

Bit Configuration of Status Operation Registers

Bit Position	15-5	4	3	1-2	0
Bit Name	not used	MEAS	TRANS	not used	CAL
Bit Weight		16	8		1

CAL                    Calibration is completed  
 TRANS                Transient is completed  
 MEAS                 Measurement is completed

##### STATus:OPERation:CONDition?

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

Query Syntax	STATus:OPERation:CONDition?
Parameters	None
Examples	STAT:OPER:COND?
Returned Parameters	<NR1>(Register value)

##### STATus:OPERation:ENABle

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

Command Syntax	STATus:OPERation:ENABle <NRf+>
Parameters	0 to 32727
*RST Value	0
Examples	STAT:OPER:ENAB 32 STAT:OPER:ENAB 1
Query Syntax	STATus:OPERation:ENABle?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:OPER:EVEN

**STATus:OPERation:EVENT?**

This query returns the value of the Operation Event register. The Event register is a read-only register which holds (latches) all events. Reading the Operation Event register clears it.

Query Syntax	STATus:OPERation:EVENT?
Parameters	None
Returned Parameters	<NR1>(Register Value)
Examples	STAT:OPER:EVEN?
Related Commands	*CLS

## 4.8.2 Status – Questionable Status Group

### Subsystem Syntax

**STATus**  
 :QUESTionable  
     :CONDition           Returns the value of the condition register  
     :ENABle <n>       Enables specific bits in the Event register  
     :EVENT               Returns the value of the event register

### Bit Configuration of Questionable Registers

Bit Position	15	14	13	12	11-10	9	8-5	4	3	2	1	0
Bit Name	LOV	LUV	CV	CC	n/a	RI	n/a	OT	n/a	OVP	CF	VF
Bit Weight	32768	16384	8192	4096		512		16	8	4	2	1

LOV           Line input over voltage protection has tripped  
 LUV           Line input under voltage protection has tripped  
 CV            Unit operating in Constant Voltage mode (status bit)  
 CC            Unit operating in Constant Current mode (status bit)  
 RI            remote inhibit is active  
 OT            over-temperature protection has tripped  
 OVP           Over voltage protection has triggered  
 CF            Over current protection has tripped  
 VF            Output under voltage (UV) or over voltage (OV) has occurred

### STATus:QUESTionable:CONDition?

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

Query Syntax	STATus:QUESTionable:CONDition?
Example	STAT:QUES:COND?
Returned Parameters	<NR1>(Register value)

### STATus:QUESTionable:ENABle

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

Command Syntax	STATus:QUESTionable:ENABle <NRf+>
Parameters	0 to 65535
*RST Value	0
Examples	STAT:QUES:ENAB 18
Query Syntax	STATus:QUESTionable:ENABle?
Returned Parameters	<NR1>(Register value)
Related Commands	STAT:QUES:EVENT?

**STATus:QUESTIONable:EVENT?**

This query returns the value of the Questionable Event register. The Event register is a read-only register which holds (latches) all events. Reading the Questionable Event register clears it.

Query Syntax	STATus:QUESTIONable:EVENT? QUESTIONable\:\EVENT?
Parameters	None
Returned Parameters	<NR1>(Register Value)
Examples	STAT:QUES:EVEN?
Related Commands	*CLS

### 4.8.3 Status - Instrument Isummary Group

#### Subsystem Syntax

STATus	
:INSTrument	
:ISUMmary	
:CONDition	Returns the selected phase's condition register
:ENABLE	Enable specific bits in selected phase's Event register values
:EVENT	Returns the selected phase's event register value

#### STATus:INSTrument:ISUMmary:CONDition?

This query returns the value of the Questionable Condition register. The Condition register is a read-only register which holds the real-time (unlatched) questionable status of the DC power supply.

Query Syntax	STATus:INSTrument:ISUMmary:CONDition?
Example	STAT: INST:ISUM:COND?
Returned Parameters	<NR1> (Register value)

#### STATus:INSTrument:ISUMmary:ENABLE

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

Command Syntax	STATus:INSTrument:ISUMmary:ENABLE <NRf+>
Parameters	0 to 32767
*RST Value	0
Examples	STAT:INST:ISUM:ENAB 18
Query Syntax	STATus:INSTrument:ISUMmary:ENABLE?
Returned Parameters	<NR1> (Register value)

#### STATus:INSTrument:ISUMmary:EVENT?

This query returns the value of the Questionable Event register. The Event register is a read-only register, which holds (latches) all events. Reading the Questionable Event register clears it.

Query Syntax	STATus:INSTrument:ISUMmary:EVENT? \:EVENT?
Parameters	None
Returned Parameters	<NR1>(Register Value)
Examples	STAT:INST:ISUM:EVENT?
Related Commands	*CLS STAT:INST:ISUM:NTR STAT:INST:ISUM:PTR

## 4.9 System Commands

The system commands control the system level functions of the Power supply.

### Subsystem Syntax

SYSTem:

COMMunicate:	Communication settings
GPIB:ADDRess	GPIB address, 0 to 31
SERial:	
BUAD	9600   19200   38400   57600   115200
PARity	NONE   EVEN   ODD
BITS	7   8
SBITS	1   2
CONFigure:	
[:BYTE]	System configuration query
ETIMe	Elapsed Time query
ERRor?	Returns the error number and error string
LOCal	Go to local mode
PON	Define the power on register number
REMote	Go to remote mode
RWLock	Local Lock out (front panel LOCAL button) (RS-232 only)
TEMPerature	Controller temperature query or error state.
VERSion	IEEE488.2 version

### SYSTem:COMMunicate:GPIB:ADDRess

This command will set the GPIB address. Address 0 is generally reserved for the GPIB controller.

Command Syntax	SYSTem:COMMunicate:GPIB:ADDRess <NRF>
Parameters	0 to 31 <listen address>
Example	SYST:COMM:GPIB:ADDR 2
Returned Parameters	<NR1>

### SYSTem:COMMunicate:SERial:BAUD

This command will set the SERial communication baud rate.

Command Syntax	SYSTem:COMMunicate:SERial:BAUD <NRF>
Parameters	9600   19200   38400   57600   115200
Example	SYST:COMM:SER:BAUD 38400
Returned Parameters	<NR1>

### SYSTem:COMMunicate:SERial:PARity

This command will set the SERial communication parity.

Command Syntax	SYSTem:COMMunicate:SERial:PARity <parity>
Parameters	NONE EVEN ODD
Example	SYST:COMM:SER:PAR NONE
Returned Parameters	<CRD>

### SYSTem:COMMunicate:SERial:BITS

This command will set the SERial communication number of data bits.

Command Syntax	SYSTem:COMMunicate:SERial:BITS <NRF>
Parameters	7 8
Example	SYST:COMM:SER:BITS 8
Returned Parameters	<NR1>

#### **SYSTem:COMMunicate:SERial:SBITs**

This command will set the SERIAL communication number of stop bits.

Command Syntax	SYSTem:COMMunicate:SERial:SBITs <NRF>
Parameters	1 2
Example	SYST:COMM:SER:SBITs 1
Returned Parameters	<NR1>

#### **SYSTem:CONFigure[:BYTE]**

This command will set the system configuration. This is a protected command and requires a password. The query format is available however. Refer to the \*OPT? Command for system options.

Command Syntax	SYSTem:CONFigure[:BYTE]
Parameters	SCPI, DO160,MIL704, FT
Example	SYST:CONF SCPI      SYST:CONF?
Returned Parameters	<CRD>
Related Commands	*OPT?

#### **SYSTem:ERRor?**

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

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

#### **SYSTem:ETIMe?**

This command will return the total number of accumulated hours, minutes and seconds.

Command Syntax	SYSTem:ETIMe?
Parameters	none
Example	SYST:ETIM?
Returned Parameters	<NR1>,<NR1>,<NR1>

#### **SYSTem:LOCal**

This command sets the interface in Local state, which enables the front panel controls. This command only applies to the RS232C interface. If IEEE 488 is used, the remote/local state is determined by the REN line on the IEEE 488 interface. In GPIB model, the SYST:LOC and SYST:REM commands will not be accepted and will generate an error message.



Command Syntax	SYSTem:LOCal
Parameters	None
Example	SYST:LOC
Related Commands	SYST:REM

### SYSTem:PON

This command is used to define the register value (0 though 7) the power supply will use to initialize its parameters at power up. If data in the selected register is not valid or the selected register is 8, the initialization will be with factory default values.

Command Syntax	SYSTem:PON <NRf>
Parameters	0 to 8
Examples	SYST:PON 1
Query Syntax	SYST:PON?
Returned Parameters	<NR1>
Related Commands	*RST *SAV

### SYSTem:REMOte

This command sets the interface in the Remote state, which disables all front panel controls. This command only applies to the RS232C interface. If IEEE 488 is used, the remote/local status is determined by the REN line on the IEEE 488 interface. In GPIB model, the SYST:LOC and SYST:REM commands will not be accepted and will generate an error message.

Command Syntax	SYSTem:REMOte
Parameters	None
Example	SYST:REM
Related Commands	SYST:LOC SYST:RWL

### SYSTem:TEMPerature

The query format of this command will return the internal ambient temperature of the DC power supply in degrees Celsius. The non-query mode can be used to disable Error 21. If Error 21 is disabled, the output will not trip when the maximum allowable ambient temperature is exceeded. This could result in intermittent controller errors however if the ambient temperature ratings are exceeded and is not recommended. Factory setting is fault 21 enabled. Once the state for Error 21 is set, it is retained at power up.

The trip point is set to 55° C as measured by the KDC controller. The operating range maximum is 40° C. Note that the non-query format is only supported on units with firmware revision 1.20 or higher.

Command Syntax	SYSTem:TEMPerature
Parameters	0   1
Example	SYST:TEMP?
Returned Parameters	<NR2>

### SYSTem:RWLock

This command can only be used with the RS-232 interface. It locks out the Local key on the front panel. The SYST:LOC command must be used to unlock the front panel. For GPIB, use the Local Lockout bus message.

Command Syntax	SYSTem:RWLock
Parameters	None
Example	SYST:RWL
Related Commands	SYST:LOC

**SYSTem:VERSion?**

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

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

## 4.10 Trigger Subsystem

This subsystem controls the triggering of the DC power supply. See chapter 0 under "Triggering Output Changes" for an explanation of the Trigger Subsystem. The INITiate commands control the initialization of the transient system. The trigger subsystem must first be enabled using the INITiate commands or no triggering action will occur. Refer to Figure 5-1 for a block diagram of the trigger system.

### Subsystem Syntax

TRIGger		
:ACQuire		
[:IMMediate]		Initiate data acquisition
:SOURce		Set the acquire trigger source
:STATe?		Return the trigger state (IDLE   WTRIG   BUSY)
[:TRANsient]		
[:IMMediate]		Initiate transient
:SOURce		Sets the trigger source (IMMediate   BUS   EXTernal)

### TRIGger:ACQuire[:IMMediate]

This command initiates a measurement acquisition if the measurement system is armed.

Command Syntax	TRIGger:ACQuire[:IMMediate] <CRD>
Parameters	none
*RST Value	NONE
Examples	TRIG:ACQ
Related Commands	ABOR TRIG:ACQ:SOUR INIT INIT:ACQ

### TRIGger:ACQuire:SOURce

This command selects the trigger source for a triggered measurement sequence as follows:

IMMediate	Acquire the measurements immediate. The FETCh commands return the requested calculation from the set of acquired data as soon as the measurement calculations are completed by the controller.
BUS	IEEE-488 device, *TRG, or <GET> (Group Execute Trigger)
TTLTrg	The signal driving the Trigger Out BNC
EXTernal	Triggering occurs when an external trigger input is received.

Command Syntax	TRIGger:ACQuire:SOURce<CRD>
Parameters	IMMediate   BUS   TTLTrig   EXTernal
*RST Value	IMM
Examples	TRIG:ACQ:SOUR BUS
Query Syntax	TRIGger:ACQuire:SOURce?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG:SOUR INIT INIT:CONT *TRG

### TRIG:STATe?

This is a query only command. The response will define the state of the transient trigger system as follows:

- IDLE      The trigger is in idle state.

- WTRIG The trigger is waiting for internal synchronization or external trigger
- BUSY The triggered transient in progress.

Query Syntax	TRIGger:STATe?
Returned Parameters	<CRD>
Related Commands	ABOR TRIG:SOUR TRIG:PHAS INIT INIT:CONT *TRG

#### **TRIGger[:TRANSient][:IMMediate]**

This command generates an immediate trigger to the transient system.

Command Syntax	TRIGger[:TRANSient][:IMMediate] TRIGger:TRANSient:IMMediate
Parameters	IMMediate
*RST Value	NONE
Examples	TRIG
Related Commands	ABOR INIT INIT:CONT

#### **TRIGger[:TRANSient]:SOURce**

This command will determine the source of the trigger event for the transient. The following sources are available:

- IMMediate Triggering occurs as soon as the INIT command is received.
- BUS Triggering occurs following the INIT command after receiving the \*TRG command or a Group Execute Trigger (GET) IEEE signal.
- EXTernal Triggering occurs when an external trigger input is received.

Command Syntax	TRIGger[:TRANSient]:SOURce
Parameters	IMMediate   BUS   EXTernal
*RST Value	IMMediate
Examples	TRIG:SOUR BUS
Related Commands	ABOR <group execute trigger>

## 4.11 Common Commands

Common commands begin with an \* and consist of three letters (command) or three letters and a ? (query). Common commands are defined by the IEEE 488.2 standard to perform some common interface functions. The power supply responds to the required common commands that control status reporting, synchronization, and internal operations. KDC Series units also respond to optional common commands that control stored operating parameters.

Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data or status specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately. The description for each common command or query specifies any status registers affected. Refer to Chapter 6 which explains how to read specific register bits and use the information that they return.

### Common Commands Syntax

*CLS	Clear status
*ESE <n>	Standard event status enable
*ESE?	Return standard event status enable
*ESR?	Return event status register
*IDN?	Return instrument identification
*OPT?	Return installed options
*RCL <n>	Recall instrument state
*RST	Reset
*SAV <n>	Save instrument state
*SRE <n>	Set service request enable register
*SRE?	Return service request enable register
*STB?	Return status byte
*TRG	Trigger
*TST	Self test
*WAI	Hold off bus until all device commands done

### 4.11.1 \*CLS

This command clears the following registers (see chapter 6 for descriptions of all status registers):

- Standard Event Status
- Status Byte
- Error Queue

Command Syntax	*CLS
Parameters	None

### 4.11.2 \*ESE

This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event register (see \*ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Enable Register are logically OR-ed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. See Section 16 for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

*Table 4-3: Bit Configuration of Standard Event Status Enable Register*

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

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

Command Syntax	*ESE <NRf>
Parameters	0 - 255
Power-On Value	128
Example	*ESE 129
Query Syntax	*ESE?
Returned Parameters	<NR1>(Register value)
Related Commands	*ESR? *STB?

### 4.11.3 \*ESR?

This query reads the Standard Event Status register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see \*ESE).

Query Syntax	*ESR?
Parameters	None
Returned Parameters	<NR1>(Register value)
Related Commands	*CLS *ESE *ESE?

If the \*ESR? returns the value of the status bits in the ESR register. Refer to Table 4-4 for the status bits and their meaning.

Table 4-4: Bit Configuration of Standard Event Status Register

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

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

### 4.11.4 \*IDN?

This query requests the power supply to identify itself. It returns the data in four fields separated by commas. Note that any installed options such as -IF are not reflected in the \*IDN? query response. The presence of these options that may affect the power supply's capabilities must be determined by using other queries.

Query Syntax	*IDN?										
Returned Parameters	<AARD>										
	<table border="0"> <tr> <td style="border-bottom: 1px solid black;">Field</td> <td style="border-bottom: 1px solid black;">Information</td> </tr> <tr> <td>AX</td> <td>Manufacturer</td> </tr> <tr> <td>xx-xx</td> <td>Model number, Volt - Current</td> </tr> <tr> <td>nnnnn</td> <td>Serial number or 0</td> </tr> <tr> <td>Rev. x.x</td> <td>Revision level of firmware</td> </tr> </table>	Field	Information	AX	Manufacturer	xx-xx	Model number, Volt - Current	nnnnn	Serial number or 0	Rev. x.x	Revision level of firmware
Field	Information										
AX	Manufacturer										
xx-xx	Model number, Volt - Current										
nnnnn	Serial number or 0										
Rev. x.x	Revision level of firmware										
Example	"AX,KDC80-63,12345,Rev 1.0"										

#### 4.11.5 \*OPT?

This query identifies any options that are installed. Options are identified by a string keyword or a number. A 0 indicates no options are installed. If an option is installed, its corresponding keyword will be returned. Refer also to the SYSTem:CONFigure[:BYTE] command.

Valid options are:

Parameter	Description
SCPI	SCPI command language
SYST	System master/auxiliary enabled
MIL704	Mil-Std 704 rev E test option
DO160	Reserved
FT	Fast Transient version. –FT option.

Query Syntax	*OPT?
Returned Parameters	<CRD>,[<CRD>]

#### 4.11.6 \*RCL

This command restores the power supply to a state that was previously stored in memory with a \*SAV command to the specified location. The KDC Series offers eight non-volatile memory locations for storing instrument setups.

Command Syntax	*RCL <NRf>
Parameters	0 through 7
Example	*RCL 3
Related Commands	*RST *SAV



**WARNING:** *Recalling a previously stored state may place hazardous voltages at the power supply output if the output state saved is ON.*



### 4.11.7 \*RST

This command resets the power supply to a setting defined by the values in the register defined by PON if valid or by the following factory-defined states:

Table 4-5: \*RST Default Parameter Values

Item	Value	Item	Value
VOLT	0.0 V	VOLT HIGH	Max. available
		VOLT LOW	0
CURR	Max. available	CURR HIGH	Max. available
		CURR LOW	0
OUTP	0 (OFF)		

Command Syntax	*RST
Parameters	None
Related Commands	*SAV

Table 4-6: Status Registers - Power on Conditions

Register	Status on Power On	Status after *RST
- ESE register	All 0's	All 0's
- ESR register	PON bit = 1, all others 0	PON = 0, all others unaffected
- STB register	All 0's	All 0's
- SRE register	All 0's	All 0's

### 4.11.8 \*SAV

This command stores the present state of the power supply to a specified location in memory. The KDC Series offers eight non-volatile memory locations for storing instrument setups.

Command Syntax	*SAV
Parameters	0 through 7
Related Commands	*RCL *RST

**4.11.9 \*SRE**

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (see \*STB for its bit configuration) are allowed to set the Master Status Summary (MSS). A 1 in any Service Request Enable Register bit position enables the corresponding Status Byte Register bit and all such enabled bits then are logically OR-ed to cause Bit 6 of the Status Byte Register to be set. See Section 6 for more details concerning this process.

When the controller conducts a STB? command, the MSS bit is not cleared. When \*SRE is cleared (by programming it with 0), the source cannot generate an MSS bit.

Command Syntax	*SRE <NRf>
Parameters	0 to 255
Default Value	128
Example	*SRE 255
Query Syntax	*SRE?
Returned Parameters	<NR1>(Register binary value)
Related Commands	*ESE *ESR

**4.11.10 \*STB?**

This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. Reading the Status Byte register does not clear it. The input summary bits are cleared when the appropriate event registers are read (see Section 6 for more information).

*Table 4-7: Bit Configuration of Status Byte Register*

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

OPER	operation status summary	MSS	master status summary
ESB	event status byte summary		
QUES	questionable status summary	MAV	message available

Query Syntax	*STB?
Returned Parameters	<NR1> (Register binary value)

**4.11.11 \*TRG**

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

Command Syntax	*TRG
Parameters	None
Related Commands	ABOR INIT

---

#### 4.11.12 \*TST

---

This command executes an internal selftest.

Command Syntax	*TST
Parameters	None
Related Commands	ABOR

---

#### 4.11.13 \*WAI

---

This command instructs the power supply not to process any further commands until all pending measurement and calibration operations are completed. Pending operations are complete when all commands sent before \*WAI have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect output voltage or state and trigger actions are overlapped with subsequent commands sent to the power supply. The \*WAI command prevents subsequent commands from being executed before any overlapped commands have been completed.

The \*WAI command may be used when performing measurement queries (MEAS or FETCh) to force the power supply to respond to the query before processing any subsequent command.

\*WAI can be aborted by sending any other command after the \*WAI command.

Command Syntax	*WAI
Parameters	None
Related Commands	*OPC

---

## 5 Programming Examples

---

### 5.1 Introduction

---

This chapter contains examples on how to program the Power supply. Simple examples show you how to program:

- Output functions such as voltage and current
- Measurement functions

More complex examples are used to illustrate programming of:

- Output transients
- Triggered measurements

The examples in this chapter use generic SCPI commands. See Section 3 for information about encoding the commands as language strings. Where appropriate, optional commands are shown for clarity in the examples.

### 5.2 Programming the Output

---

#### 5.2.1 Power-on Initialization

---

When the Power supply is first turned on, it comes up with the output state defined by the PON register number. If the register number or the register has no valid data, the Power supply initializes to the following state.

```
VOLT 0.0  
OUTP 0
```

The following commands are given implicitly at power-on:

```
*RST  
*CLS  
*SRE 128  
*ESE 0
```

\*RST is a convenient way to program all parameters to a known state. Refer to Table 4-5 under the \*RST command in paragraph 4.11.7 to see how each programmable parameter is set by \*RST.

#### 5.2.2 Enabling the Output

---

To enable the output, use the command:

```
OUTP 1
```

##### **Voltage**

The output voltage is controlled with the VOLTage command. For example, to set the output voltage to 125 volts, use:

```
VOLTage 125
```

The maximum voltage that the Power supply can output is limited by model configuration and the user set voltage limits. These values are defined by the LIMIT subsystem.

### Current Limit

To set the current limit use:

```
CURRent <n>
```

where <n> is the current limit in amperes.

If the load attempts to draw more current than the programmed limit, the power supply switches to a constant current mode and will reduce the output voltage as required if in constant current mode. In constant voltage mode, the output trips off if the current limit is exceeded. The mode is controlled by the SOURce:MODE command.

The maximum current that the Power supply can output is limited by model configuration and is defined by the LIMIT subsystem.

## 5.2.3 Voltage and Current Slew Rates

### Voltage Slew

The AC source has the ability to control the slew rate of amplitude and current changes. This can be used to generate ramps or to protect sensitive loads. To set the voltage slew rate to 20 volts per second, use:

```
VOLTage:SLEW 20
```

At \*RST the slew rate is set to Maximum, which means that voltage changes occur at the fastest possible slew rate. The slew rate applies to programmed changes in output voltage while the unit is operating in fixed mode. Voltage changes made by the step, pulse, and list transients are controlled by the same rules that apply to all other functions that are subject to transient control. See “Programming Output Transients”, section 5.5.

### Current Slew

The DC supply also has the ability to control the slew rate of current changes. To set the current slew rate to 2.5 A per second, use:

```
CURRent:SLEW 2.5
```

At \*RST the slew rate is set to MAXimum, which means that programmed current changes occur instantaneously. The current slew rate applies to programmed changes in current while the unit is operating in fixed mode. Current changes made by the step, pulse, and list transients are controlled by the same rules that apply to all other functions that are subject to transient control. See “Programming Output Transients”, section 5.5.

## 5.3 Basic Measurements

The KDC supply has the capability to return current, peak current, power and voltage measurements.

### Voltage Measurement

To measure the voltage, use:

```
OUTP 1                               /* Close output if it was not closed already.
MEAS:VOLT?
```

### Current Measurements

To measure the average DC current, use:

```
MEAS:CURR?
```

To measure the peak current, use:

```

MEAS:CURRE:AMPL:RES /* resets track and hold for peak curr. meas.
/* It may be need to turn the output on or program
/* as specific voltage at this time before taking a
/* peak current reading.
MEAS:CURRE:AMPL:MAX? /* retrieves peak current reading.

```

### Power Measurements

To measure the power, use:

```
MEAS:POW?
```

## 5.4 Setting the Power on Initialization Values

The power supply is shipped with default factory settings when the unit is powered up. The factory settings are:

Parameter	Factory default setting
Voltage	0.0 Volt
Current limit	Maximum available current.
Display mode	Current measurement
Output state	OFF
Local / Remote State	Local. Front panel unlocked.

*Table 5-1: Factory Default Power on Settings*

It is possible to change the power on initialization values in one of two ways:

1. Using the RS232 or optional IEEE-488 interface.
2. Using the front panel.

To change the power on initialization values over the bus, proceed as follows:

1. Set the unit up in the desired way using the relevant bus commands.
2. Save the new setup you want to make the power-on setting to one of the 8 setup registers using the \*SAV <n> command. (n = 0 through 7)
3. Assign the register as the Power-on register using the SYST:PON <n> command where <n> is register used in step 2.

---

## 5.5 Programming Output Transients

---

Output transients are used to:

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

The following functions are subject to transient control:

- Output voltage
- Current limit
- Voltage slew rate
- Current slew rate

The following transient modes can be generated:

Step	generates a single triggered output change.
Pulse	generates an output change which returns to its original state after some time period.
List	generates a sequence of output changes, each with an associated dwell time or paced by triggers.
Fixed	turns off the transient functions, which means that only the IMMEDIATE values are used as the data source for a particular function.

After a \*RST or Device Clear command, all functions are set to FIXEd, which turns off the transient functions.

### 5.5.1 Transient System Model

When a trigger is received in step or pulse modes, the triggered functions are set from their IMMEDIATE to their TRIGGERED value. In Step mode, the triggered value becomes the immediate value. In Pulse mode, the functions return to their immediate value during the low portion of the pulse.

If there are no further pulses, the immediate value remains in effect. In List mode, the functions remain at the last list value at the completion of the list.

You can not mix STEP, PULSE, and LIST modes among functions.

### 5.5.2 Step and Pulse Transients

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

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

**Step 2:** Set the triggered level of the function that will generate the transient. For example, if the previously programmed voltage function is going to step the output voltage amplitude to 135 volts upon receipt of a trigger, use:

```
VOLTage:TRIGger 135
```

**Step 3:** Select the trigger source that will generate the trigger. For example, to select the Immediate use:

```
TRIGger:SOURce IMM
```

Trigger sources are discussed in detail under “Triggering Output Changes”

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

PULSE:COUNT 1	specifies 1 output pulse
PULSE:PERIOD 1	specifies a pulse period of 1 second
PULSE:DCYCLE 50	specifies a duty cycle of 50%
PULSE:WIDTH .5	specifies a pulse width of .5 seconds (not necessary in this case since a duty cycle has already been specified)

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

```
INITiate
```

#### Example

The following example programs a voltage dropout for 2 seconds. The dropout is triggered by IEEE-488 bus trigger.

*RST	Begin at power-on state
VOLT 120	Set initial output voltage (immediate-level)
OUTP ON	Enable the output
VOLT:MODE PULS	Enable output to generate pulses when triggered
VOLT:TRIG 0	Set the voltage dropout (triggered level)
PULS:WIDT 2	Set pulse width for 2 seconds



PULS:PER 4	Set pulse period for 4 seconds
TRIG:SOUR BUS	Respond to IEEE-488 bus triggers
INIT	Set to Wait-for-trigger state
<device trigger>	Send the IEEE-488 bus trigger

### 5.5.3 List Transients

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

You can program up to 99 settings (or points) in the list, the time interval (dwell) that each setting is maintained, the number of times each data point is repeated, the number of times that the list will be executed, and how the settings change in response to triggers.

All list point data can be stored in nonvolatile memory using one of the available setup register. This means that the programmed data for any list function will be retained when it is saved in the register when the DC supply is turned off.

---

**Note:** *If the transient list is programmed over the bus, the transient must be executed at least once or it will not be saved to the setup register when the \*SAV command is sent. Unexecuted transient lists have not been compiled yet by the KDC power supply controller and will not be saved as part of a front panel setup. This is not the case when the transient list is entered from the front panel. Front panel entered transient lists will be saved even if they have not been executed yet.*

---

Lists are paced by a separate list of dwell times, which define the duration of each output setting. Therefore, each of the up to 99 list points has an associated dwell time, which specifies the time (in seconds) that the output remain at that setting before moving on to the next setting.

The following procedure shows how to generate a simple list of voltage and current changes.

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

```
VOLTage:MODE LIST
CURRent:MODE LIST
```

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

```
LIST:VOLTage 24, 20, 16
```

You can specify lists for more than one function. For example, to synchronize the previous voltage list with another list that varies the maximum current allowed at each voltage, the lists may include the following values:

```
LIST:VOLTage 24, 20, 16
LIST:Current 100, 120, 160
```

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

```
LIST:VOLTage 24, 20, 16;CURR 160
```

is the same as:

```
LIST:VOLTage 24, 20, 16
```

```
LIST:CURREnt 160, 160, 160
```

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

```
LIST:DWELL 10, 20, 10
```

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

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

```
LIST:COUNT 10
```

At \*RST, the count is set to 1.

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

```
LIST:STEP AUTO
```

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

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

```
LIST:STEP ONCE
```

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

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

## 5.6 Triggering Output Changes

The following transient trigger sources can be selected:

- IMMEDIATE Generates a trigger when the trigger system is initiated.
- BUS Selects IEEE-488 bus triggers.
- EXTERNAL External trigger input

### 5.6.1 Trigger System Model

Figure 5-1 is a model of the trigger system. The rectangular boxes represent states. The arrows show the transitions between states. These are labeled with the input or event that causes the transition to occur.

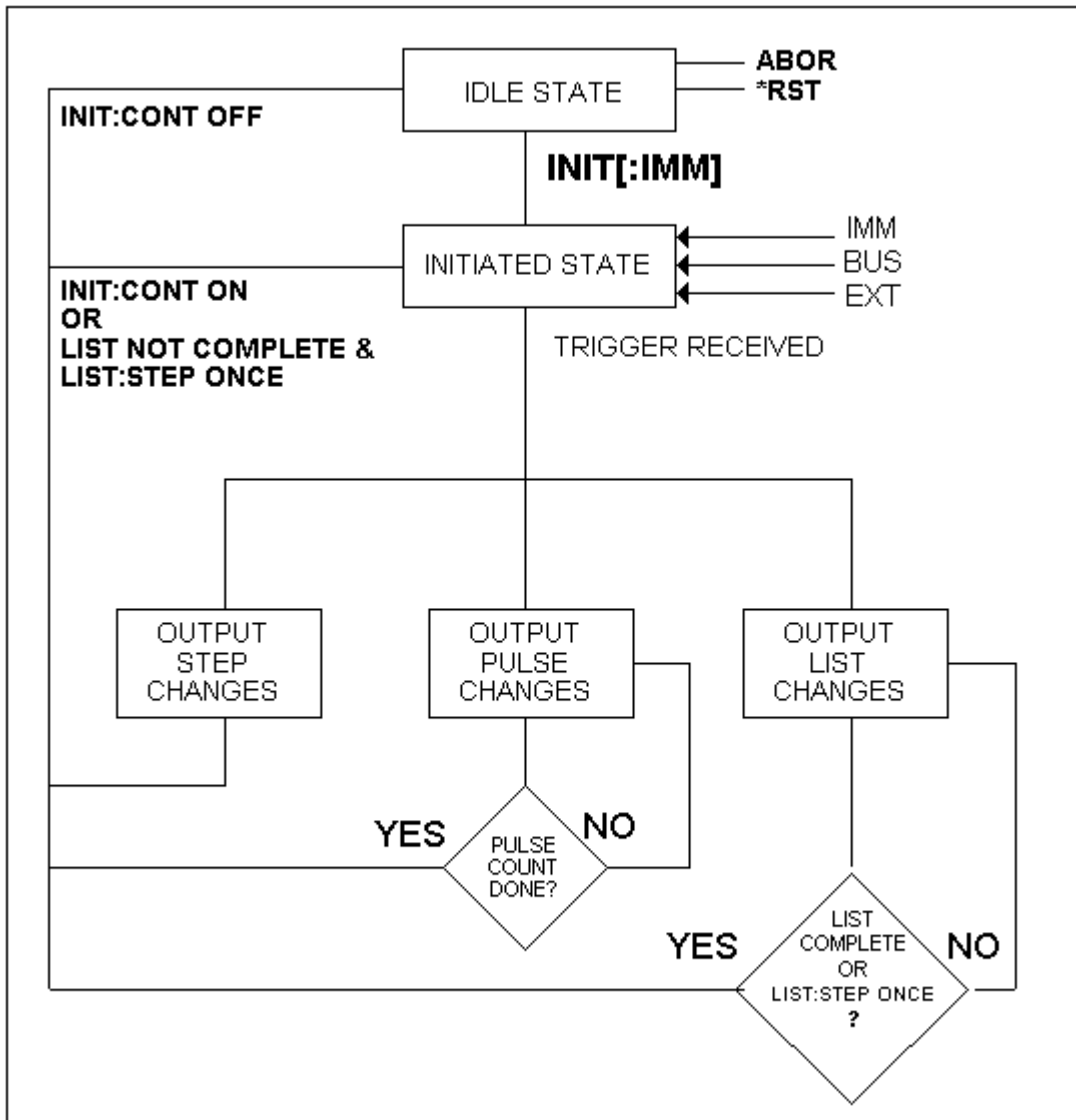


Figure 5-1 : Transient Trigger System Model

---

## 5.6.2 Initiating the Output Trigger System

---

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

```
ABORT
*RST
```

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

```
INITiate:IMMEDIATE
```

After a trigger is received and the action completes, the trigger system will return to the Idle state. Thus it will be necessary to initiate the system each time a triggered action is desired. To keep a trigger system initiated for multiple actions without having to send an initiate command for each trigger, use:

```
INITiate:CONTINUOUS ON
```

### Selecting the Trigger Source

The trigger system is waiting for a trigger signal in the Initiated state. Before generating a trigger, a trigger source must be select.

To select IEEE-488 bus triggers (group execute trigger, device trigger, or \*TRG command), use:

```
TRIGger:SOURce BUS
```

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

```
TRIGger:SOURce IMM
```

The immediate source can be combined with INITiate:CONTINUOUS ON to generate repetitive output transients. A transition from the Initiated state to the Delay state is made when the trigger signal is received.

---

## 5.6.3 Generating Triggers

---

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

### Single Triggers

By sending one of the following over the IEEE-488:

- INIT
- \*TRG
- a group execute trigger

### Continuous Triggers

By sending the following commands over the IEEE-488:

- TRIGger: SOURce IMMEDIATE
- INITiate:CONTINUOUS ON

When the trigger system enters the Output Change state upon receipt of a trigger, the triggered functions are set to their programmed trigger levels. When the triggered actions are completed, the trigger system returns to the Idle state.

### Specifying a Dwell Time for Each List Point

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

```
LIST:DWELL <n> ,<n>
```

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

At the end of the dwell interval, the transition to the next state depends on whether or not the list has completed its sequencing and the state of the LIST:STEP command (see Figure 5-1).

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

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

```
LIST:STEP ONCE    programs the trigger system to return to the Initiated state to wait for the
                  next trigger.
LIST:STEP AUTO    programs the trigger system to immediately execute the next list point.
```

## 5.7 Acquiring Measurement Data

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

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

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

The query response for measurements is not immediate. The source will accept commands from the interface while the measurement is in progress. To prevent the source from accepting additional commands during measurement the \*WAI must be used with the measurement query command.

```
MEAS:VOLT?;*WAI
```

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

---

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

---

### 5.7.1 Voltage, Current and Power Measurements

---

To the rms voltage or current, use:

```
MEASure:VOLTage?or
```

```
MEASure:CURREnt?
```

To measure the maximum current amplitude, use:

```
MEASure:CURREnt:AMPLitude:MAXimum?
```

To reset the peak measurement , use

```
MEAS:CURR:AMPL:RESet
```

To measure the power, use:

```
MEASure:POWer?
```

### 5.7.2 Returning Voltage and Current Data From the Data Buffer

---

The MEASure and FETCh queries can also return all 4096 data values of the instantaneous voltage and current buffers. These are:

```
MEASure:ARRay:CURREnt[:DC]?;*WAI
```

```
MEASure:ARRay:VOLTage[:DC]?;*WAI
```

### 5.7.3 Triggering Measurements

---

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

The following measurement trigger sources can be selected:

IMMediate	Aquire the measurements immediate
BUS	IEEE-488 device, *TRG, or <GET> (Group Execute Trigger)
TTLTrg	The signal driving the Trigger Out

### 5.7.4 Measurement Trigger System Model

---

Figure 5-2 is a model of the measurement trigger system. The rectangular boxes represent states. The arrows show the transitions between states. These are labeled with the input or event that causes the transition to occur.



Figure 5-2: Measurement Acquisition Trigger Model

### 5.7.5 Initiating the Measurement Trigger System

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

```
ABORt    *RST    *RCL
```

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

```
INITiate:IMMediate:ACQuire
```

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

---

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

---

### 5.7.6 Selecting the Measurement Trigger Source

The trigger system is waiting for a trigger signal in the Initiated state. Before you generate a trigger, you must select a trigger source. To select the external trigger to start the measurement:

```
TRIGger:ACQuire:SOURce EXT
```

To select IEEE-488 bus triggers (group execute trigger, device trigger, or \*TRG command), use:

```
TRIGger:ACQuire:SOURce BUS
```

To select the signal driving the Trigger Out from the transient, use:

```
TRIGger:ACQuire:SOURce TTLTrg
```

---

### 5.7.7 Generating Measurement Triggers

---

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

- By sending one of the following over the IEEE-488:
  - TRIGger:ACQuire:IMMediate
  - \*TRG
  - a group execute trigger (GET)
- By generating an output transient that causes the Trig Out to output a pulse.

---

## 5.8 Controlling the Instantaneous Voltage and Current Data Buffers

---

Measurements taken by the DC power supply are based on a digital representation of the voltage and current output. Both parameters are digitized with 18 bits of amplitude resolution and 20.833  $\mu$ sec of time resolution. (48 Ks/sec real-time sampling rate). The available memory depth to hold this information is 4K (4096 samples) for each parameter. This section covers the SCPI commands that can be used to control the position of the trigger point in the 4K deep data buffer.

---

### 5.8.1 Query the Voltage and Current Sampling Rate

---

The output voltage and current sampling rate is 48 Ks/sec (period = 20.833  $\mu$ sec). This means that it takes about 85.33 milliseconds to fill up 4096 data points in the voltage and current data buffers with the information required to make a measurement calculation. You can query this data sampling rate with:

```
SENSe:SWEep:TINTerval?
```

The sample interval can be increased (sample rate decreased) up to 10 times using the SENS:SWE:TINT <n> command where <n> is a multiple of 20.833.

---

### 5.8.2 Pre-event and Post-event Triggering [ix series only]

---

The range for this offset is -85 msec to 1000 msec at the maximum sampling rate. As shown in the following figure, when the offset is negative, the values at the beginning of the data record represent samples taken prior to the trigger. When the value is 0, all of the values are taken after the trigger. Values greater than zero can be used to program a delay time from the receipt of the trigger until the data points that are entered into the buffer are valid.

(Delay time = Offset  $\times$  Sample period)



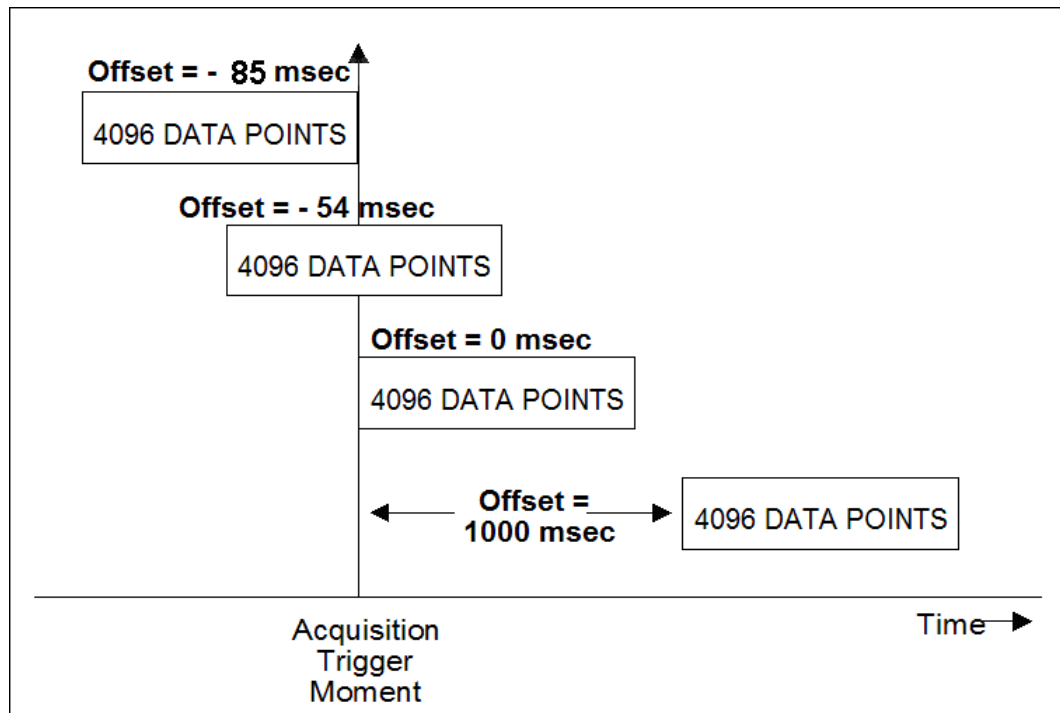


Figure 5-3: Pre-event and Post-event Triggering

## 5.9 Trigger System Summary

Figure shows a functional block diagram of the AC/DC source's trigger system. If both trigger systems are initiated at the same time, the Acquisition trigger system will respond to the first trigger generated while the transient trigger system will respond to the second trigger. Thus, the acquisition has the highest priority.

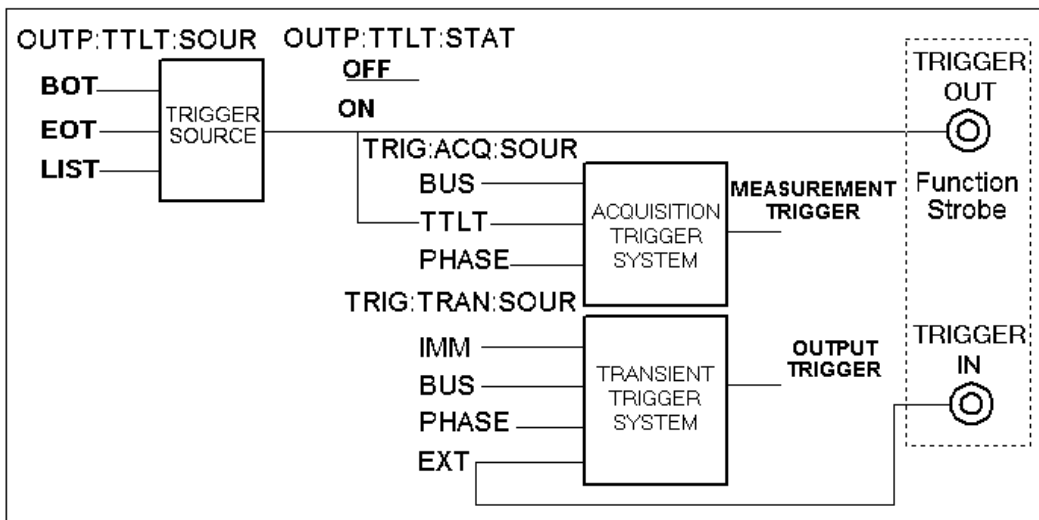


Figure 5-4: Trigger system block diagram

## 6 Status Registers

---

You can use status register programming to determine the operating condition of the power supply at any time. For example, you may program the Power supply to generate an MSS bit when an event such as a current limit occurs. When the MSS bit is set, your program can then act on the event in the appropriate fashion.

Figure 6-1 shows the status register structure of the power supply. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard IEEE-488 functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.

### 6.1 Power-On Conditions

---

Refer to the \*RST command description in paragraph 4.11.7 for the power-on conditions of the status registers.

### 6.2 Standard Event Status Group

---

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

Command	Action
*ESE	programs specific bits in the Standard Event Enable register.
*ESR?	reads and clears the Standard Event register.

The PON bit in the Standard Event register is set whenever the Power supply is turned on.

### 6.3 Status Byte Register

---

This register summarizes the information from all other status groups as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The bit configuration is shown in Table 4-7.

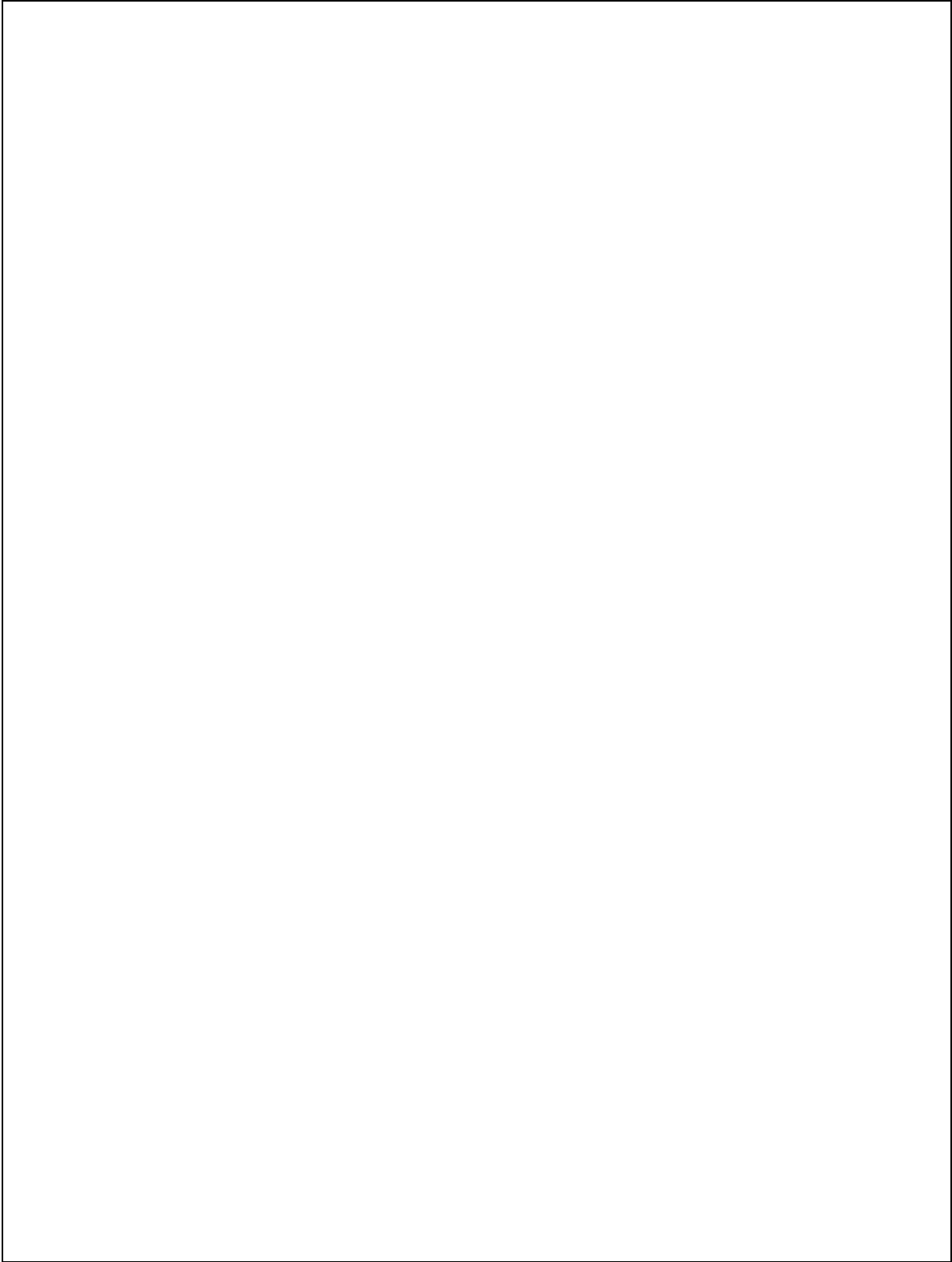
Command	Action
*STB?	reads the data in the register but does not clear it (returns MSS in bit 6)

The MSS Bit

This is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the Power supply has one or more reasons for needing 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 MAV bit and Output Queue

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



*Figure 6-1: Power Supply Status System Model*

---

## 6.4 Examples

---

The following section contains examples of commonly used operations involving the status registers.

You can determine the reason for an MSS bit set by the following actions:

**Step 1:** Determine which summary bits are active. Use:

\*STB?

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

ESR?

---

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

---

**Step 3:** Remove the specific condition that caused the event. If this is not possible, the event may be disabled by programming the corresponding bit of the status group Enable.



---

## 7 Option Commands

---

### 7.1 Introduction

---

The KDC Series offers a number of options that are implemented in the power supply controller. If one or more of these options are installed, they may be used from both the front panel and the interface. For front panel operation of any of these firmware options, consult the User Manual supplied with the unit. For use in an automated test system, this chapter contains the command sub system syntax for the available test options.

Specifically, the following firmware options are available at the time of this manual's printing:

- MIL-STD 704 rev D and E, Option -704

The DC power supply will report all installed options, including any firmware options, when queried using the `SYST:CONF:BYTE?` or `*OPT?` command. Additional firmware options may be added over time. Consult the factory for additional information on firmware options not listed here but present on your unit.

## 7.2 Mil-Std 704 Option -704

MIL704 tests can be executed remotely from the GPIB or RS232 using the following command tree. See the KDC Series User Manual (P/N 5005-960) for more details about each 704 test step.

### 7.2.1 DC System

```

:MIL704
  :VERSion D|E
  :SSTate
    :VOLT
      [:LEVel]
      :RIPPlE
  :TRANsient
    :VOLTage
      [:LOW]
      :HIGH
  :ABNormal
    :VOLTage
      [:UNDEr]
      :OVER
  :EMERgency
    :VOLTage

```

Note that none of these commands accept parameters nor have a query format except for the MIL704:VERSion command, which accepts a letter D or E as a parameter. Its query format response will return the letter D or E based on the revision selected.

#### **:MIL704:VERSion**

This command sets the Mil Std 704 standard revision to be used. Available versions are D and E. Older and or newer versions may be added through firmware updates. Note that this is the only MIL704 command that has a parameter and supports the query format.

Command Syntax	:MIL704:VERSion
Parameters	D   E
*RST Value	E
Examples	:MIL704:VERS D
Query Syntax	:MIL704:VERSion?
Returned Parameters	<CRD>
Related Commands	none

#### **:MIL704:SSTate:VOLTage[:LEVel]**

This command initiates the Steady state voltage level test step.

Command Syntax	:MIL704:SSTate:VOLTage[:LEVel]
Parameters	none
Examples	:MIL704:SST:VOLT
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:SSTate:VOLTage:RIPPlE

**:MIL704:SSTate:VOLTage:RIPPLE**

This command initiates the Steady state voltage ripple test step.

Command Syntax	:MIL704:SSTate:VOLTage:RIPPLE
Parameters	none
Examples	:MIL704:SST:RIPP
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:SSTate:VOLTage

**:MIL704:TRANSient:VOLTage[:LOW]**

This command initiates the Transient low voltage level test step.

Command Syntax	:MIL704:TRANSient:VOLTage[:LOW]
Parameters	none
Examples	:MIL704:TRAN:VOLT
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:TRANSient:VOLTage:HIGH

**:MIL704:TRANSient:VOLTage:HIGH**

This command initiates the Transient high voltage level test step.

Command Syntax	:MIL704:TRANSient:VOLTage:HIGH
Parameters	none
Examples	:MIL704:TRAN:VOLT:HIGH
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:TRANSient:VOLTage:LOW

**:MIL704:ABNormal:VOLTage[:UNDer]**

This command initiates the Abnormal under-voltage level test step.

Command Syntax	:MIL704:ABNormal:VOLTage[:UNDer]]
Parameters	none
Examples	:MIL704:ABN:VOLT
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:ABNormal:VOLTage:OVER

**:MIL704:ABNormal:VOLTage:OVER**

This command initiates the Abnormal over-voltage level test step.

Command Syntax	:MIL704:ABNormal:VOLTage:OVER
Parameters	none
Examples	:MIL704:ABN:VOLT:OVER
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	:MIL704:ABNormal:VOLTage:UNDer



**:MIL704:EMERgency:VOLTage**

This command initiates the Emergency voltage level test step.

Command Syntax	:MIL704:EMERgency:VOLTage
Parameters	none
Examples	:MIL704:EMER:VOLT
Query Syntax	n/a
Returned Parameters	n/a
Related Commands	none

**7.2.2 Programming Considerations**

Some of the tests take a long time to complete. The \*OPC command may be used to determine the completion of the command.

Example:

```
MIL704:SST:VOLT;*OPC 1
```

The query \*OPC? Will return a 0 while the test is in progress and a 1 when the test is completed.

## Appendix A: SCPI Command tree

Command Syntax Tree.

ABORt	Resets the trigger system to the Idle state
CALibrate	
:MEASure	
:CURRent	
[:AMBient]	Ambient temperature calibrations
[:FSCale]	Calibrate full-scale DC current measurements
:ZERO	Calibrate DC current measurements offset
:TEMPerature	Elevated temperature calibrations
[:FSCale]	Calibrate full-scale DC current measurements at higher temperature
:VOLTage	
[:AMBient]	Ambient temperature calibrations
[:FSCale]	Calibrate full-scale DC voltage measurements
:ZERO	Calibrate DC voltage measurements offset
:TEMPerature	Elevated temperature calibrations
[:FSCale]	Calibrate full-scale DC voltage measurements
[:SOURce]	
:CURRent	
:EXTernal	RPI input calibration
:5V	5V input range
:10V	10V input range
:RESistance	Resistive input range
:INTernal	Internal
:OFFSet	offset calibration
:RANGe	full scale calibration
:VOLTage	
:EXTernal	RPI input calibration
:5V	5V input range
:10V	10V input range
:RESistance	Resistive input range
:INTernal	Internal
:OFFSet	offset calibration
:RANGe	full scale calibration
:PASSword	
:SAVE	
INITiate	
[:IMMEDIATE]	Initiates the system for one trigger
[:TRANSient]	
:ACQuire	
:CONTinuous	sets if the initiate continuous to (ON OFF)
MEASure/FETch	
:ARRay	
:CURRent[:DC]?	Returns the DC current data points.
:MODE	BINary or ASCii data format
:POWer[:DC]?	Returns the DC power measurement data points.
:VOLTage[:DC]?	Returns the DC voltage measurement data points.
[:SCALar]	
:CURRent	
[:DC]?	Returns the average DC value of current
:AMPLitude	
:MAX?	Returns peak current measurement reading.
:RESet	Resets the peak current measurements.

	:POWer[:DC]?	Returns the DC power measurement reading.
	:VOLTage[:DC]?	Returns the DC voltage measurement reading.
MIL704		Option -704 Commands.
	:VERSion DJE	Version selection
	:SState	
	:VOLT	
	[:LEVel]	Steady State voltage test
	:RIPple	Steady state ripple test
	:TRANSient	
	:VOLTage	
	[:LOW]	Transient voltage low test
	:HIGH	Transient voltage high test
	:ABNormal	
	:VOLTage	
	[:UNDer]	Abnormal under voltage test
	:OVER	Abnormal over voltage test
	:EMERgency	
	:VOLTage	Emergency voltage test
OUTPut		
	[:STATe]	Enable/disable output voltage, current, power, etc.
	:TTLTrg	
	[:STATe]	Enable/disable trigger out drive
	:SOURce	Selects a TTLTrg source (BOT EOT LIST)
	:IMPedance	
	[:STATe]	Enable/disable output impedance program
	:REAL	Sets resistive part of output impedance
SENSE		
	:SWEep	
	:OFFSet	Define trigger points relative to start of the record
	:TINTerval?	Queries the digitizer sample interval
	[:SOURce]	
	:ALC	Enables or disables ALC mode on –FT models.
	:CURRent	
	[:LEVel]	
	[:IMMediate]	
	[:AMPLitude]	
	[:DC]	Set the current value in amps. 0 to range
	:HIGH	Set the maximum current allowed. 0 to range
	:LOW	Set the minimum current allowed. 0 to range
	:MODE	FIXed   STEP   PULSe   LIST
	:PROTection	
	:DELay	0.1 to 5 sec
	:STATE	n/a
	:REFerence:	INTernal   EXT5V   EXT10V   EXTRES
	:SLEW	
	[:IMMediate]	Current slew rate in A/sec
	:MODE	FIXed   STEP   PULSe   LIST
	:TRIGger	0 to range
	:TRIGgered	
	[:AMPLitude]	0 to range
:LIMit		
	:CURRent?	Current range limit: Model specific
	:VOLTage?	Voltage range limit. Model specific
:LIST		
	:COUNt	Number of times to repeat transient list
	:CURRent	

[:LEVEL]	
[:AMPLitude]	
:POINTs	
:SLEW	
[:RATE]	0.0001 to 1e6 / sec
:POINTs	
:DWELl	
[:TIME]	Dwell time in seconds
:POINTs	Dwell data point value
:REPeat	
[:TIME]	Number of time to repeat list data point
:POINTs	Repeat data point value
:STEP	ONCE   AUTO
:TTLTrig	
[:SWITCh]	0.0001 to 1e6 / sec
:POINTs	
:VOLTage	
[:LEVEL]	
[:AMPLitude]	
:POINTs	
:SLEW	
[:RATE]	0.0001 to 1e6 / sec
:POINTs	
:MODE	CVOLTage   CCURrent   CPOWER
:POWER	
:PULSE	
:COUNT	1 to 2e8
:DCYClE	0 to 100% pulse width must be >= 0.001
:HOLD	WIDTh   DCYClE
:PERiod	0.002 to 90,000
:WIDth	0.001 to < 90,000
:VOLTage	
[:LEVel]	
[:IMMediate]	
[:AMPLitude]	
[:DC]	Set the voltage value in volts. 0 to range
:HIGH	Set the maximum voltage allowed. 0 to range
:LOW	Set the minimum voltage allowed. 0 to range
:MODE	FIXed   STEP   PULSe   LIST
:PROTection	
[:LEVel]	0 to 110%
:DELay	0.1 to 5 sec
:MODE	RANGe   SVOLTage   ABSolute
:REFerence:	INTernal   EXT5V   EXT10V   EXTRES
:SLEW	
[:IMMediate]	Voltage slew rate in V/sec
:MODE	FIXed   STEP   PULSe   LIST
:TRIGger	0 to range
:TRIGgered	
[:AMPLitude]	0 to range
STATUs	
:OPERation	
:CONDition	Returns the value of the condition register
:ENABle	Enables specific bits in the Event register
:EVENt	Returns the value of the event register
:QUEStionable	

:CONDition	Returns the value of the condition register
:ENABle	Enables specific bits in the Event register
:EVENT	Returns the value of the event register
:INSTrument	
:ISUMmary	
:CONDition	Returns the selected phase's condition register
:ENABle	Enable specific bits in selected phase's Event register values
:EVENT	Returns the selected phase's event register value
SYSTem:	
COMMunicate:	Communication settings
GPIB:ADDRess	GPIB address, 1 to 31
SERial:	
BUAD	9600   19200   38400   57600   115200
PARity	NONE   EVEN   ODD
BITS	7   8
SBITS	1   2
CONFigure:	
[:BYTE]	System configuration query
ERRor?	Returns the error number and error string
ETIMe?	Elapsed Time query
LOCal	Go to local mode
PON	Define the power on register number
REMote	Go to remote mode
RWLock	Local Lockout
TEMPerature	Controller temperature query or error state.
VERSion	IEEE488.2 version
TRIGger	
:ACQuire	
[:IMMEDIATE]	Initiate data acquisition
:SOURce	Set the acquire trigger source
:STATe?	Return the trigger state (IDLE   WTRIG   BUSY)
[:TRANsient]	
[:IMMEDIATE]	Initiate transient
:SOURce	Sets the trigger source (IMMEDIATE   BUS   EXTernal)

---

## Appendix B: SCPI Conformance Information

### SCPI Version

The KDC Series DC power supplies conform to SCPI version 1990.0.

## Appendix C: Error Messages

Any errors that occur during operation from either the front panel or the remote control interface will result in error messages. Error messages are displayed on the LCD display. They are also stored in the error message queue from which they can be queried using the SYST:ERR? Query. The error queue has a finite depth. If more error messages are generated than can be held in the queue, a queue overflow message will be put in the last queue location. To empty the queue, use the error query until the No Error result is received.

Errors appearing on the LCD will generally remain visible until the user moves to another screen. If multiple error messages are generated in succession, only the last message will be visible as there is only space for one error message on the LCD display.

The same area of the display is also used to display status messages. While error messages always have a negative error number, status messages have a positive number.

The table below displays a list of possible error and status messages along with their possible cause and remedy.

Number	Message String	Cause	Remedy
0	"No error"	No errors in queue	
-100	"Command error"	Unable to complete requested operation	Unit may be in a mode inconsistent with request.
-102	"Syntax error"	Command syntax incorrect.	Misspelled or unsupported command
-103	"Invalid separator"	SCPI separator not recognized	See SCPI section of programming manual.
-104	"Data type error"	Data type invaled.	Check command for supported data types
-108	"Parameter not allowed"	One or more additional parameters were received.	Check programming manual for correct number of parameters
-109	"Missing parameter"	Too few parameters received for requested operation	Check programming manual for correct number of parameters
-110	"Command header error"	Command header incorrect	Check syntax of command.
-111	"header separator error"	Invalid command separator used.	Use semi-colon to separate command headers
-112	"Program mnemonic too long"	Syntax error	Check programming manual for correct command syntax
-113	"Undefined header"	Command not recognized error	Check programming manual for correct command syntax
-120	"Numeric data error"	Data received is not a number	Check programming manual for correct command syntax
-121	"Invalid character in number"	Number received contains non-numeric character(s)	Check programming manual for correct command syntax
-123	"Exponent too large"	Exponent in number exceeds limits	Check programming manual for correct parameter range
-128	"Numeric data not allowed"	Number received when number is not allowed.	Check programming manual for correct command syntax
-168	"Block data not"	Block data was sent.	Check programming manual for

Number	Message String	Cause	Remedy
	allowed"		correct command syntax
-200	"Execution error"	Command could not be executed	Command may be inconsistent with mode of operation.
-201	"Invalid while in local"	Command issued but unit is not in remote state	Put instrument in remote state before issuing GPIB commands.
-203	"Command protected"	Command is locked out	Some commands are supported by the unit but are locked out for protection of settings and are not user accessible.
-210	"Trigger error"	Problem with trigger system.	Unit could not generate trigger for transient execution or measurement.
-211	"Trigger ignored"	Trigger request has been ignored.	Trigger setup incorrect or unit was not armed when trigger was received. Check transient system or measurement trigger system settings.
-213	"Init ignored"	Initialization request has been ignored	Unit was told to go to armed state but was unable to do so. Could be caused by incorrect transient system or measurement acquisition setup.
-220	"Parameter error"	Parameter not allowed.	Incorrect parameter or parameter value. Check programming manual for allowable parameters
-221	"Setting conflict"	Requested setting conflicts with other setting in effect.	Check other settings.
-222	"Data out of range"	Parameter data outside of allowable range.	Check programming manual for allowable parameter values
-223	"Too much data"	More data received than expected	Check programming manual for number of parameters or data block size
-224	"Illegal parameter value"	Parameter value is not supported	Check programming manual for correct parameters
-226	"Lists not same length"	One or more transient lists programmed has different length.	All lists must be of same length or transient cannot be compiled and executed.
-241	"Hardware missing"	N/A	N/A
-254	"Media full"	No storage space left to save settings or data.	Delete other settings or data to make room.
-255	"Directory full"	Too many directory entries	Delete one or more files from waveform memory to make room.
-256	"File name not found"	File requested not in directory	Check file directory for waveform names present.
-257	"File name error"	Incorrect filename	Too many or non ASCII characters used in file definition.
-258	"Illegal variable name"	Variable name illegal.	Use ASCII characters only
-300	"Device specific error"	Hardware related error	Check hardware for proper operation.
-311	"Memory error"	memory checksum	May be the result of incomplete data



Number	Message String	Cause	Remedy
		error.	download. Check interface and try downloading data again. Successful download may clear this error condition.
-314	"Save/recall memory lost"	User setup register contents lost	Store setup in same register again.
-315	"Configuration memory lost"	Hardware configuration settings lost.	Contact Argantix service department at <a href="mailto:support@argantix.com">support@argantix.com</a> to obtain instructions on restoring configuration data.
-330	"Self-test failed"	Internal error	Contact Argantix service department at <a href="mailto:support@argantix.com">support@argantix.com</a>
-350	"Queue overflow"	Message queue full.	Too many message. Read status using SYST:ERR query until 0, "No Error" is received indicating queue empty.
-400	"Query error"	Unable to complete query.	Check programming manual for correct query format and parameters
-410	"Query INTERRUPTED"	Query issued but response not read.	Check application program for correct flow. Response must be read after each query to avoid this error.
-420	"Query UNTERMINATED"	Query incomplete.	Check for terminator after query command.
-430	"Query DEADLOCKED"	Query cannot be completed	Check application program for multiple queries
-440	"Query UNTERMINATED"	Query incomplete.	Check for terminator after query command.
1	"Output volt fault"	Output voltage does not match programmed value.	Load exceeds current limit and unit is in Constant Voltage (CV) mode of operation. Reduce load or increase CL setting Output voltage is driven above programmed voltage by external influence (Load, voltage kickback, etc.)
2	"Current limit fault"	Current limit exceeded.	Load exceeds current limit and unit is in Constant Voltage (CV) mode of operation. Reduce load or increase CL setting
3	"Temperature fault" or "Temperature / Volt Sense fault"	Temperature of heat sink too high or volt sense connection problem.	Reduce load. Ensure proper airflow and exhaust clearance. Check fan(s) for operation. Check voltage sense connection.
4	"External sync. error"	Could not sync to external sync signal.	External sync signal missing, disconnected or out of range.
5	"Initial memory lost"	Power on settings could not be recalled.	Save power on settings again to overwrite old content.
6	"Limit memory lost"	Hardware configuration settings lost.	Contact Argantix service department at <a href="mailto:support@argantix.com">support@argantix.com</a> to obtain instructions on restoring configuration data.
7	"System memory lost"	Memory corrupted.	Recycle power.

Number	Message String	Cause	Remedy
8	"Calibration memory lost"	Calibration data lost.	Contact Argantix service department at <a href="mailto:support@argantix.com">support@argantix.com</a> to obtain instructions on restoring calibration data or recalibrate unit.
13	"Missing list parameter"	One or more transient list parameters missing.	Check programmed lists.
14	"Voltage peak error "	Peak voltage exceeds internal bus voltage	This error may occur when selecting user defined wave shapes with higher crest factors. Reduce programmed RMS value.
15	"Slew time exceed dwell"	Time needed to slew to final value is less than dwell time.	Check dwell times in transient list settings. Increase dwell time or change slew rate for affected parameter.
16	"Illegal during transient"	Operation requested not available while transient is running.	Wait till transient execution is completed or abort transient execution first.
17	"Output relay must be closed"	Operation not possible with open output	Close output before attempting operation. E.g. transient execution requires output relay to be closed.
18	"Trans. duration less than 1msec"	Dwell time below minimum or 1 msec	Increase dwell time to at least 1 msec.
19	"Clock and sync must be internal"	Operation not possible with external clock	N/A
20	"Input buffer full"	Too much data received.	Break up data in smaller blocks.
21	"Ambient temperature exceed limit"	Ambient air temperature measured at controller exceeds operating range..	Improve airflow / circulation or use climate control to keep ambient temperature below 40° C. Output will trip off. (For firmware older than 1.20, this error message is not available).
23	"Input line under voltage"	AC Input line too low or missing phase.	Check AC Line input against input specifications or check for phase loss.
24	"Input line over voltage"	AC Input line too high.	Check AC Line input against input specifications.
25	"Over voltage protection trip"	Output exceeds OVP trip level.	Check OVP mode and setting.

Table 7-1: Bus Error Messages

## Index

---

*	
*OPT?	79
*TRG	81
*TST	82
*WAI	82
<b>A</b>	
<b>ABORT</b>	25
address	
GPIB	12
IEEE	12
ALC	45
ALC mode	45
<b>B</b>	
bus address	
setting	12
<b>C</b>	
cable	
RS232	13
CALibrate:MEASure:CURRent:TEMPerature[:FSCale] <NRf>	27
CALibrate:MEASure:CURRent[:AMBient]:ZERO <NRf>	27
CALibrate:MEASure:CURRent[:AMBient][:FSCale] <NRf>	27
CALibrate:MEASure:VOLTage:TEMPerature[:FSCale] <NRf>	28
CALibrate:MEASure:VOLTage[:AMBient]:ZERO <NRf>	28
CALibrate:MEASure:VOLTage[:AMBient][:FSCale] <NRf>	28
CALibrate:PASSword	32
CALibrate:SAVE	32
CALibrate[:SOURce]:CURRent:EXTernal:10V <NRf+>	29
CALibrate[:SOURce]:CURRent:EXTernal:5V <NRf+>	29
CALibrate[:SOURce]:CURRent:EXTernal:RESistance <NRf+>	30
CALibrate[:SOURce]:CURRent:INTernal <NRf+>	30
CALibrate[:SOURce]:CURRent:OFFSet <NRf+>	30
CALibrate[:SOURce]:CURRent:RANGe <NRf+>	30
CALibrate[:SOURce]:VOLTage	
INTernal <NRf+>	31
CALibrate[:SOURce]:VOLTage:EXTernal:10V <NRf+>	31
CALibrate[:SOURce]:VOLTage:EXTernal:5V <NRf+>	30
CALibrate[:SOURce]:VOLTage:EXTernal:RESistance <NRf+>	31
CALibrate[:SOURce]:VOLTage:RANGe <NRf+>	31
CALibrate[:SOURce]:VOLTage:OFFSet <NRf+>	31
CLS	77
command	
subsystem	25
commands	
bus	25
common	21
Connect	13
Conventions	19
CURRent	46

CURRent:HIGH ..... 47  
 CURRent:LOW ..... 47  
 CURRent:MODE ..... 47  
 CURRent:PROTection:DELAy ..... 48  
 CURRent:PROTection:STATE ..... 48  
 CURRent:REFerence ..... 48  
 CURRent:SLEW:MODE ..... 49  
 CURRent:SLEW:TRIGered ..... 49  
 CURRent:SLEW[:IMMediate] ..... 48  
 CURRent:TRIGered[:AMPLitude] ..... 49

**E**

error messages ..... 110  
 ESE ..... 77  
 ESR? ..... 78

**F**

factory  
     default power on setting ..... 85  
 FETCh:ARRAy:CURRent[:DC]? ..... 36  
 FETCh:ARRAy:POWEr[:DC]? ..... 38  
 FETCh:ARRAy:VOLTAge[:DC]? ..... 38  
 FETCh:CURRent:AMPLitude:RESet ..... 39  
 FETCh:CURRent:AMPLitude:MAX? ..... 39  
 FETCh:CURRent[:DC]? ..... 39  
 FETCh:POWEr[:DC]? ..... 39  
 FETCh:VOLTAge[:DC]? ..... 40  
 format  
     waveform data ..... 36  
 formats  
     data ..... 24  
 front panel ..... 9  
 FT option ..... 45  
 function strobe ..... 42

**G**

GPIB ..... 12  
     address selection ..... 12  
     troubleshooting ..... 18

**H**

header  
     **optional** ..... 23  
     separator ..... 23  
**Header** ..... 20

**I**

IDN? ..... 78  
 IEEE-488 ..... 12  
     address selection ..... 12  
     troubleshooting ..... 18  
 initialization  
     changing ..... 85  
 INITiate:ACQuire ..... 33

INITiate:CONTInuous.....	34
INITiate[:IMMediate][:TRANsient] .....	33
installation .....	13
program .....	13
Installing	
software .....	14
Introduction .....	7
<b>K</b>	
KDCGUI .....	9
<b>L</b>	
LabView	
RS232C terminator.....	10
LIMit:CURRent .....	50
LIMit:VOLTagE .....	50
LIST:COUNT .....	51
LIST:CURRent:SLEW:POINts? .....	52
LIST:CURRent:SLEW[:RATE] .....	52
LIST:CURRent[:LEVel]:POINts? .....	52
LIST:CURRent[:LEVel][:AMPLitude].....	52
LIST:DWELI:POINts? .....	53
LIST:DWELI[:TIME] .....	53
LIST:REPeat:POINts? .....	53
LIST:REPeat[:TIME] .....	53
LIST:STEP .....	54
LIST:TTLTrg:POINts? .....	54
LIST:TTLTrg[:SWITch].....	54
LIST:VOLTagE:SLEW:POINts? .....	55
LIST:VOLTagE:SLEW[:RATE] .....	55
LIST:VOLTagE[:LEVel]:POINts? .....	55
LIST:VOLTagE[:LEVel][:AMPLitude].....	54
<b>M</b>	
MEASure:ARRay:CURRent[:DC]? .....	36
MEASure:ARRay:MODE.....	36
MEASure:ARRay:POWer[:DC]? .....	38
MEASure:ARRay:VOLTagE[:DC]? .....	38
MEASure:CURRent:AMPLitude:RESet .....	39
MEASure:CURRent:AMPLitude:MAX? .....	39
MEASure:CURRent[:DC]? .....	39
MEASure:POWer[:DC]? .....	39
MEASure:VOLTagE[:DC]? .....	40
Measurement Query	
*WAI.....	35
MIL704:ABNormal:VOLTagE:OVER .....	103
MIL704:ABNormal:VOLTagE[:UNDer] .....	103
MIL704:EMERgency:VOLTagE.....	104
MIL704:SState:VOLTagE:RIPple .....	103
MIL704:SState:VOLTagE[:LEVel] .....	102
MIL704:TRANsient:VOLTagE:HIGH .....	103
MIL704:TRANsient:VOLTagE[:LOW].....	103
MIL704:VERSion.....	102
MODE.....	56

**O**

OUTPut:IMP:REAL.....	41
OUTPut:IMPedance[:STATe].....	41
OUTPut:RI[:LEVel].....	42
OUTPut:TTLTrg:SOURce.....	43
OUTPut:TTLTrg[:STATe].....	42
OUTPut[:STATe].....	42

**P**

POWer.....	56
PULSe:COUNt.....	57
PULSe:DCYCLE.....	57
PULSe:HOLD.....	58
PULSe:PERiod.....	58
PULSe:WIDTh.....	58

**Q**

queries.....	21
Query	
*WAI.....	35

**R**

RCL.....	79
registration.....	18
remote inhibit.....	42
root.....	23
<b>Root</b> .....	20
RS232	
cable.....	13
RST.....	80

**S**

SAV.....	80
SCPI.....	19
message.....	19
program message.....	19
response message.....	19
SENSe:SWEEp:OFFSet.....	44
SENSe:SWEEp:TINTerval.....	44
Separator.....	23
setting	
GPIB address.....	12
Setup	
program.....	13
simulation mode.....	13
SOURce:ALC.....	45
SOURce:CURREnt:MODE.....	47
SOURce:CURREnt:PROTection:DELay.....	48
SOURce:CURREnt:PROTection:STATE.....	48
SOURce:CURREnt:REFerence.....	48
SOURce:CURREnt:SLEW:MODE.....	49
SOURce:CURREnt:SLEW:TRIGgered.....	49
SOURce:CURREnt:SLEW[:IMMediate].....	48
SOURce:CURREnt:TRIGgered[:AMPLitude].....	49

SOURce:CURRent[:LEVel][:IMMediate] [:AMPLitude][:DC] .....	46
SOURce:CURRent[:LEVel][:IMMediate]:HIGH .....	47
SOURce:CURRent[:LEVel][:IMMediate]:LOW .....	47
SOURce:MODE .....	56
SOURce:POWer .....	56
SOURce:VOLTagE:MODE .....	60
SOURce:VOLTagE:PROTection:DElay .....	61
SOURce:VOLTagE:PROTection:MODE .....	61
SOURce:VOLTagE:PROTection[:LEVel] .....	61
SOURce:VOLTagE:REFerence .....	62
SOURce:VOLTagE:SLEW:MODE .....	62
SOURce:VOLTagE:SLEW:TRIGgered .....	63
SOURce:VOLTagE:SLEW[:IMMediate] .....	62
SOURce:VOLTagE:TRIGgered[:AMPLitude] .....	63
SOURce:VOLTagE[:LEVel][:IMMediate]:HIGH .....	60
SOURce:VOLTagE[:LEVel][:IMMediate]:LOW .....	60
SOURce:VOLTagE[:LEVel][:IMMediate][:AMPLitude][:DC] .....	59
SRE .....	81
STATus :INSTrument:ISUMmary:CONDition? .....	69
STATus :INSTrument:ISUMmary:ENABle .....	69
STATus:INSTrument:ISUMmary :EVENT? .....	69
STATus:OPERation:CONDition? .....	65
STATus:OPERation:ENABle .....	65
STATus:OPERation:EVENT? .....	66
STATus:QUEStionable:CONDition? .....	67
STATus:QUEStionable:ENABle .....	67
STATus:QUEStionable:EVENT? .....	68
STB? .....	81
SYSTem:COMMunicate:GPIB:ADDRess .....	70
SYSTem:COMMunicate:SERial:BAUD .....	70
SYSTem:COMMunicate:SERial:BITS .....	71
SYSTem:COMMunicate:SERial:PARity .....	70
SYSTem:COMMunicate:SERial:SBITs .....	71
SYSTem:CONFigure[:BYTE] .....	71
SYSTem:ERRor? .....	71
SYSTem:ETIMe? .....	71
SYSTem:LOCal .....	72
SYSTem:PON .....	72
SYSTem:REMote .....	72
SYSTem:RWLock .....	72
SYSTem:TEMPerature .....	72
SYSTem:VERSion? .....	73

## T

terminator .....	24
RS232C .....	10
TRIG:STATe? .....	75
TRIGger:ACQuire:SOURce .....	74
TRIGger:ACQuire[:IMMediate] .....	74
TRIGger[:TRANsient]:SOURce .....	75
TRIGger[:TRANsient][:IMMediate] .....	75

## V

VB sample	
waveform conversion .....	36

VOLTage:MODE .....	60
VOLTage:PROTection:DELAy .....	61
VOLTage:PROTection:MODE .....	61
VOLTage:PROTection[:LEVel].....	61
VOLTage:REFerence.....	62
VOLTage:SLEW:MODE .....	62
VOLTage:SLEW:TRIGered .....	63
VOLTage:SLEW[:IMMediate] .....	62
VOLTage:TRIGered[:AMPLitude] .....	63
VOLTage[:LEVel][:IMMediate]:HIGH .....	60
VOLTage[:LEVel][:IMMediate]:LOW .....	60
VOLTage[:LEVel][:IMMediate][:AMPLitude][:DC].....	59

**W**

waveform	
data format modes.....	36