PROGRAMMING MANUAL
Agilent Technologies Electronic Load Family

Agilent Technologies

Agilent Part No. 06060-90005
Microfiche Part No. 06060-90006

Printed in USA: September, 1991
Update April, 2000

PRINTING HISTORY

The manual printing date and part number indicate the current edition. Reprints between editions will have the same printing date and may include change pages with corrections or additions to be made to the manual by the user.

New editions of this manual will have a new printing date and, in some cases, may have a new part number. The new edition will include all changes and corrections made since the previous edition.

Update - April, 2000

Edition 3 - September, 1991

Edition 2 - August, 1989

Edition 1- December, 1988

SAFETY GUIDELINES

The beginning of the Electronic Load Operating Manual has a Safety Summary Page. Be sure that you are familiar with the information on that page before programming the electronic load for operation from a controller.

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

1

Introduction

Purpose

The purpose of this guide is to enable you to use HPSL commands to remotely control your Agilent Technologies electronic load from a controller using HPSL programming language. It is assumed that the following has been done:

- The electronic load has been installed and is operating normally from its front panel.
- The controller has been connected to the electronic load and the electronic load's GPIB address has been set.

Note The electronic load GPIB address cannot be set by program. It must be set from the front panel of the electronic load (refer to the *Installation* chapter of your electronic load *Operating Manual*).

Documentation

Important Read your electronic load Operating Manual before using this guide.

Supplied

Every Electronic Load comes with the following documentation:

Operating Manual	Installation and Basic Operating Instructions, including local front-panel operation. *Be sure to read that document first.*
Programming Reference Guide	This guide for remote operation from a controller.

Recommended

The following reference documents are recommended:

- [1]Tutorial Description of the General Purpose Interface Bus
  - Highly recommended for those not experienced with IEEE 488.1 and 488.2.
- [2]ANSI/IEEE Standard 488.2-1987
  - The source document for programming via IEEE 488.1 and IEEE 488.2.

[1]Contact your local Agilent Sales office
[2]Contact Institute of Electrical and Electronics Engineers, 345 E. 47 Street. New York, NY 10017, USA

How To Use This Guide

Chapter	Synopsis
2 - Introduction to HPSL	The basics of HPSL to help you understand the terminology and diagrams in *Chapter 4.*
3 - Introduction to Programming	How to understand the command tree diagram and construct typical operating programs.
4 - Language Dictionary	An alphabetically ordered description of all electronic load HPSL commands.
5 - Status Reporting	An explanation of how the electronic load status registers are affected by the HPSL programming statements.
Index	

What You Should Already Know

This guide does not assume that you know anything about HPSL or are a programmer. It is assumed that you do know:

- the basics of the General Purpose Interface Bus (GPIB).
- how to send and receive ASCII data to and from a GPIB instrument (or where, in your computer and GPIB interface documentation, to find instructions to do this).
- how to incorporate the HPSL statements as ASCII strings within output and input statements of the programming language you are using.
- the basic operating principles of the electronic load as explained in Chapters 2 and 5 of your electronic load *Operating Manual*.

GPIB Capabilities Of The Electronic Load

The GPIB capabilities of a typical electronic load are listed in Table 1-1.

Note Refer to the General Information chapter of your electronic load Operating Manual for its exact capabilities.

Table 1-1. GPIB Capabilities of Electronic Loads

GPIB Capabilities	Response	Interface Function
Talker/Listener	All electronic load functions except for setting the GPIB address are programmable over the GPIB. The electronic load can send and receive messages over the GPIB. Status information is sent using a serial poll. Front panel annunciators indicate the present GPIB state of the electronic load.	AH1, SH1, T6. L4
Service Request	The electronic load sets the SRQ line true if there is an enabled service request condition. Refer to *Chapter 5 - Status Reporting* for more information.	SR1
Remote/Local	In local mode, the electronic load is controlled from the front panel but will also execute commands sent over the GPIB. The electronic load powers up in local mode and remains in local mode until it receives a command over the GPIB. Once the electronic load is in remote mode the front panel RMT annunciator is on, all front panel keys (except Local) are disabled, and the display is in normal metering mode. Pressing Local on the front panel returns the electronic load to local mode. Local can be disabled using local lockout so that only the controller or the power switch can return the electronic load to local mode.	RL1
Device Trigger	The electronic load will respond to device trigger function.	DT1
Device Clear	The electronic load responds to the Device Clear (**DCL**) and Selected Device Clear (**SDC**) interface commands. They cause the electronic load to clear any activity that would prevent it from receiving and executing a new command (including ***WAI** and ***OPC?**). **DCL** and **SDC** do not change any programmed settings.	DCL, SDC

Introduction To HPSL

What Is HPSL?

HPSL is a system programming language developed by Agilent Technologies for controlling instrument functions. HPSL is intended to function with *standard* GPIB hardware. HPSL conforms to the *IEEE 488.2 Standard Digital Interface for Programmable Instrumentation*. This standard provides codes, formats, protocols, and common commands not defined in the original *IEEE 488.1* standards. Unless you intend to do some very intricate programming, you need not be expert in the *IEEE 488.2* standard, although it is a good reference document to have available.

Note TMSL (Test and Measurement Systems Language) is a later version of HPSL that has been made available outside of Agilent Technologies for industry use. Although it is very similar, the HPSL used in the electronic loads may not be totally compatible with TMSL.

HPSL Statements

HPSL statements are instrument control commands and queries. A command statement sends an instruction to the electronic load and a command query requests information from the electronic load.

Simple Command Statements

The simplest command statement consists of a command, or keyword, usually followed by a parameter or data:

VOLT 25	*Simple command statements*
CURR 50	
TRIG	

Compound Command Statements

When two or more keywords are connected by colons (2:), it creates a compound command statement. The last keyword usually is followed by a parameter or data.

VOLT:SLEW 1000	*Compound common statements*
CURR:RANG 6	
TRIG:SOUR BUS	

Simple Command Queries

The simplest command query consists of a keyword followed by a question mark:

VOLT?	*Simple command queries*
CURR? CHAN?	

Compound Command Queries

When two or more keywords are connected by colons and followed by a question mark, it creates a compound query statement.

VOLT:TRIG? *Compound command queries*
CURR:PROT? MEAS:POW?

HPSL Keywords

Keywords (also known as "Instrument Control Headers") are recognized by the electronic load's decoder, or "parser". Each keyword is intended to be descriptive of the statement function. Refer to Figure 4-2 in *Chapter 4 - Language Dictionary* for a quick look at all the electronic load keywords.

Forms of Keywords

Every keyword has two forms:

Long Form	The word is spelled out completely to identify its function. STATUS, RESISTANCE, and TRIGGER are long form keywords.
Short Form	The word contains only the first three or four letters of the long form. STAT, RES, and TRIG are short form keywords.

Short forms are constructed according to the following rules:

- If the keyword consists of *four or fewer* letters -
  - then all the letters are used

- If the keyword consists of *five or more* letters -
  - and the fourth letter IS NOT a vowel (*a,e.i,o,u*)
    - then the *first four letters are used*

  - and the fourth letter is a vowel
    - then only the *first three letters are used*

Note The short form provides the fastest program execution.

Keyword Conventions

In keyword definitions and diagrams in this guide, the short form part of each keyword is emphasized in **boldface** UPPER-CASE letters to help you remember it.

- **TRIG**ger
- **IMM**ediate
- **RES**istance
- **SHOR**t

The HPSL parser (decoder) is not sensitive to case. It will accept *Trig, trig, trigger, TRIGGER, triGgER,* etc. Regardless of which form you use, you must spell out all the letters. For example, RESI or TRI will not be recognized.

Keyword Parameters

Parameters are data values that the parser expects to find after certain keywords. All data programmed to or returned from the electronic load is ASCII. The data may be numerical data or character strings. HPSL uses the parameter forms in Section 7 of *IEEE 488.2 Standard Digital Interface for Programmable Instrumentation* with the additions described here.

Numerical Data Formats HPSL accepts the first four numerical data types listed in Table 2-1 and described in Section 7 of *IEEE 488. 2 Standard Digital Interface for Programmable Instrumentation.* In addition, HPSL recognizes an expanded form of decimal numeric value known as < **NRf** > . This allows the characters *MIN* and *MAX* to be entered for the minimum and maximum values that the parameter can be set to under the existing operating conditions.

Table 2-1. Numerical Data Formats

Symbol	Data Form
NR1	Digits with no decimal point. The decimal point is assumed to be to the right of the least-significant digit. For example, **273, 0273**
NR2	Digits with a decimal point. E.g.,**273., 27.3, .0273**
NR3	Digits with a decimal point and an exponent. E.g., **2.73E+2, 2.73E-2**
NRf	Flexible decimal form that induces NR1 or NR2 or NR3. E.g., **273, 27.3, 2.73E+2**
NRf +	Expanded decimal form that includes NRf and MIN,MAX. E.g., **273, 27.3, 2.73E-2, MIN, MAX. MIN** and **MAX** are the minimum and maximum limit values for the parameter and are implicit in the range specification for the parameter.

Numerical Data Suffixes and Multipliers. Numeric data may be followed by a suffix that dimensions the data. A suffix may be preceded by a multiplier. Section 7 of *IEEE 488.2 Standard Digital Interface for Programmable Instrumentation* describes the approved data suffixes and multipliers. Where no suffix is entered, the dimension is implied by the syntax of the command. The electronic loads make use of the suffixes and multipliers listed in Table 2-2 and Table 2-3. Note the special consideration given to the multiplier for mega. In most cases, mega is represented by *MA.* However, there are exceptions made for megahertz *(MHZ)* and megohm *(MOHM).* In only these two cases, *M is* understood to be lE + 6. Do not confuse the mega multiplier *MA* with the combination suffix and multiplier *MA*, which represents milliamperes (lE-3).

Table 2-2. Suffix Elements

Class	Preferred Suffix	Secondary Suffix	Referenced Unit
Current	A		Ampere
Frequency	Hz		Hertz
		MHZ	Megahertz
Resistance	OHM		ohm
		MOHM	Megohm
Time	s		Second
Amplitude	V		Volt
Power	W		Watt
Slew Rate	A/s		Amperes/Second
	V/s		Volts/Second

Table 2-3. Most-Used Suffix Multipliers

Multiplier	Mnemonic	Definition
1E6	MA	mega
1E3	K	kilo
1E-3	M	milli
1E-6	U	micro
1E-9	N	nano

Note You may construct compound suffixes of multipliers and elements. For example: 1 KHz for 1000 Hz; 1 A/μs for 1000000 A/s.

Numerical Data Conventions. In this guide, numerical data types are shown in emphasized text within angle brackets, such as < **NR1 >** or **< NRf** > . On drawings, numerical data appears within boxes $\boxed{<\text{NRf}>}$.

Data suffixes are shown inside brackets within boxes $\boxed{\text{suffix}}$. The brackets around the suffix indicate that the entry is optional. That is because there is a default suffix for the data that accompanies each command.

Character Data Formats. For command statements, the **< NRF + >** data format permits entry of required characters. For query statements, character strings may be returned in either of the forms shown in Table 2-4, depending on the length of the returned string.

Table 2-4. Query Character String Formats

Symbol	Character Form
crd	Character Response Data. Permits the return of up to 12 characters.
aard	Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type is an implied message terminator (refer to “Separators and Terminators”).

Character Data Conventions. In this guide, character string parameters are emphasized similar to keywords, such as **ON**, **OFF**, and **CONT**inuos. This applies both to text and drawings.

Separators and Terminators

In addition to keywords and parameters, HPSL program statements require the following:

Data Separators. Data must be separated from the previous command keyword by a space. This is shown in examples as a space (**VOLT 25**) and on diagrams by the letters *SP* inside a circle.

Keyword Separators. Keywords (or headers) are separated by a colon (:), a semicolon (;), or both. For example:

- **INP:SHOR**
- **MEAS:CURR?;VOLT?**
- **CURR 25;:VOLT 50**

Important Proper use of the (:) and the (;) is very important to the construction of command messages. This is explained in *Chapter 3 - Introduction to Programming*.

Program Line Terminators. A terminator informs HPSL that it has reached the end of a statement. Normally, this is sent automatically by your GPIB programming statements. The termination also occurs with other terminator codes, such as EOI. In this guide, the terminator is assumed at the end of each example line of code. If it needs to be indicated, it is shown by the symbol < **nl** >, which stands for “new line”' and represents the ASCII coded byte 0A hexadecimal (or l0 decimal).

Common Commands

Common statements are not derived from HPSL but are generic commands and queries defined by the IEEE 488.2 standard. The following examples are common statements:

- ***RST**
- ***IDN?**
- ***TRG**

Common statements are executed independently of HPSL statements. Their relationship to HPSL statements is described more fully in “Chapter 3”. The function of each common statement is summarized in *Chapter 4 - Language Dictionary* and fully described in Section 10 of *IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.*

3

Introduction To Programming

Types Of Commands and Queries

The electronic load responds to two types of commands and queries, *Common* and *Root.* Common commands were introduced in *Chapter* 2-*Introduction to HPSL* and are relatively simple to use. The root commands are organized in a hierarchy that is best shown via a command tree diagram.

Understanding The Command Tree

Figure 4-2 in *Chapter 4-Language Dictionary* is a tree diagram of all the root commands for the electronic load. Notice the following:

- The originating point of the diagram is at the *root.* In this case, the diagram resembles an inverted tree. "Root" is not a command, but the origin for all commands.

- The root divides into two major branches-*Channel-Specific* commands and *Channel-Independent* commands. Both types are accepted by all electronic loads.

- Each major branch divides into several branches. The **CURR**ent commands are a branch. The **VOLT**age commands are another branch. So are the **TRIG**ger commands.

- Some keywords are within brackets [ ]. These are *implied* keywords. Implied keywords are optional, but you may want to use them in certain situations (See *Implied Keywords,* later in this chapter).

Note For fastest program execution, omit implied keywords.

- Commands followed by ? are queries. They cause the electronic load to store information in its output buffer from where it can be read by the controller over the GPIB.

Understanding A Typical Branch

Here are the keywords for the **RES**istance branch as they appear in *Chapter 4-Language Dictionary*:

Command and Function

RESistance[:**LEV**el][:**IMM**ediate]
Specify input resistance for **RES**istance mode

RESistance[:**LEV**el]:**TRIG**gered
Preset input resistance level pending trigger occurrence

RESistance:**RANG**e
Specify full-scale resistance input range

RESistance:**TLEV**el
Specify resistance level for **TRAN**sient function

Note Ignore the meanings of these commands for now. All keywords are defined in the ***Language Dictionary*** and command functions are explained in the electronic load ***Operating Manual***.

Figure 3-1 shows the **RES**istance commands, which form a typical branch that forms a ’’subtree" of its own. You can see that the **RES**istance branch has three subbranches; **LEV**el, **RANG**e, and **TLEV**el. When it reaches the end of a branch, the parser expects a parameter, a question mark (that identifies the keyword as a query), a semicolon or semicolon-colon combination, or an end-of-line terminator.

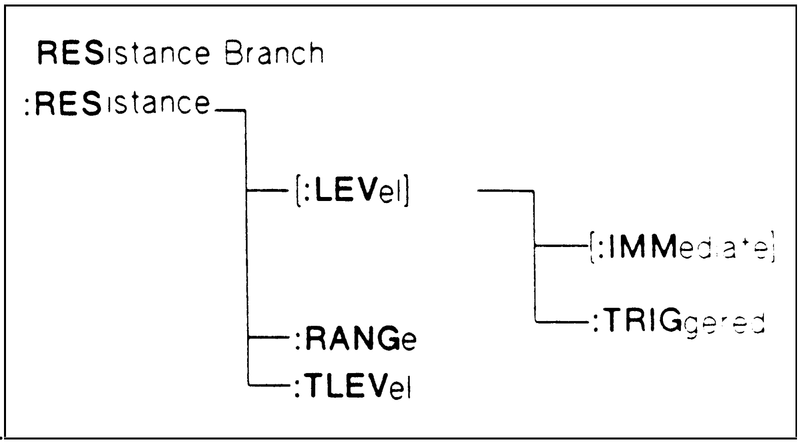


Figure 3-1. RESistance Branch Subtree

Figure 3-2 is the syntax diagram for the **RES**istance branch. You can still identify the tree structure, although it runs from left-to-right instead of from top-to-bottom. Note the following symbols that were discussed in *Chapter 2-Introduction to HPSL.*

- Keywords with the short form shown in bold-faced capital letters. Implied keywords are not within brackets in diagrams.
- Spaces shown as "SP" within circles.
- Boxes showing the form (NRf +) of the parameter.
- Boxes showing optional suffixes. Multipliers are not shown.
- Question marks that convert a command into a query
- Colons (:) that precede each keyword. They are important and are discussed later in more detail.

Traversing The RESistance Branch

From Figure 3-2 note that there are two implied keywords that you can usually ignore. This makes two of the commands very simple. If you enter:

RES 1.5

you will send the electronic load an immediate resistance level command. The command actually is:

RES:LEV:IMM 1.5

but the parser assumes that the two implied keywords are there. For the same reason, you can query the immediate resistance value with:

RES?

Note When sending a query, do not enter a space between the keyword and the question mark.

The newline character **< nl >** or EOI terminator sends the parser back to the root level. Many controllers automatically send this character at the end of each program output string. To program the resistance range and the **TLEV**el value, you could send:

RES:RANG 1000 < nl >
RES :TLEV 5000 < nl >

Note There is an alternate way to do this, which is shown under *"Traversing the Command Tree."'*

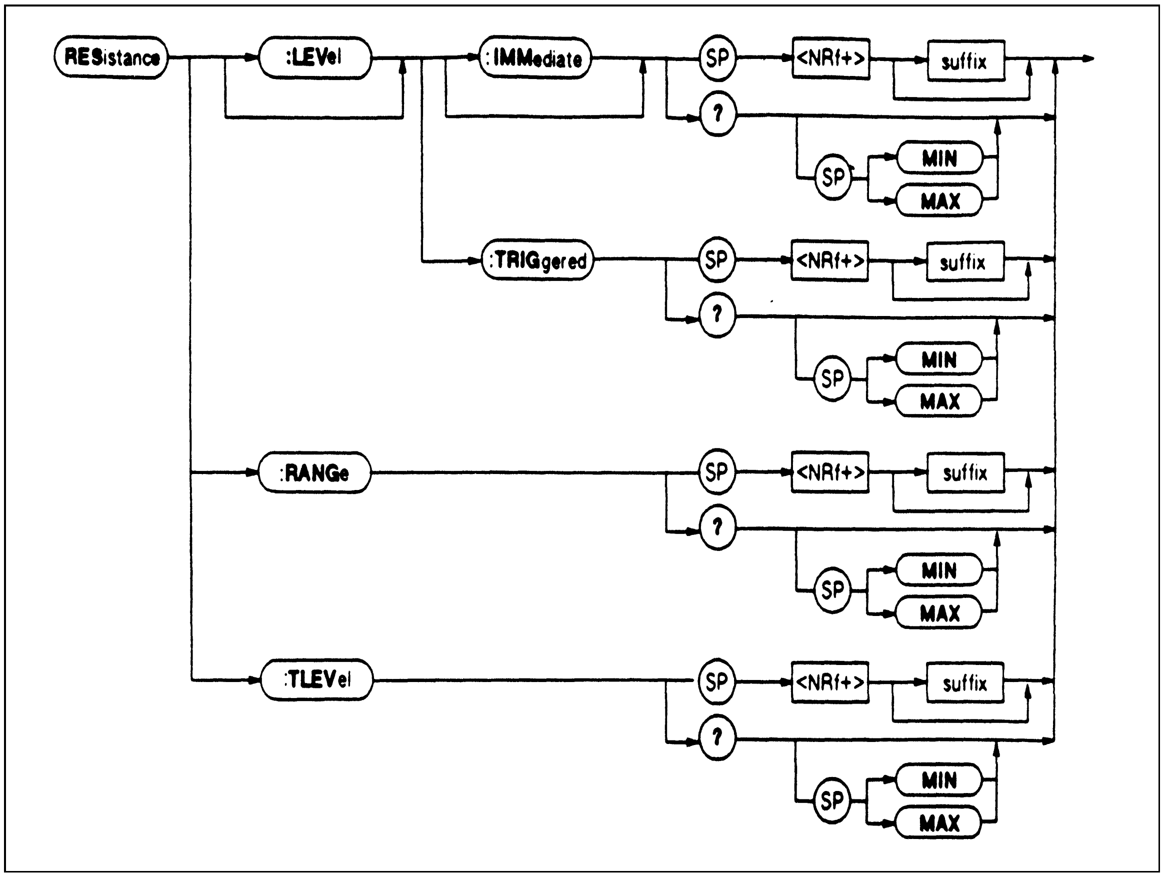


Figure 3-2. RESistance Branch Syntax Diagram

Using the NRf+ Format

Referring to Figure 3-2, note that all parameters are of the **<NRf+ >** type. This allows you to easily set a numeric parameter to its maximum or minimum value. If you wanted to increase the immediate resistance of the presently selected range to its maximum value, you could send:

RES MAX < nl >

Note *MAX* and *MIN* are the maximum and minimum values allowed in the **present operating mode** of the electronic load. For the electronic load this generally means the limits within the present range.

MAX and ***MIN*** may also be used with queries to find the maximum and minimum permitted settings of the present mode. For example:

RES? MAX	*Returns the maximum permitted value of the present range*

Traversing The Command Tree

Note The HPSL parser traverses the command tree as described in ***Appendix A of IEEE 488.2 Standard Digital Interface for Programmable Instrumentation*.** The Enhanced Tree Walking Implementation given in that appendix is not implemented in HPSL . The simplified explanation given here is sufficient for using the electronic load command set in most applications.

Use of the Colon

When you examined the **RES**istance branch, you noticed the colon (:) that separated keywords from each other. A colon represents a change in branch level. **ROOT** is the highest level. Whenever you enter a (:), the parser expects it to be followed by a command of the next lower level. For example:

INP:PROT:CLE
CURR:LEV:TRIG

INP and **CURR** are root-level commands. **PROT** and **LEV** are first branches, and **CLE** and **TRIG** are second branches. Each (:) *instructs the parser to move down to the next branch.*

Note A colon after a keyword always moves the parser down, never up the command tree.

If you enter **INP:PROT:CLE:,** you will get an error because the parser expects to find another keyword after the last (:). In this example, you will also get an error if you enter **INP:PROT** because another keyword is required after **PROT**. However, a command like **MODE:RES** is o.k. because no keyword or parameter is expected after **RES**. You will know what is required in each case by referring to the *Language Dictionary*.

Use of the Semicolon

The semicolon (;) is a "back-up" command. It instructs the parser to return to the previous colon. Figure 3-3 illustrates how the semicolon moves the parser backwards.

Note The semicolon by itself can back the parser up to a colon only within the same branch.

The semicolon allows you to combine command *statements* on one line to create command *messages***.**

VOLT:SLEW 5000 < nl > *Statements are on separate lines*

VOLT:TLEV 55 < nl >

VOLT: SLEW 5000;TLEV 55 < nl > *Combined statements form a message*

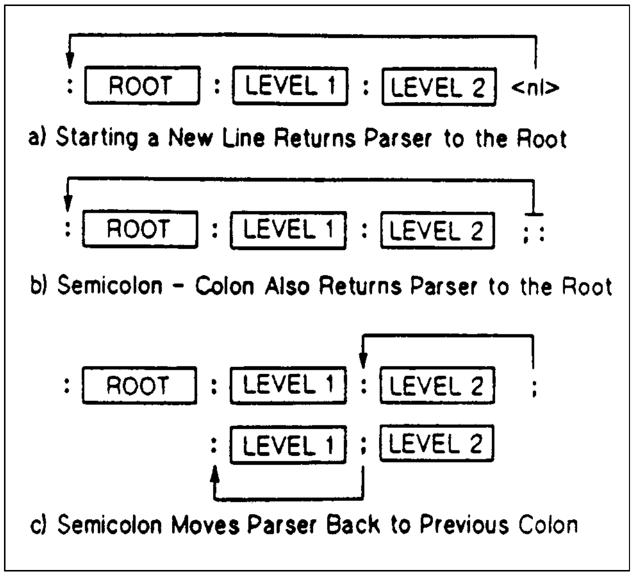

: ROOT : LEVEL 1 : LEVEL 2 <nl>
a) Starting a New Line Returns Parser to the Root
: ROOT : LEVEL 1 : LEVEL 2 ; :
b) Semicolon - Colon Also Returns Parser to the Root
: ROOT : LEVEL 1 : LEVEL 2 ;
: LEVEL 1 ; LEVEL 2
c) Semicolon Moves Parser Back to Previous Colon

Figure 3-3. What the Semicolon Does

Note There is no single command to move the parser back two colons. In example a above, backing up from Level 2 to Level 1 requires a return to the root.

Getting Back to the Root

To go from a command in one branch to a command in another branch, you must first return to the root. You can do this by:

- entering a new-line character. This is symbolized by (<**nl**>) and can be any control character that starts a new line, such as:
- linefeed (LF).
- an end-of-line (EOI).
- entering a semicolon followed by a colon (;:). This instructs the parser to return to the root.

Looking at Figure 4-2 in the *Language Dictionary*, suppose you wanted to set two trigger levels; the **CURR**ent level to 15.5 and the **VOLT**age level to 25.5. Either of the following commands would do this.

CURR:TRIG 15.5 < nl >
VOLT:TRlG 25.5 < nl >
or
CURR:TRIG 15.5;:VOLT:TRlG 25.5 < nl >

Similarly, the following query would return the present values of current, power, and voltage and the state of the output port.
MEAS:CURR?;POW?;VOLT?;:PORT0? < nl >

Note The **< nl >** notation is assumed and will be omitted in later programming examples.

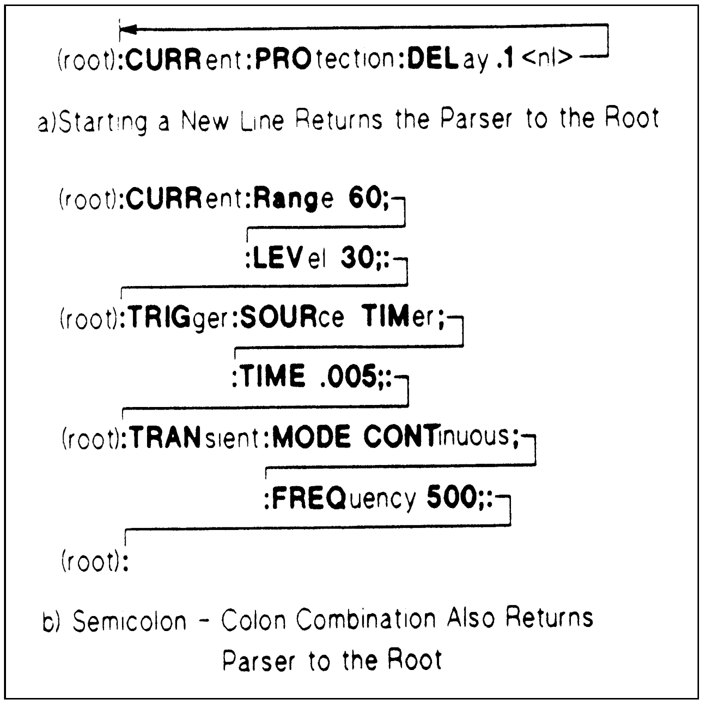

(root):CURRent:PROtection:DELay .1 <nl>
a)Starting a New Line Returns the Parser to the Root
(root):CURRent:Range 60;
:LEVel 30;:
(root):TRIGger:SOURce TIMer;
:TIME .005;:
(root):TRANsient:MODE CONTinuous;
:FREQuency 500;:
(root):
b) Semicolon - Colon Combination Also Returns Parser to the Root

Figure 3-4. Returning the Parser to the Root

Implied Keywords

Keywords shown within brackets, such as **CURR[:LEV**el], are implied keywords. If they are omitted, the parser will execute them automatically.

How to Use Implied Keywords. Because [**LEV**el] is an implied keyword, the parser regards the following two commands as the same:

CURR:LEV 30
CURR 30

Under most circumstances, implied keywords are optional and you may omit them as in the above example. Sometimes you may choose to use them in order to make the semicolon move the parser in the desired way. Returning to our previous keyword diagram under Figure 3-2, note that you can set the immediate resistance value with:

RES .5

The parser automatically assumes that you want to program **LEV**el. If you wanted to program both **LEV** and **TLEV**el in one program line and sent:

RES .5:TLEV 1 *Incorrect parser positioning*

the parser would end up back at the root and you would get an error because there is no **TLEV**el command at the root. The correct statement would be:

RES:LEV .5;TLEV 1 *Correct parser positioning*
or
RES .5;RES:TLEV 1

By inserting the implied keyword in **RES:LEV .5**; you allowed the parser to interpret the (;) as a command to move back to the branch containing **RANG** and **TLEV**. Without **LEV** in the command, the parser would "find" only **RES, CURR**, **STAT** or other root-level commands.

HPSL Queries

You can program more than one query in a single line such as:

CURR?;RES?;VOLT? *Return present values of programmed current, resistance, and voltage*,

However, observe the following precautions:

- You must read back the results of the queries before sending another command line to the electronic load. Otherwise, a *Query Interrupted* error will occur and the returned data will be lost.
- Multiple queries must be separated by semicolons and some controllers may have problems interpreting this format. In this case, you must enter each query and its corresponding readback on a separate line.

HPSL Compatibility

The SOURce Implied Keyword

Referring to Figure 4-2 in the La*nguage Dictionary,* note that several of the Channel-Specific branches include the implied keyword [**SOUR**ce]. It is there to make electronic load programs compatible with other HPSL devices. Although the electronic load will accept it, you can omit [**SOUR**ce] and consider the Channel-Specific branch as connected directly to the root.

Aliases

Looking at Figure 4-2 you will notice that some electronic load commands will accept two keywords that perform the identical function. For example:

MODE|FUNCtion	*Examples of two commands that do the same thing*
INPut|OUTPut	
CHANnel|INSTrument	

These alternate keywords are called "aliases" and are supported by the electronic load in order to make it compatible with other HPSL instruments.

Value Coupling

When you program, you must be aware of the effect known as *value coupling.* Value coupling results when a command directed toward one parameter changes the value of another parameter. For example, the **CURR**ent branch includes the following keywords:

Command

CURRent[:**LEV**el][:**IMM**ediate]
CURRent:**RANG**e
CURRent:**TLEV**el

There is value coupling among the **RAN**ge**, LEV**el**,** and **TLEV**el commands. If a previously programmed **LEV**el value is outside a particular range, then changing to that range will affect the value of **LEV**el. There are several instances of value coupling among the electronic load commands and you should always check a command's description in the *Language Dictionary* to determine if it is value coupled to another command.

Common Commands

Common commands, while not part of the command tree, can be mixed in with regular commands. The electronic load responds to the Common commands and queries listed in Figure 4-1 of the *Language Dictionary*. You can mix Common commands in with regular programming statements; the Common command will be executed without affecting the position of the parser.

Programming Examples

The following programming examples are practical applications of the electronic load. Although they are in HP Series 300 Basic, the principles can be applied to any other version of BASIC or even to another language.

Battery Testing

The principal measurement of a battery's performance is its rated capacity. The capacity of a fully charged battery, at a fixed temperature, is defined as the product of the rated discharge current in amperes and the discharge time in hours, to a specified minimum termination voltage in volts (see Figure 3-5). A battery is considered completely discharged when it reaches the specified minimum voltage called the "end of discharge voltage" (EODV).

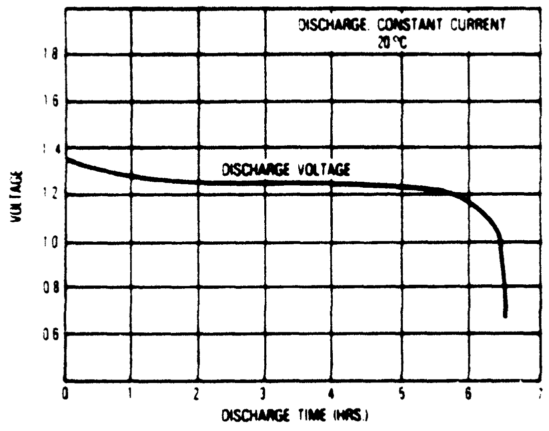


Figure 3-5. Typical Discharge Curve

In this example, the electronic load discharges three nickel-cadmium batteries to determine their discharge rates at a fixed temperature (see Figure 3-6). The batteries are connected in series so that when the EODV is reached, it is still above the minimum operating voltage of the electronic load. The EODV for nickel-cadmium batteries is typically 1.0 volts.

Power Supply Testing

A typical use for electronic loads when testing power supplies involves power supply burn-in. One of the problems associated with burn-in is what to do if the power supply fails before the test is over. One solution involves continuously monitoring the supply and removing the load if the supply fails during the test (see Figure 3-7).

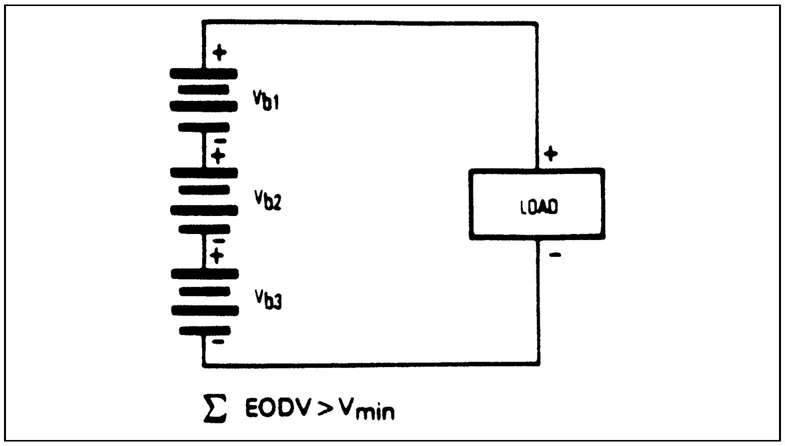


Figure 3-6. Batteries in Series

In this example, the Electronic Load is used to burn-in a power supply at its rated output current. Because the Electronic Load is operating in CC mode, if the power supply's output current drops below the rated output current during the test, the UNR (unregulated) condition will be set on the Electronic Load. This can be used to indicate that a failure has occurred on the power supply. If the unregulated condition persists for a specified time, the inputs of the Electronic Load are turned off.

The purpose of this example is not to illustrate power supply testing, but to explain how to program and use the status registers on the Electronic Load. The part of the program that runs the test simply monitors the supply at the rated output current for one hour and stops the test. You can replace this portion of the program with your own routine to test the power supply. Although SRQ (service request) is enabled to interrupt only on the UNR bit in this example, you can modify the program to interrupt on other conditions.

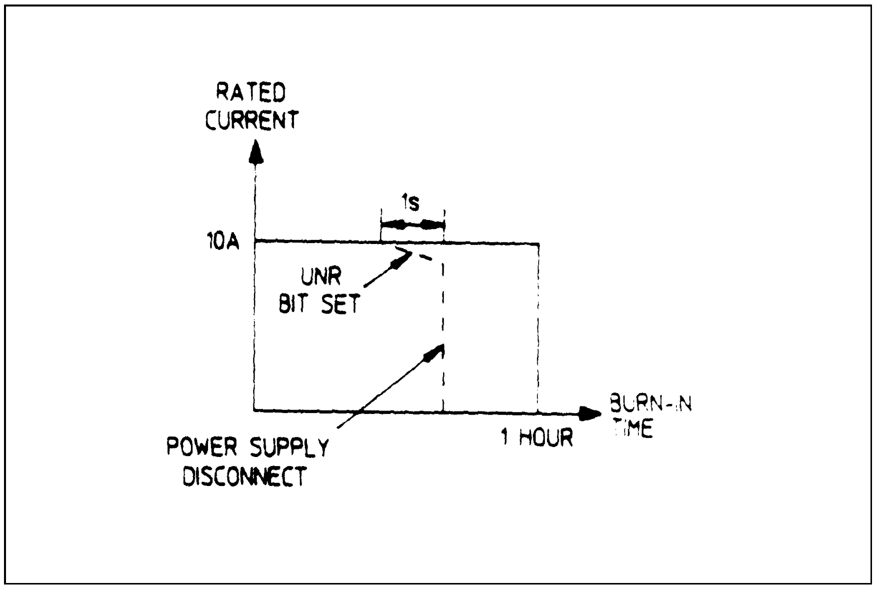


Figure 3-7. Typical Burn-In Test

Battery Test Example Program

```
l0      ! Battery Test Example Program
20      !
30      Eodv=l.0                  ! End of discharge voltage for single cell
40      Number_of_cells=3         ! Number of cells to be discharged in series
50      Discharge_at .05          ! Constant current discharge rate in amperes
60      !
70      OUTPUT 705;"INPUT OFF"                       ! Disables the inputs
80      OUTPUT 705;"MODE:CURRENT"                    ! Sets CC mode
90      OUTPUT 705;"CURRENT:LEVEL";Discharge_at      ! Sets the CC level
l00     OUTPUT 705;"INPUT ON"                        ! Enables the inputs
110      !
120     Start_time=TIMEDATE                          ! Records test start time
130     !
140     Start_test:                                  ! Starts test routine that
150     OUTPUT 705;"MEASURE:VOLTAGE?"                ! continuously measures and reads
160     ENTER 705;Sum_of_volts                       ! back the voltage and current
170     OUTPUT 705;"MEASURE:CURRENT?"                ! until batteries are completely
180     ENTER 705;Actual_current                     ! discharged
190     !
200     PRINT "Total cell voltage: ";Sum_of_volts
210     PRINT "Actual current: ";Actual_current
220     PRINT "Elapsed time in seconds: ";TIMEDATE-Start_time
230     !
240     IF Sum_of_volts>(Number_of_cells*Eodv) THEN GOTO Start_test
250     !    Checks if the total voltage is less than the
260     !    sum of the minimum cell voltages of all cells
270     !
280     OUTPUT 705;"INPUT OFF"                       ! Disables the inputs
290     !
300     END
```

Power Supply Test Example Program

```
l0      ! Power Supply Test Example Program
20      !
30      Current=10                               ! Load current in amperes
40      Burn_in_time=36000                       ! One hour burn-in time
50      !
60      ON INTR 7 GOSUB Srq_service              ! Set up interrupt linkage
70      ENABLE INTR 7;2                          ! Enable interrupts for SRQs
80      !
90      OUTPUT 705;"INPUT OFF"                   ! Disables the inputs
l00      OUTPUT 705;"*SRE 4"                     ! Enable SRQ (SRQ enable)
110     OUTPUT 705;"STAT:CSUM:ENAB 2"            ! Enable Chan 1 (channel summary)
120     OUTPUT 705;"STAT:CHAN:ENAB l024"         ! Enable UNR bit (channel status)
130     OUTPUT 705;"MODE:CURRENT"                ! Sets CC mode
140     OUTPUT 705;"CURRENT:LEVEL";Current       ! Sets the CC level
l50     OUTPUT 705;"INPUT ON"                    ! Enables the inputs
160     !
170     PRINT "Burn-in test started at ";TIME$(TIMEDATE)
180     !
190     FOR I=1 TO Burn_in_time                  ! Loop on wait You can write your
200         WAIT .1                              ! own power supply test routine and
210      NEXT I                                  ! insert it in this section
220     !
230     OUTPUT 705;"INPUT OFF"                   ! Disables the inputs at end of test
240     PRINT "Burn-in test complete at ";TIME$(TIMEDATE)
250     STOP
260     !
270      Srq_service                             ! Service request subroutine
280     Load_status=SPOLL(705)                   ! Conduct serial poll
290     IF BIT(Load_status, 6) THEN              ! Check if SRQ bit is set
300         GOSUB Check_unr
310     ELSE
320          PRINT "A condition other than UNR generated SRQ at ";TIME$(TIMEDATE)
330     END IF                                   ! You can also check the other bits
340     ENABLE INTR 7                            ! Re-enable interrupts before return
350     RETURN
360     !
370     Check_unr                                ! Check if UNR bit still set
380     WAIT 1                                   ! Wait 1 s before reading UNR bit
390     OUTPUT 705;"STAT:CHAN:COND?"             ! Read channel condition register
400     ENTER 705;Value
410     IF Bit(Value, l0)=0 THEN                 ! Return value for UNR bit only
420        OUTPUT 705;"*CLS"                     ! If 0, clear channel event register
430        PRINT "UNR was momentarily asserted at ";TIME$(TIMEDATE)
440     ELSE
450         OUTPUT 705;"INPUT OFF"               ! Disables the inputs
460         PRINT "UNR is asserted at ";TIME$(TIMEDATE);" Inputs are turned off"
470         STOP
480     END IF
490     RETURN
500     END
```

4

Language Dictionary

Introduction

This section gives the syntax and parameters for all the IEEE 488.2 common commands and HPSL commands used by the electronic loads. It is assumed that you are familiar with the material in Chapters 2 and 3, which explain the terms, symbols, and syntactical structures used here and provide an introduction to programming. You should also be familiar with the *Operation Overview and Remote Operation* chapters in the electronic load *Operating Manual* that was shipped with the electronic load. Those chapters explain how the electronic load functions and how to write simple programs to perform basic functions from a controller.

Because the versatility of HPSL allows such freedom in programming, the programming examples show just simple applications of the commands. With experience, you will find ways of combining simple statements into more complex compound ones, or forming iterations within compound statements. Because HPSL functions are the same in all electronic loads, the examples in this chapter are generic. If you send a command or query in a manner consistent with the syntax of your programming language, then the statement or query will always perform the specified function.

Keywords

Keyword explanations use the “long form” of the word, but the short form is used in the examples. If you have any concern that the meaning of a keyword in your program listing will not be obvious at some later time, then use the long form to help make your program self-documenting.

Parameters

Most commands require a parameter and most queries will return a parameter. The range for a parameter may vary according to the model of electronic load. For consistency, the examples and explanations use parameters for the Model Agilent 6060A Electronic Load. However, these examples and explanations are valid for any electronic load. Parameters for all current models can be found in Table 4-1 at the end of this chapter.

Related Commands

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

Order Of Presentation

All the electronic loads commands and queries are included in this dictionary. The dictionary is organized as follows:

- IEEE 488.2 common commands, in alphabetical order.
- Root level commands, in alphabetical order. These consist of:
    - Single commands.
    - Subsystems. The individual commands for a subsystem are listed in alphabetical order under the subsystem.

Common Commands

Introduction

Common commands are defined by the IEEE 488.2 standard to perform some of the basic instrument functions, such as identification, reset, determining how status is read and cleared, and how commands and queries are processed. Common commands are accepted and processed when they are sent as separate commands and also when they are included within program messages. Execution of a common command does not change the position of the parser in the program tree but leaves it in the same place it was before the common command was executed (refer to *Chapter* 2 *-Introduction to HPSL).* This does not mean that a common command has no effect on the rest of a programming message.

The electronic loads respond to the 13 required common commands that control internal operation, synchronization, status and event registers, and system data. Because they have full trigger capability, the electronic loads also respond to ***TRG**. In addition, the electronic loads accept six optional common commands. The description for each common command or query specifies if it affects status registers. In order to make use of this information, you must refer to *Chapter 5 - Status Reporting,* which explains how to read the status registers and use the information that they return.

Order of Presentation

Figure 4-1 shows the common commands and queries, which are presented in alphabetical order. 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.

*CLS

Clear Status Command

Type Device Status

Description This command causes the following actions (See *Chapter 5 - Status Reporting* for descriptions of all registers):

- Clears the following registers without affecting any corresponding Enable registers or Transition Filters:
  - Channel Status Event registers for all channels.
  - Channel Summary Event register.
  - Questionable Status Event register.
  - Standard Event Status Event register.
  - Operation Status Event register.
- Clears the Error Queue.
- Forces a previously executed ***OPC** command to appear as if it had been completed. It does not do this with the ***OPC?** command. (See ***OPC?** for more details).
- If ***CLS** immediately follows a program message terminator (<nl>), then the output queue and the MAV bit are also cleared.

Command Syntax ***CLS**

Parameters None

Query Syntax ***OPC** ***OPC?**

Syntax Diagram

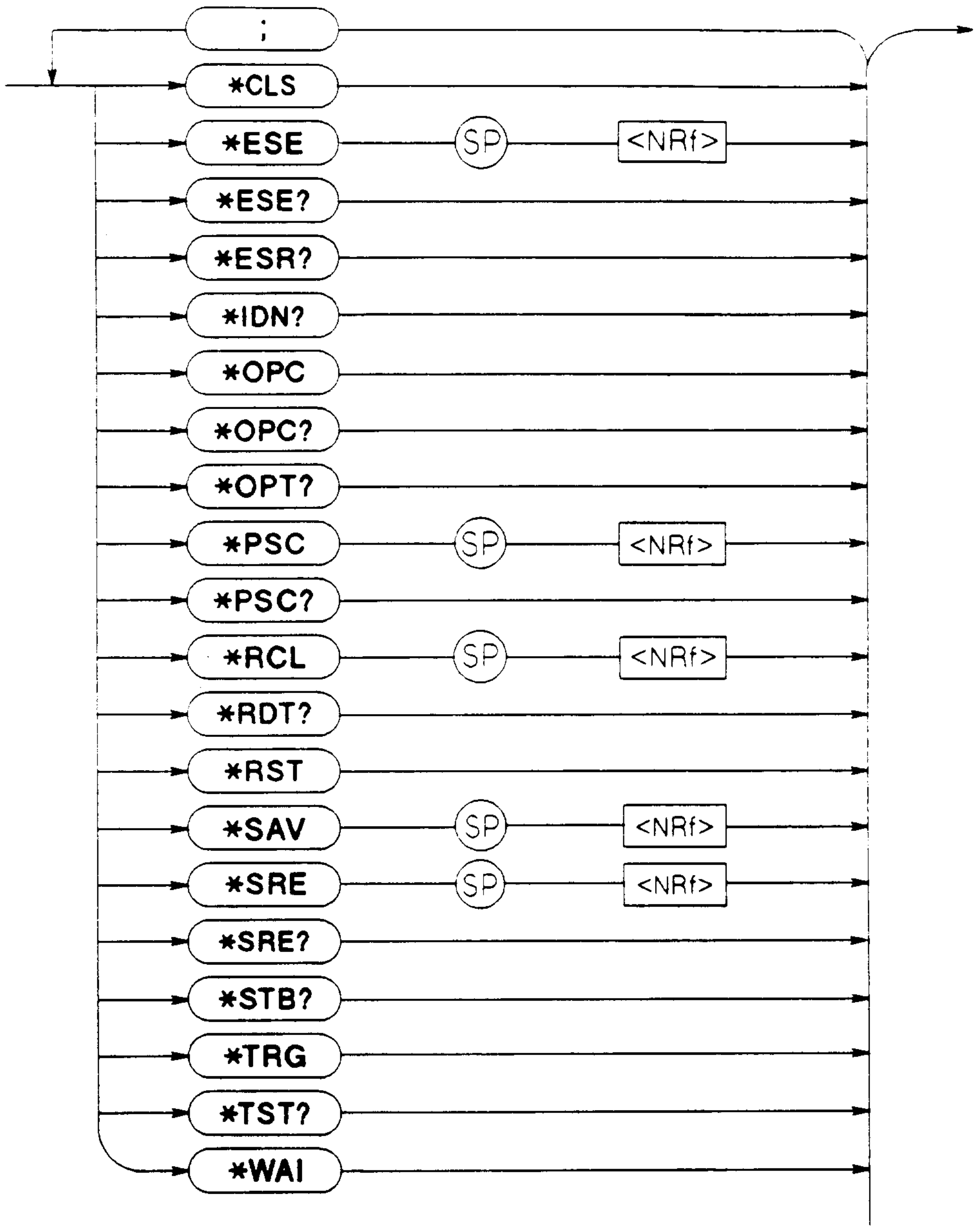


Figure 4-1. Common Commands Syntax Diagram

*ESE

Standard Event Status Enable Command/Query

Type Device Status

Description This command sets the condition of the Standard Event Status Enable register, which 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 Event register are logically ORed to cause the ESB (bit 5) of the Status Byte register to be set. See *Chapter 5 - Status Reporting* for descriptions of all three registers.

Command Syntax ***ESE <NRf>**

Parameters *0 to 255*

Suffix None

Example ***ESE 129** *Enables the OPC and PON events of the Standard Event Status Event register.*

Query Syntax ***ESE?**

Returned Parameters **<NR1>** Value: *0* to *255*

Related Commands ***PSC** ***STB?**

*ESR?

Standard Event Status Register Query

Type Device Status

Description This query reads the Standard Event Status Event register. Reading the register clears it. See *Chapter 5 - Status Reporting* for a detailed explanation of this register.

Standard Event Status Event Register

Bit Position	7	6	5	4	3	2	1	0
Bit Name	PON	0	CME	EXE	DDE	QYE	0	OPC
Bit Weight	128	64	32	16	8	4	2	1

Query Syntax ***ESR?**

Returned Parameters **<NR1>** Value: *0* to *255*

Suffix None

Related Commands ***OPC** ***CLS**

*IDN? *Identification* Query

Type System Interface

Description This query requests the electronic load to identify itself.

Query Syntax ***IDN?**

Returned Parameters **<aard>** form consisting of four strings separated by commas. The content of each string is:

String	**Information**
Agilent Technologies	Manufacturer
xxxxA	Four-digit model number followed by a letter suffix
0	Always returns zero
a.xx.xx	Revision level of primary interface firmware

Example `Agilent Technologies,6060A,0,A.01.02` *This identifies an Agilent Model 6060A Electronic Load; with primary interface firmware revision A.01.02*

Related Commands ***OPT** ***RDT?**

*OPC *Operation Complete* Event Bit Command

Type Device Status

Description This command causes Bit 0 of the Standard Event Status Event register to be set when the electronic load has completed all pending operations. (See ***ESR?** for the bit configuration of this register.) *Pending operations* are complete when:

- All previous commands have been executed.
- Any change in the input level caused by previous commands has been completed. (Effects of slew rate have been accounted for.)
- No pending trigger level operations are set for the single electronic load or for any channel of the multiple electronic load.

***OPC** does not prevent processing of subsequent commands but Bit 0 will not be set until all pending operations are complete.

Command Syntax ***OPC**

Parameters None

Related Commands ***WAI** ***OPC?**

*OPC? *Operation Complete* Output Query

Type Device Status

Description This query causes the electronic load to place an ASCII "1" in the Output Queue when all pending operations are completed. *Pending operations* are complete when:

- All commands that were issued before an ***OPC** command have been executed.
- Any change in the input level caused by these previous commands has been completed. (Effects of slew rate have been accounted for.)
- No pending trigger level operations are set for the single electronic load or for any channel of the multiple electronic load.

Unlike ***OPC, *OPC?** prevents processing of all subsequent commands. When all pending operations are completed, an ASCII "1" is placed in the Output Queue.
***OPC?** is intended to be used at the end of a command line so that the program can then monitor the bus for data until it receives the "1" from the Output Queue.

CAUTION

Do not follow a :LEV:TRIG command (**CURR:TRIG, VOLT:TRIG** or **RES:TRIG**) with ***OPC?** unless **TRIG:SOUR** has been previously set to **EXT**ernal, **LINE** or **TIM**er. These are the only triggers that can be processed after ***OPC? TRIG:IMM, *TRG**, and GPIB bus triggers sent after ***OPC?** will be prevented from executing, stopping system operation. If this occurs, the only programmable way to restore operation is by sending the electronic load a GPIB **DCL** (Device Clear) command.

Command Syntax ***OPC?**

Returned Parameters **<NR1>** ASCII *1* is placed in the Output Queue when the electronic load has completed all pending operations.

Related Commands ***OPC** ***TRIG:SOUR** ***WAI**

*OPT? *Options* Identification Query

Type Device Status

Description This query specifies options installed in the multiple electronic load. The query presently is not supported and returns a zero for all electronic loads.

Query Syntax ***OPT?**

Returned Parameters *0*

Suffix None

Related Commands ***IDN?** ***RDT?**

*PSC

Power-on Status Clear Command/Query

Type Device Initialization

Description This command controls the automatic clearing at power turn-on of:

- The Service Request Enable register.
- The Standard Event Status Enable register.

If the command parameter = *0*, then the electronic load can be programmed to request service at turn on. Any non-zero parameter causes both registers to be cleared at turn on, preventing the electronic load from being capable of requesting service at this time. See *Chapter 5 - Status Reporting* for details of these registers.

Command Syntax ***PSC <NRf>**

Parameters 0 or not zero

Suffix None

Query Syntax ***PSC?**

Returned Parameters **<NR1>** *0* = power-on clear flag is *false*; affected registers not cleared at turn on.
1 = power-on clear flag is *true*; affected registers cleared at turn on.

Suffix None

Related Commands None

*RCL

Recall Instrument State Command

Type Device State

Description This command restores the electronic load to a state that was previously stored in memory with a ***SAV** command to the specified location (see ***SAV**). ***RCL** also does the following:

- Forces an **ABORt** command before resetting any parameters. (This removes all pending trigger levels.)
- After all parameters have been recalled, executes an **INP:PROT:CLE** to clear the electronic load's protection circuits.
- Sets **CAL:MODE** to *OFF* (See the electronic load *Operating Manual* for the calibration commands).
- Sets **CHAN** to *1* in the multiple electronic load.

At power turn-on, the equivalent of an ***RCL 0** is executed to restore the electronic load to the state stored in location 0. The same state is also set if the ***RCL** command is directed to a location where no state was stored since the last time power was cycled.

Note ***RCL** does not affect any Status Enable registers or Transition Filters.

Command Syntax	***RCL <NRf>**
Parameters	0 through 6 where: States 1-6 Volatile states previously stored by ***SAV** State 0 Nonvolatile state previously stored by ***SAV 0**
Suffix	None
Related Commands	***RST** ***SAV**

*RDT

Resource Description Transfer Query

Type Device Specification

Description This query returns the model number of a single electronic load or the model number of the module installed in each channel of a multiple electronic load. For multiple electronic loads, a semicolon (;) separates each module and the string terminated with a LF (line feed).

Query Syntax ***RDT?**

Returned Parameters <**aard**> String value as follows:

single electronic load	*CHAN1:nnnnL*; where “nnnnL” = model number
multiple electronic load	*CHAN <c>:nnnnnL*; where “c” = channel number and “nnnnnL” = number and suffix letter of the module in that channel.

Related Commands ***IDN?** ***OPT?**

*RST

Reset Command

Type Device State

Description This command sets the electronic load to its factory-defined state. (Refer to “Factory Default Settings” in the Operating Manual of the electronic load model that you are programming.) There are no parameters with this command; it sets all channels of the multiple electronic load to the same state.

*RST also does the following:

- Forces an **ABORt** command before resetting any parameters.
- After all parameters have been reset, executes an **INP:PROT:CLE** to clear the electronic load’s protection circuits.

Note ***RST** does not affect any Status Enable registers or Transition Filters.

Command Syntax ***RST**

Parameters None

Related Commands ***RCL** ***SAV**

*SAV

Save Command

Type Device State

Description This command stores the present state of the single electronic load and the states of all channels of the multiple electronic load in a specified location in memory. Location 0 is in nonvolatile memory and retains its state throughout power cycling. The electronic load will be set to the state in location 0 at power turn-on. If no state has been saved to location 0, then it will still contain the factory-default state (refer to "Factory Default Settings" in the Operating Manual of the electronic load model that you are programming). States stored in locations 1 through 6 are lost whenever power is cycled.

Note To restore the factory-default state to Location 0, execute ***RST;*SAV 0**

The parameters stored by ***SAV** are identical to those affected by ***RST** *except* that the following states are *not* stored:

- **CAL:MODE ON|OFF** (Refer to the electronic load; *Operating Manual*).
- **CHAN.**

Note ***SAV** also does not store the states of Status Enable registers or Transition Filters.

Command Syntax ***SAV <NRf>**

Parameters *0* to *6*

Suffix None

Example ***SAV 2** *Save the present state of the electronic load to location 2*

Related Commands ***RCL** ***RST**

*SRE

Service Request Enable Command/Query

Type Device Interface

Description This command sets the condition of the Service Request Enable register, which determines which events of the Status Byte register (see ***STB**) are allowed to set the MSS (Master Status Summary) bit. A "1" in the bit position enables the corresponding Status Byte bit to set the MSS bit. All the enabled bits are logically ORed to cause Bit 6 (the Master Summary Status Bit) of the Status Byte register to be set. See *Chapter 5 - Status Reporting* for more details concerning the Status Byte register.

Command Syntax ***SRE <NRf>**

Parameters *0* to *255*

Suffix None

Example ***SRE 20** *Enables either the CSUM or MAV condition to cause a service request.*

Query Syntax ***SRE?**

Returned Parameters **<NR1>** Value: *0* to *255*

Suffix None

Related Commands ***PSC**

---

*STB?

Read *Status Byte* Query

Type Device Status

Description This query reads the Status Byte register. Note that the MSS (Master Summary Status) bit and not the RQS bit is returned in Bit 6. This bit indicates whether or not the electronic load has at least one reason for requesting service. ***STB?** does not clear the Status Byte register, which is cleared only when subsequent action has cleared all its set bits. Refer to *Chapter 5 - Status Reporting* for more information about this register.

Status Byte Register

Bit Position	7	6	5	4	3	2	1	0
Condition	OPER	MSS	ESB	MAV	QUES	CSUM	[1]0	[1]0
Bit Weight	128	64	32	16	8	4	2	1
[1]Always zero.								

Parameters None

Returned Parameters **<NR1>** *Value: 0* to *255*

Suffix None

Related Commands None

---

*TRG

Immediate *Trigger* Command

Type Device Trigger

Description This command which is essentially the same as the Group Execute Trigger **(<GET>),** generates a trigger to the electronic load only if **TRIG:SOUR** is set to **BUS** (refer to the TRIGger Subsystem Root commands).

Command Syntax ***TRG**

Parameters None

Related Commands **<GET>** **TRIG:SOUR**

*TST? *Self Test* Query

Type Device Test

Description This query causes the electronic load to go through a limited self-test (a more complete one is done at power turn on). The testing does not alter the mode or parameter settings of the electronic load.

Query Syntax ***TST?**

Returned Parameters **<NR1>** 0 = test passed

Nonzero indicates a self-test failure. For single electronic loads, the returned value is of concern only to service personnel. For multiple electronic loads, the returned values indicate failures in the following modules:

.

Multiple Electronic Load Failure Bit Map

Bit Position	6	5	4	3	2	1	0
Failed Channel	6	5	4	3	2	1	Mainframe
Bit Weight	64	32	16	8	4	2	1

Suffix None

Related Commands None

*WAI *Wait to Continue* Command

Type Device Status

Description This command instructs the electronic load not to process any further commands until all pending operations are completed. "Pending operations" are as defined under the ***OPC** command. ***WAI** can be aborted only by sending the electronic load a GPIB **DCL** (Device Clear) command.

Command Syntax ***WAI**

Parameters None

Related Commands ***OPC** ***OPC?**

Root-Level Commands

Introduction

Root-level commands are those that are specific to the family of electronic loads. The commands are grouped as either channel-specific or channel-independent commands. In the Multiple Electronic Load, channel-specific commands are directed (via the **CHAN**nel command) to specific modules in the mainframe.

Tree Diagram

Figure 4-2 is a tree diagram of the root-level commands. Commands starting at the root directory are listed as either single commands or command subsystems. Command subsystems may consist of a single command, but usually are comprised of a set of commands that extend two or more levels below the root. Refer to *Chapter 3-Introduction to Programming* for rules for traversing the command tree and for examples.

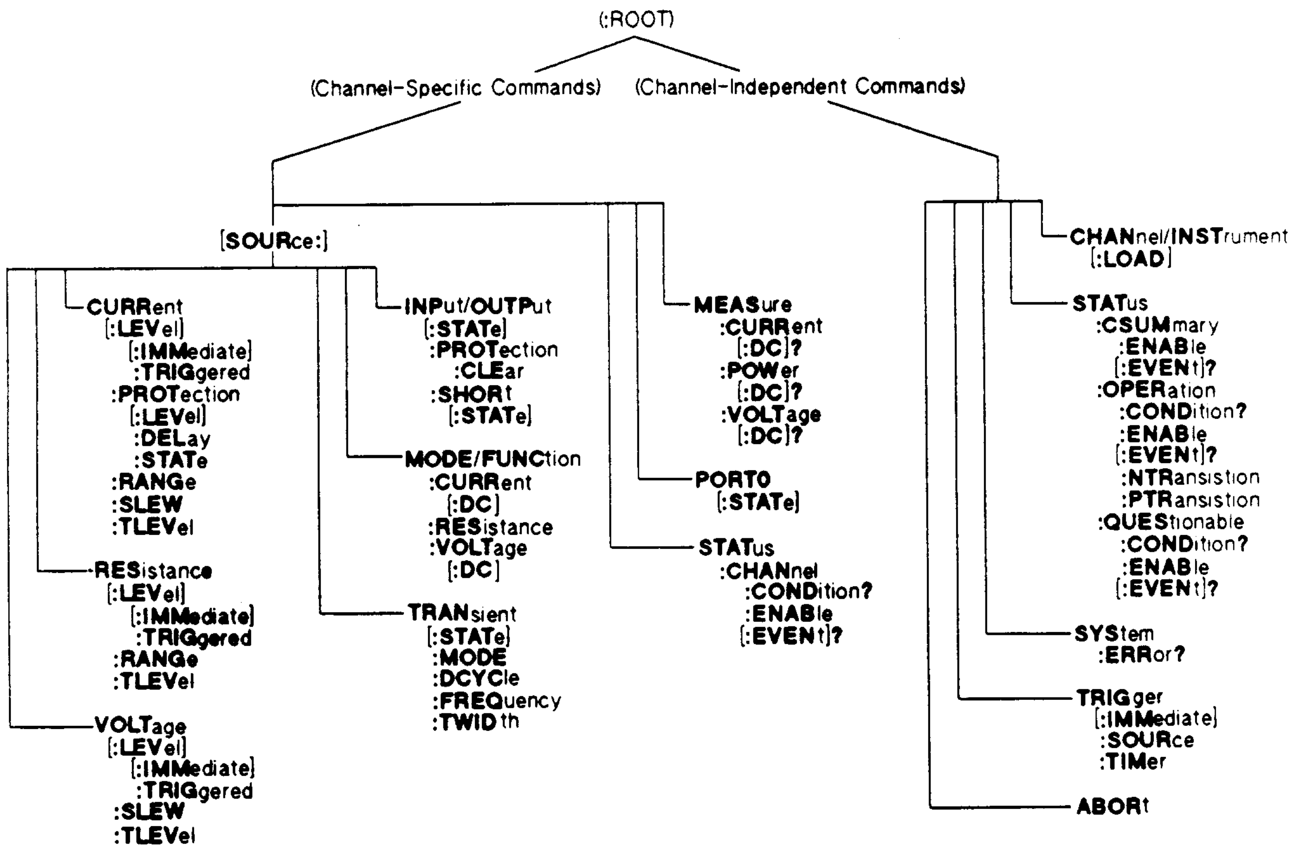

(:ROOT)
(Channel-Specific Commands)
(Channel-Independent Commands)
[SOURce:]
CURRent [:LEVel] [:IMMediate] :TRIGgered :PROTection [:LEVel] :DELay :STATe :RANGe :SLEW :TLEVel
RESistance [:LEVel] [:IMMediate] :TRIGgered :RANGe :TLEVel
VOLTage [:LEVel] [:IMMediate] :TRIGgered :SLEW :TLEVel
INPut/OUTPut [:STATe] :PROTection :CLEar :SHORt [:STATe]
MODE/FUNCtion :CURRent [:DC] :RESistance :VOLTage [:DC]
TRANsient [:STATe] :MODE :DCYCle :FREQuency :TWIDth
MEASure :CURRent [:DC]? :POWer [:DC]? :VOLTage [:DC]?
PORT0 [:STATe]
STATus :CHANnel :CONDition? :ENABle [:EVENt]?
CHANnel/INSTrument [:LOAD]
STATus :CSUMmary :ENABle [:EVENt]? :OPERation :CONDition? :ENABle [:EVENt]? :NTRansistion :PTRansistion :QUEStionable :CONDition? :ENABle [:EVENt]?
SYStem :ERRor?
TRIGger [:IMMediate] :SOURce :TIMer
ABORt

Figure 4-2. Electronic Loads Tree Diagram

ABORt

Channel- Independent Termination Command

Description

This command applies only to trigger functions. It cancels all pending [:**LEV**el]:**TRIG** operations (such as **CURR:TRIG**) in all operating modes and on all channels. As a result, subsequent triggers have no effect on the input level. This command resets the WTG bit of the Operation Condition register (refer to *Chapter 5 - Status Reporting*) and has the same effect on status as the receipt of a trigger. **ABOR**t has no affect on the Transient Subsystem.

Command Syntax

ABORt

Parameters

None

Examples

ABOR	*Aborts all pending trigger level operations in the Current, Resistance, or Voltage Subsystems*
CURR 10; CURR: TRIG 20;: ABOR;: TRIG	*Cancels the programmed trigger level so that the triggered input remains at 10 amps*

Query

None

Related Commands

CURR[:LEV]:TRIG **VOLT[:LEV]:TRIG** **STAT:OPER:COND?**

CHANnel

Channel-Independent Command/Query

Note

This command is for the multiple electronic load but is supported in single electronic loads for compatibility.

Description

CHANnel selects the multiple electronic load channel to which all subsequent channel-specific commands will be directed. If the specified channel number does not exist or is outside the MIN/MAX range, an error code is generated (See Table 4-2 at the end of this chapter). When used with the single electronic load the only valid parameter is 1.

Note

This command and **INST**rument are the *only* channel commands for electronic loads. No other methods of channel selection are supported.

Command Syntax

CHANnel[:**LOAD**] <**NRf+**>

Parameters

See Table 4-1 and the Operating Manual of the electronic load model.

Examples

CHANNEL:LOAD 5 **CHAN 5**	*Select Channel 5*
CHAN MAX	*Select highest existing channel in a multiple electronic load.*

Query Syntax

CHAN? **CHAN? MAX** **CHAN? MIN**

Returned Parameters **< NRl > CHAN?** returns number of channel presently selected. **CHAN? MAX** returns the number of channels installed in the multiple electronic load. If none are installed, *0* is returned. For single electronic loads **CHAN? MAX** returns *1*. **CHAN? MIN** always returns *1* for either single or multiple loads.

Suffix None

Related Commands None

Alternate Syntax **INST**rument can be used as an alias for **CHAN**nel.

CURRent Subsystem *Channel-Specific* Current Programming Function

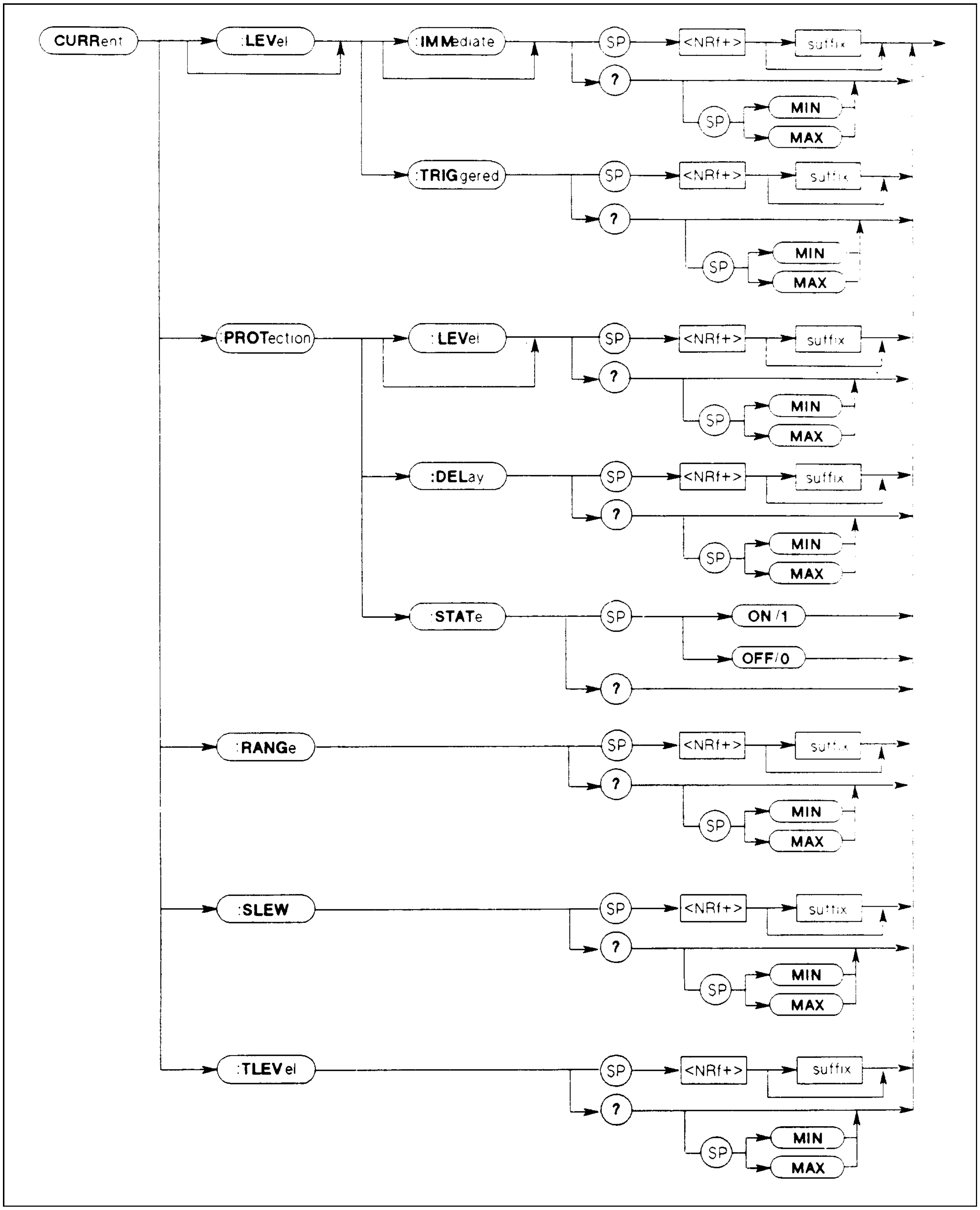
Description This subsystem programs the CC (constant-current mode) function of a single electronic load or a single channel of a multiple electronic load.

Keywords

Command	*Function*
CURRent[:**LEV**el][:**IMM**ediate]	Specify the input current level for the **CURR**ent mode.
CURRent[:**LEV**el] :**TRIG**gered	Preset a new input current level to be valid when a trigger occurs.
CURRent:**PROT**ection[:**LEV**el]	Set current limit at which protection occurs.
CURRent:**PROT**ection:**DEL**ay	Set time for which protection current limit may be exceeded.
CURRent:**PROT**ection:**STAT**e **OFF|0**	Disable protection function.
CURRent:**PROT**ection:**STAT**e **ON|1**	Enable protection function.
CURRent:**RANG**e	Specify full-scale input current range.
CURRent:**SLEW**	Specify the current level rate of change for all current ranges and for the middle and maximum resistance ranges.
CURRent:**TLEV**el	Specify input current level used with the **TRAN**sient Subsystem.

Related Subsystems **RES**istance **VOLT**age

Syntax Diagram



CURR[:LEVel] *Channel-Specific* Current Command/Query

Description This is an implied keyword that specifies the value of the programmed current level and whether that level is to be applied immediately or on occurrence of a trigger. If the specified channel is in the CC (Constant-Current) Mode, an **IMM**ediate current level is transferred to the input as soon as the command is executed. A **TRIG**gered level is stored and then transferred to the electronic load input when a trigger occurs. At that time, the change to the input level occurs at the slew rate presently in effect. Following the trigger event, subsequent triggers will not affect the input level unless the electronic load has been sent another **TRIG**gered level command.

If the electronic load is not in the CC (Constant-Current) Mode when an **IMM**ediate or **TRIG**gered level command is sent, the programmed levels are saved for the time the electronic load is placed in the CC mode. Triggered levels are processed by the Current Subsystem even when the electronic load is not in the CC Mode. In this case, the **TRIG**gered level becomes a stored **IMM**ediate level that takes effect when the electronic load is again in the CC Mode.

Note Setting an **IMM** current level to the same value as the most recent **TRIG** current level will not deactivate a pending **TRIG** level. You must use **ABOR**t to deactivate it.

The present current level changes to the pending level on any of the following conditions:

- On a **TRIG[:IMM]** command (always)
- On receipt of an external trigger signal (if **TRIG:SOUR** is set to **EXT**)
- On the next line voltage cycle (if **TRIG:SOUR** is set to **LINE**)
- On receipt of ***TRG** (unless **TRIG:SOUR** is set to **HOLD**)
- On receipt of a GPIB <GET> (if **TRIG:SOUR** is set to **BUS**)
- On the next trigger timer pulse (if multiple electronic load is set to **TRIG:SOUR TIM**)

The programmed current level (whether **IMM**ediate or **TRIG**gered can be implicitly changed with a **RANG**e command (See *Chapter 3 - Introduction to Programming* for information concerning value coupling).

Command Syntax **CURR**ent[:**LEV**el][:**IMM**ediate] **<NRf+>**
CURRent[:**LEV**el]:**TRIG**gered **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Status and Errors **TRIG**gered level commands affect the WTG bit in the Operation Condition register and the **OPC** bit of the Standard Event Status Event register (See *Chapter 5 - Status Reporting).* Programmed current levels outside the value range generate an error (See Table 4-2 at the end of this chapter). The correct current range must be programmed before the current level is programmed.

Value Coupling If **CURR:RANG**e is set to a range that is *below* either type of **LEV**el, then that **LEV**el will assume the *MAX* value of that range.

Examples

CURR 25 **CURRENT:LEVEL 25**	*Immediate commands for 25-ampere input*
CURR:TRIG 25MA **CURRENT:LEVEL:TRIGGERED 25E-3**	*Commands for 25 mA input on occurrence of a trigger*
CURR 30; :CURR:TRIG MIN	*Set input to 30* Amps *now and minimum current when trigger occurs*

Query Syntax **CURR?** **CURR? MAX** **CURR? MIN**
CURR:TRIG? **CURR:TRIG? MAX** **CURR:TRIG? MIN**

Returned Parameters **<NR3>** **CURR?** and **CURR:TRIG?** return the presently programmed current levels. After a trigger or **ABOR**t, **CURR:TRIG?** returns the same value as **CURR?** .

CURR? MAX, CURR? MIN, CURR:TRIG? MAX and **CURR:TRIG? MIN** return the maximum and minimum programmable **LEV**el and **TLEV**el values for the present range.

Related Commands **ABOR** **CURR:RANG** see also **TRAN**sient Subsystem

CURR:PROTection

Channel-Specific Current Limiting Command/Query

Description This command sets a limit to the input current that the electronic load will sink. A current limit may be specified for the single electronic load or for a channel of the multiple electronic load. When the input current reaches the current limit for the specified delay period, the input of the electronic load or channel is shut off and draws no current. This, in effect, provides a "soft circuit breaker" for the input current. The **INP**ut:**PROT**ection:**CLE**ar command (or front panel key) re-enables the input current. The trigger activated current functions (**CURR[:LEV]:TRIG** and **CURR:TLEV**) automatically keep track of incoming triggers while the input is shut down and will respond to the trigger as soon as the protection fault is cleared.

The **:PROT**ection:**DEL**ay command specifies the time that the input current may equal or exceed [**:LEV**el] before the soft circuit breaker is actuated. The **PROT**ection:**STAT**e command enables or disables the soft circuit breaker function.

Note If the soft circuit breaker function causes the input to shut down, it will not affect **INP[STAT**e]. If **INP:STAT** is programmed **ON**, it will remain so even after the **CURR:PROT** has turned the electronic load off.

Command Syntax

Command	*Function*
CURRent:**PROT**ection[:**LEV**el] **<NRf+>**	Set immediate current limit.
CURRent:**PROT**ection:**DEL**ay **<NRf+>**	Set time that current may be at or above :**LEV**el before input is turned off.
CURRent:**PROT**ection:**STAT**e **OFF|0**	Disable protection function.
CURRent:**PROT**ection:**STAT**e **ON|1**	Enable protection function.

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`CURR: PROT: LEVEL 35`	*Set input current limit to 35 amperes and*
`CURR:PROT:STAT ON`	*enable the current protection*
`CURR: PROT: LEVEL 35; DELAY . 025`	*Set input current limit to 35 amperes and allow it to be exceeded for 25 milliseconds*

Query Syntax
CURR:PROT?
CURR:PROT? MIN **CURR:PROT? MAX**
CURR:PROT:DEL?
CURR:PROT:DEL? MIN **CURR:PROT:DEL? MAX**
CURR:PROT:STAT?

Returned Parameters

<NR3>	**CURR:PROT?** returns the value of existing current limit. **CURR:PROT? MAX** and **CURR:PROT? MIN** return the maximum and minimum programmable values for the current limit.
<NR3>	**CURR:PROT:DEL?** returns the value of existing delay. **CURR:PROT:DEL? MAX** and **CURR:PROT:DEL? MIN** return the maximum and minimum programmable delay values.
<NR1>	**CURR:PROT:STAT?** returns the state of the protection circuit where **1** is *ON* and **0** is *OFF*.

Related Commands **CURR:RANG** **INP:PROT:CLE**

CURR:RANGe

Channel-Specific Current Command/Query

Description This command selects the full-scale current range of the electronic load. Programming any value within the low range automatically selects the low range and programming any value within the high range automatically selects that range. In the Agilent 6060A for example, programming a value from 6 to 60 amperes automatically selects the 60-ampere range.

Important Whenever the electronic load changes range, it momentarily goes into the OFF (minimum current) state.

When **CURR:RANG** is executed, the values of the current levels (**IMM**ediate, **TRIG**gered and **TLEV**el) are adjusted as follows:

If	*Then*
The existing current setting is within the new range.	The current level does not change.
The existing current setting is not within the new range.	The current level is set to the maximum of the new range.

For example, assume the electronic load is in the 60 A range and the main current level (**CURR:LEV**) is 30 A. Switching to the 6 A range will reduce the current to the maximum of that range, or 6 A. However, if another parameter (such as **CURR:TRIG**) was already within the new range (e.g., 4 A), then it will remain at that level in the new range. Of course, there is no change in value when switching from the lower range to the higher range. However, the accuracy of the programmed value may be compromised because of the reduced resolution of the higher range.

Command Syntax **CURR**ent:**RANG**e **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model

Value Coupling This command affects the following parameters:
CURR:LEVel **CURR:TLEV**el

Examples

CURR: RANG 60; LEV 25 . 25	*Set current to 25.25 A on 60-ampere range*
CURR:RANG 6; :TRIG 4.5	*In previous example, change range to 6 A and set Triggered level to 4.5 A. Due to coupling, the main level will drop to 6 A and remain there until a trigger occurs.*

Query Syntax **CURR:RANG?**
CURR:RANG? MAX
CURR:RANG? MIN

Returned Parameters **<NR3>** **CURR:RANG?** returns the present current range in amperes.
CURR:RANG? MAX and **CURR:RANG? MIN** return the maximum and minimum programmable ranges for the electronic load.

Related Commands **CURR:[LEV]** **CURR:TRIG** **CURR:TLEV**

CURR:SLEW

Channel-Specific Current Command/Query

Description This command sets the current programming slew rate for both CC mode ranges and the resistance programming slew rate for the middle and high CR mode ranges. The programmed slew rate is used for all programmed current changes except **INP**ut **ON** or **OFF**. The hardware implements discrete slew rates (refer to the electronic load *Operating Manual)* and automatically selects the one that is closest to the programmed value. To determine the actual rate, use the query (**CURR:SLEW?**).

Command Syntax **CURR**ent:**SLEW <NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

CURRENT:SLEW MAX	*Set slew rate for maximum of present current range*
CURR: RANGE MAX; SLEW 4E5	*Set range to 60 A and slew rate to 400,000 A/s (400 mA/µs)*

Note Programming a slew rate value greater than **MAX** sets the slew rate to maximum without generating an error message.

Query Syntax **CURR:SLEW?**
CURR:SLEW? MAX
CURR:SLEW? MIN

Note This query is not applicable to CR mode when it is operating in the low-resistance range.

Returned Parameters **<NR3>** **CURR:SLEW?** returns the internally selected slew rate (in A/S) that was chosen as closest to the programmed value within the permissible range. **CURR:SLEW? MAX** and **CURR:SLEW? MIN** return the maximum and minimum programmable slew rates for the present range.

Related Commands None

CURR:TLEVel

Channel-Specific Current Command/Query

Description This command specifies the value of the programmed current level for the **TRAN**sient input when the electronic load is in the CC Mode. When the Transient Subsystem is on, the electronic load input current will switch (under control of the Transient Subsystem) between the main level and **TLEV**el at a rate determined by the present value of **SLEW**. In order for the input current level to switch, **TLEV**el must be set to a value greater than the main level. If **TLEV**el is set to a value below the main level, no error is generated but switching will not occur until the main level is subsequently below the value of **TLEV**el. If **TLEV**el is programmed outside the specified range, an error is generated (See Table 4-2 at the end of this chapter).

Command Syntax **CURR**ent:**TLEV**el **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Value Coupling If **CURR:RANG**e is set to a range that is below **TLEV**el, then **TLEV**el will assume the ***MAX*** value of that range.

Examples

`CURR:TLEV MAX`	*Set transient level to maximum current for the range*
`CURR: RANG 6; TLEV 5 . 5`	*Set range to 6 A and Transient level to 5. 5 A*

Query Syntax **CURR:TLEV?**
CURR:TLEV? MAX
CURR:TLEV? MIN

Returned Parameters **<NR3>** **CURR:TLEV?** returns the transient current level for present current range.
If the electronic load is not in CC Mode, the level will still be set, even if it is less than the presently programmed input level. **CURR:TLEV? MAX** and **CURR TLEV? MIN** return the maximum and minimum programmable values for the present range.

Related Commands **CURR:RANG** also see **TRAN**sient Subsystem

INPut Subsystem

Channel-Specific Input Programming Functions

Description

This subsystem has commands for:

- Turning the electronic load input on or off.
- Placing a short across the electronic load input.
- Clearing any input software protection circuits that have been set.

For multiple electronic loads these commands function only on the presently specified channel.

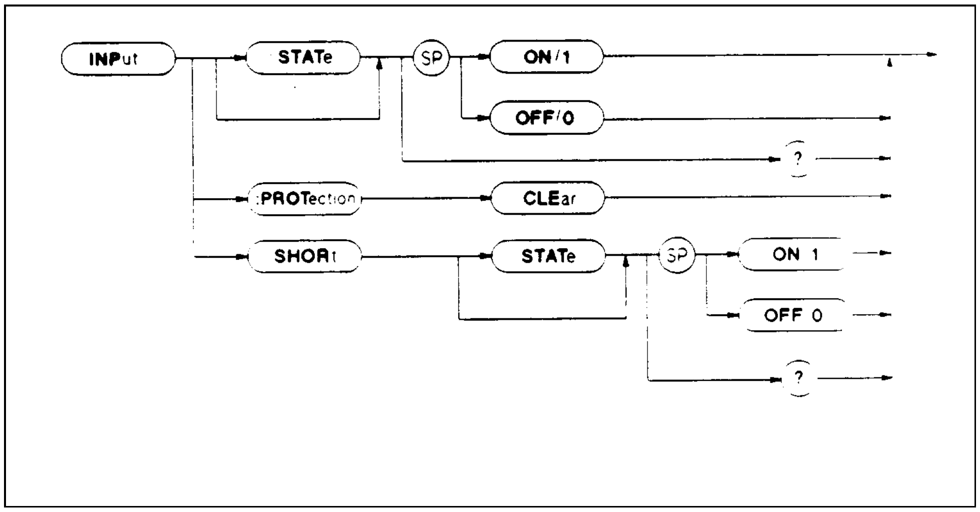
Keywords

Command	*Function*
INPut[:**STAT**e] **ON|l**	Turn electronic load input circuit ON.
INPut[:**STAT**e] **OFF|0**	Turn electronic load input circuit OFF.
INPut:**PROT**ection:**CLE**ar	Reset input protection circuits.
INPut:**SHOR**t[:**STAT**e]:**ON|l**	Place short across electronic load input.
INPut:**SHOR**t[:**STAT**e]:**OFF|0**	Remove short from electronic load input.

Alternate Syntax **OUTP**ut can be used as an alias for **INP**ut.

Related Subsystems **CURR**ent **RES**istance **VOLT**age

Syntax Diagram



Input Subsystem Syntax Diagram

INP:PROT:CLEar

Channel-Specific Input Command

Description This command resets the electronic load latched protective features (such as overvoltage, overcurrent, overpower, etc.). If an electronic load protective circuit has shut down the input and no **INP**ut:**OFF** command has been sent, **INP**ut:**CLE**ar will restore the input to the **ON** state and restore the previously existing operating mode and its parameters. Of course, if the cause of a shutdown condition is not corrected, it will reoccur.

If a trigger is received during the time a protection circuit has shut down the input, any triggered levels (**CURR:TRIG, RES:TRIG**, etc.) are stored and will be transferred to the main level after the protection is cleared. Triggers received during protective shutdown are also applied to the Transient Subsystem. This ensures that after the shutdown is cleared, the transients on the restored channel will still be synchronized with those of other channels.

Command Syntax **INP**ut:**PROT**ection:**CLE**ar

Parameters None

Query Syntax None

Related Commands **CURR:PROT[:LEV]** **CURR:PROT:DEL** **CURR:PROT:STAT**

INP:SHORt

Channel-Specific Input Command/Query

Description This command applies the equivalent of an electronic short across the input of the electronic load. The actual condition of the electronic load under a short condition depends on its operating mode as follows:

Mode of Operation	**Shorted Condition**
High-Range CC Mode	MAX current
Low-Range CC Mode	MAX current
Low-Range CR Mode	MIN resistance
Middle-Range CR Mode	MIN resistance
High-Range CR Mode	MIN resistance
CV Mode	MIN voltage

If a mode or current or resistance range is changed, the short will be reapplied to the new mode or range. If a trigger is received while **SHOR**t is **ON**, any triggered levels (**CURR TRIG, RES:TRIG,** etc.) are stored and will be transferred to the main level after **SHOR**t is programmed **OFF**. Triggers received during **SHORt:ON** are also applied to the Transient Subsystem. This ensures that after the short is programmed off, the transients on the shorted channel will still be synchronized with those of channels that had remained unshorted.

Executing **INP**ut:**SHOR**t does not affect any programmed settings and the electronic load will return to them when the short is removed. This command is subordinate to **INP**ut:[**STAT**e]. If **INP**ut[:**STAT**e] is programmed **OFF**, then the effect of **INP**ut:**SHOR**t:**ON** will not be observed until **INP**ut[:**STAT**e]:**ON** is sent to the electronic load.

Command Syntax **INPu**t:**SHOR**t[:**STAT**e] **<NRf+>**

Parameters

Value Range	Units	*RST Default
OFF or *0*	None	*0*
ON or *1*	None	

Examples

`INP:SHOR ON`	*Set shorted input condition*
`INP:SHOR 1`	

`CHAN 2;INP:SHOR 1;:CHAN 1;INP:SHOR 0`
On multiple electronic load, short Channel 1 and unshort Channel 2

Query Syntax **INP:SHOR?**

Returned parameters **<NR1>** Value; **0** for *unshorted*, **1** for s*horted*

Related Commands **INP[:STAT]**

INP[:STATe]

Channel-Specific Input Command/Query

Description This implied keyword turns the electronic load input ON and OFF. When **INP**ut is **OFF**, the electronic load will draw minimal current at its input. **INP[STAT]**e:**OFF** overrides **INP**put:**SHOR**t:**ON**. The presently programmed slew rate setting is not used when turning the input on or off; the current and voltage change at their maximum rates .

Command Syntax **INPu**t [:**STAT**e] **<NRf+>**

Parameters

Value Range	Units	*RST Default
OFF or *0*	None	*0*
ON or *1*	None	

Examples

`INP:STAT OFF`	*Turn electronic load input off*
`INP 0`	

`INP:STAT OFF;SHOR ON;:INP ON` *Turn input off, short the input, and turn input on.*

Query Syntax **INP?**

Returned Parameters **<NR1>** Value; **0** for *OFF,* **1** for *ON*

Related Commands **CURR:PROT** **INP:SHOR**

MEASure

Channel-Specific Measurement Query

Description This function consists of queries that return the current, voltage, and power at the input of the electronic load.

Query Syntax

Query	*Value Returned*
MEASure:**CURR**ent[:**DC**]?	electronic load input current
MEASure:**POW**er[:**DC**]?	Computed electronic load input power
MEASure:**VOLT**age[:**DC**]?	electronic load input voltage

Returned parameters **<NR3>** Value representing amperes, watts, or volts

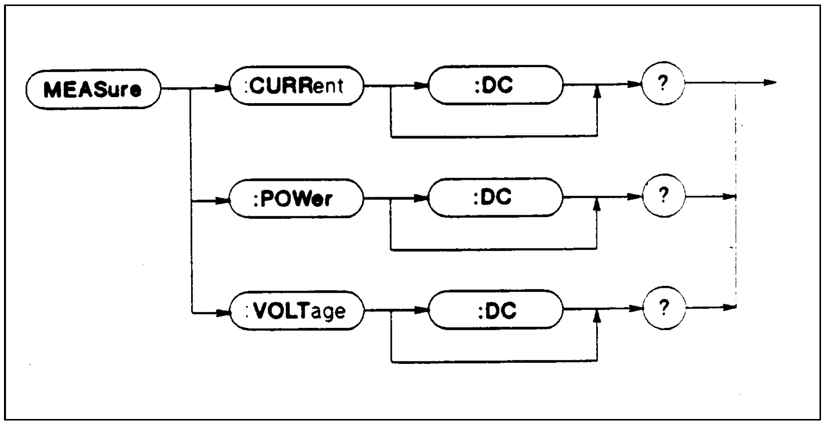
Note If the input voltage or current exceeds the maximum measurement capability of the electronic load, an 9.9E+37 out-of-range indication will be returned in place of the normal measurement reading.

Examples

MEAS:VOLT?	*Return input voltage*
MEAS: CURR?; VOLT?; POW?	*Return input current, voltage and power*
CHAN 1; :MEAS CURR?;:CHAN 2;:MEAS POW?	*Return Channel 1 current & Channel 2 power*

Related Commands None

Syntax Diagram



Measure Query Syntax Diagram

MODE

Channel-Specific Command/Query

Description This command selects the operating mode of the electronic load, which can be:

CURRent	Constant-current (CC) input
RESistance	Constant-resistance (CR) input
VOLTage	Constant-voltage (CV) input

If the mode is changed while the input is on, the input is momentarily turned off as if **INP**ut(:**STAT**e] **OFF** were executed. However, changing modes does not change the state of the input nor does it turn off the TRANsient function. For multiple electronic loads, the mode of each channel is programmed independently.

Command Syntax

Command	*Function*
MODE:CURRent**[:DC]**	Sets electronic load to constant-current mode.
MODE RESistance	Sets electronic load to constant-resistance mode.
MODE:VOLTage**[:DC]**	Sets electronic load to constant-voltage mode.

Parameters Enter the desired mode as a string variable in either the full or abbreviated format. There are no units. The *RST Default is *CURRent.*

Examples

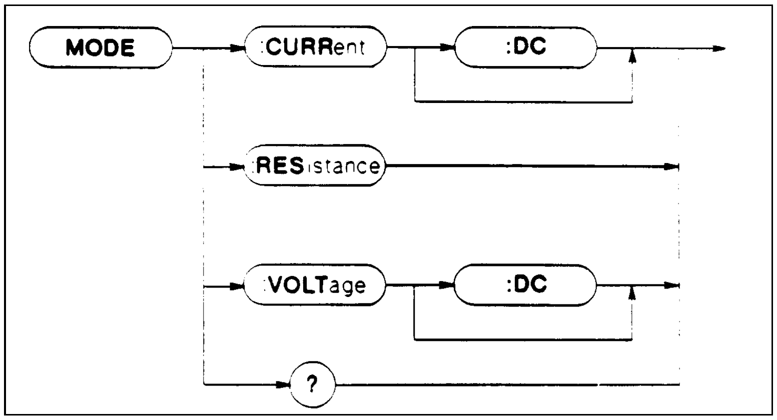
MODE :CURR	*Set electronic load input to CC mode*
CHAN 1;: MODE:CURR; :CHAN 2; :MODE:RES	Set *Channel 1 input to CC mode and Channel 2 input to CR mode*

Query Syntax **MODE?**

Returned Parameters **<aard>** String value: *CURR, RES,* or *VOLT*

Alternate Syntax **FUNC**tion can be used as an alias for **MODE.**

Syntax Diagram



Mode Command Syntax Diagram

PORT0

Channel-Specific Output Command/Query

Description This command controls the general-purpose digital port on the rear of each electronic load or each channel of the multiple electronic load. The state of the port is with respect to the common side of the electronic load input. Setting a port to the **OFF** or **0** state connects it to the common; setting a port to the **ON** or 1 state disconnects it from common and sets it to a TTL logical high.

The command is the same for both single and multiple electronic loads. To control a port in a multiple electronic load you must first select the channel and then use **PORT0** on that channel.

Command Syntax **PORT0[:STATe] <NRf+>**

Note This command does not have a four-character short form. You must always include the fifth character (zero).

Parameters

Value Range	Units	*RST Default
OFF or *0*	None	*0*
ON or *1*	None	

Examples	`PORT0 ON`	*Set the port output high (logical 1)*
	`CHAN 1;: PORT0 ON;:CHAN 2;:PORT0 OFF`	*Set Channel 1 port high and Channel 2 port low.*
Query Syntax	**PORT0?**	
Returned Parameters	**<NR1>**	**1** for *ON* or **0** for *OFF*
Related Commands	None	

RESistance Subsystem

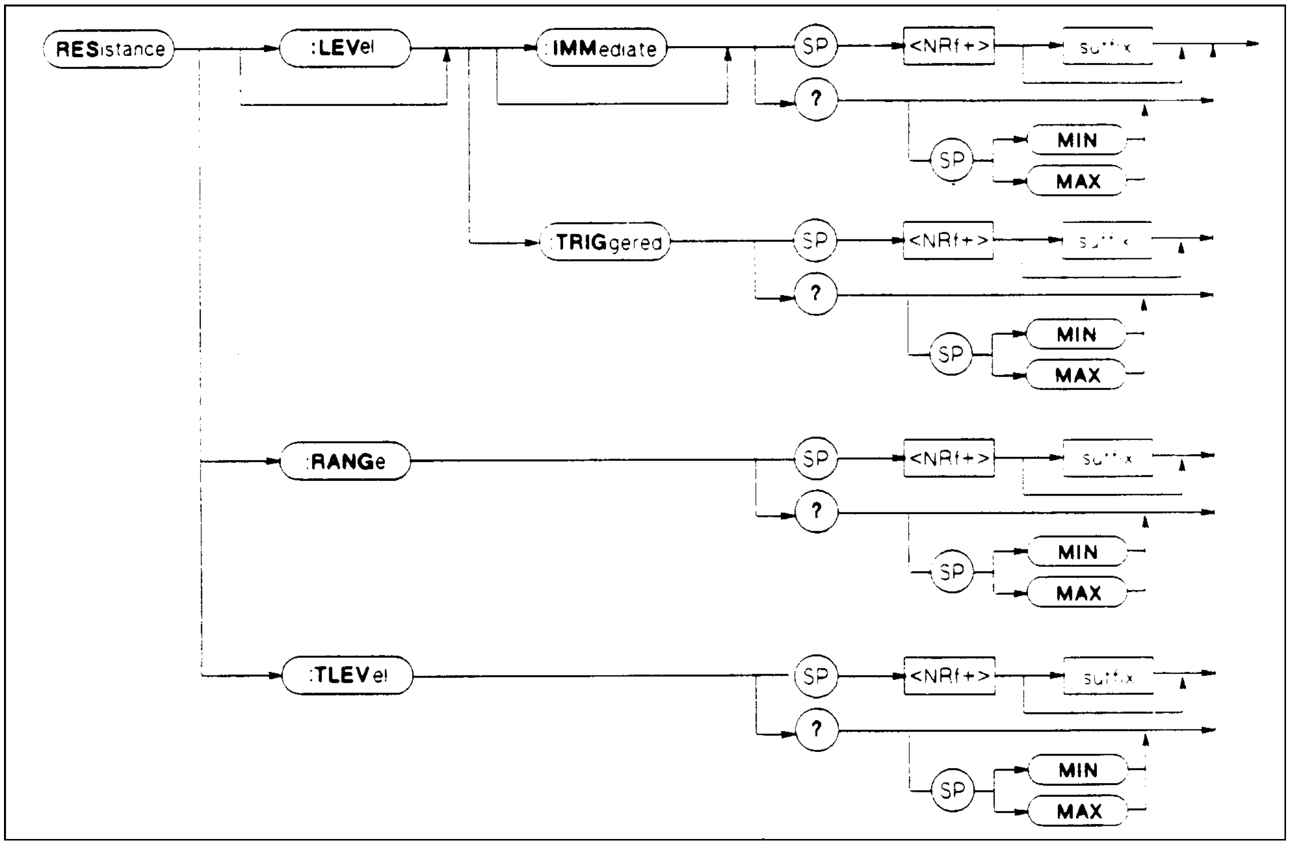
Channel-Specific Resistance Programming Function

Description This subsystem programs the constant-resistance (CR) mode function of a single electronic load or a single channel of a multiple electronic load. There is no **SLEW** command in this subsystem. In the lowest resistance range the level changes in Volts/sec at the rate specified by the present **VOLT:SLEW** command. The middle and high ranges change level in Amps/sec at the rate specified by the present **CURR:SLEW** command.

Keywords

Command	*Function*
RESistance[:**LEV**el][:**IMM**ediate]	Specify input resistance for CR mode.
RESistance[:**LEV**el]:**TRIG**gered	Preset a new input resistance to be valid when a trigger occurs.
RESistance:**RANG**e	Specify full-scale input resistance range.
RESistance:**TLEV**el	Specify input resistance used with **TRAN**sient Subsystem.

Syntax Diagram



Resistance Subsystem Syntax Diagram

RES[:LEVel]

Channel-Specific Resistance Command/Query

Description This is an implied keyword that specifies the value of the programmed input resistance and whether that value is to be applied immediately or on occurrence of a trigger. If the specified channel is in the CR (Constant-Resistance) Mode, an **IMM**ediate resistance level is transferred to the input as soon as the command is executed. A **TRIG**gered level is stored and transferred to the electronic load input when the trigger occurs. At that time, the change to the input level occurs at the presently active voltage slew rate (for the lowest resistance range) or presently active current slew rate (for the middle and highest ranges). Following the trigger event, subsequent triggers will not affect the input level unless the electronic load has been sent another **TRIG**ered level command.

If the electronic load is not in the CR Mode when an **IMM**ediate or **TRIG**gered level command is sent, the programmed levels are saved for the next time the electronic load is placed in the CR Mode. Triggered levels are processed by the Resistance Subsystem even when the electronic load is not in the CR Mode. Thus, the **TRIG**ered level becomes a stored **IMM**ediate level that takes effect when the electronic load is again in the CR mode.

Note Setting an **IMM** resistance level to the same value as the most recent **TRIG** resistance level will not deactivate a pending **TRIG** level. You must use **ABOR**t to deactivate it.

The present resistance level of each channel changes from the present level to the pending level on any of the following:

- On a **TRIG[:IMM]** command (always).
- On receipt of an external trigger signal (if **TRIG:SOUR** is set to **EXT**).
- On the next line voltage cycle (if **TRIG:SOUR** is set to **LINE**).
- On receipt of ***TRG** (unless **TRIG:SOUR** is set to **HOLD**).
- On receipt of a GPIB <GET> (if **TRIG:SOUR** is set to **BUS**).
- On the next trigger timer pulse (if multiple electronic load is set to **TRIG:SOUR TIM**).

The programmed resistance level (whether **IMM**ediate or **TRIG**gered) can be implicitly changed with a **RANG**e command (refer to *Chapter 3 - Introduction to Programming* for information concerning value coupling).

Command Syntax

RESistance**[:LEV**el**][:IMM**ediate**l <NRf+>**
RESistance**[:LEV**el**] :TRIG**gered **< NRf+ >**

Parameters

See Table 4-1 and the Operating Manual of the electronic load model.

Note

The higher values of resistance levels have less resolution. Programmed levels are set to the nearest value obtainable by the hardware.

Status and Errors

TRIGgered level commands affect bits in the **STAT**us **OPER**ating **COND**ition and **STAT**us **CHAN**nel **COND**ition registers (See *Chapter 5 - Status Reporting).* Programmed resistance values outside the value range generate an error (See Table 4-2 at the end of this chapter). The correct resistance range must be programmed before the resistance level is programmed. Note from the parameters that there is considerable overlap between the middle and high ranges, but *no overlap between the low and middle ranges.*

Value Coupling

If **RES:RANG**e is set to a range that is *below* either type of **LEV**el, then that **LEV**el will assume the ***MAX*** value of that range.

Examples

RES 25 **RESISTANCE:LEVEL 25**	*Immediate commands for 25-ohm input*
RES:TRIG: . 025 **RESISTANCE:LEVEL:TRIGGERED 25E-3**	*Commands for 25 milliohm input on occurrence of a trigger*
RES:IMM .1;TRIG 1	*Set input to 100 milliohms now and to 1 ohm when trigger occurs*

Query Syntax

RES? **RES? MAX** **RES? MIN**
RES:TRIG? **RES:TRIG? MAX** **RES:TRIG? MIN**

Returned Parameters

<NR3> **RES?** and **RES:TRIG?** return the presently programmed input resistance levels. After a trigger or **ABOR**t, **RES:TRIG?** returns the same value as **RES?.**
RES? MAX, RES? MIN, RES:TRIG? MAX and **RES:TRIG? MIN** return the maximum and minimum programmable **RES** and **RES:TRIG** values for the present range.

Related Commands

ABOR **RES:RANG** **TRAN**sient Subsystem

RES:RANGe

Channel-Specific Resistance Command/Query

Description This command selects the full-scale resistance range of the electronic load. Programming any value equal to or greater than zero (≥ 0) and less than or equal to (≤) the maximum value of the lowest range automatically selects that range. Programming any value greater than the minimum range and less than or equal to (≤) the maximum value of the middle range automatically selects that range. *There is no overlap between the low and middle ranges.* Programming a value greater than the maximum value of the middle range automatically selects the highest range

Note Whenever the electronic load changes resistance range, the input momentarily goes into the OFF (minimum current) state.

The values of the input resistance **(IMM**ediate**, TRIG**gered and **TLEV**el) are adjusted as follows:

If	*Then*
The existing resistance setting is within the new range.	The resistance value does not change.
The existing resistance value is not within the new range.	The resistance is set to the closest limit of the new range.

For example, assume the Agilent 6060A Electronic Load; is in the 10000-ohm range and the input resistance (**RES:LEV**) is 2000 ohms. Switching to the 1000-ohm range will reduce the resistance to the closest limit of that range, or 1000 ohms. However, if another parameter (such as **RES:TRIG**) was already within the new range (e.g., 500 ohm), then it will remain at that level in the new range. When switching between two ranges that overlap, the accuracy of the programmed value may be compromised because of reduced resolution when going from a lower to a higher range.

Command Syntax **RES**istance:**RANG**e **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Value Coupling This command affects the following parameters:

RES:LEV   RES:TLEV

Examples **RES:50**   *Set resistance 50 ohms.*

RES:RANG 1; :TRIG 45E-3 *Change range to* 1 *ohm and set Triggered resistance to 45 milliohms. The main resistance will drop to 1 ohm and remain there until a trigger occurs.*

Query Syntax **RES:RANG?**
RES:RANG? MAX   RES:RANG? MIN

Returned Parameters **<NR3>**   **RES:RANG?** returns the present resistance range in ohms. **RES:RANG? MAX** and **RES:RANG? MIN** return the maximum and minimum programmable ranges for the electronic load.

Related Commands **RES:[LEV]   RES:TRIG   RES:TLEV**

RES:TLEVel

Channel-Specific Resistance Command/Query

Description This command specifies the value of the programmed resistance level for the **TRAN**sient input when the electronic load is in the CR (Constant-Resistance) Mode. When the Transient Subsystem is on, the electronic load input resistance will switch (under control of the Transient Subsystem) between the main resistance and **TLEV**el at a rate determined by the present value of **VOLT:SLEW** (lowest resistance range) or by **CURR:SLEW** (middle and high resistance ranges). In order for input resistance level switching to occur, **TLEV**el must be programmed as follows:

Lowest range	**TLEV**el > **LEV**el
Middle and highest ranges	**TLEV**el < **LEV**el

If the above rules are not followed, no error is generated but transient switching will not occur. If the main resistance is subsequently programmed to the proper level with respect to **TLEV**el, switching will begin provided the **TRAN**sient subsystem is on. If **TLEV**el is programmed outside the specified range, an error is generated (See Table 4-2 at the end of this chapter).

Command Syntax **RES**istance:**TLEV**el **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Value Coupling The higher values of resistance levels have less resolution. Programmed levels are set to the nearest value obtainable by the hardware.

Examples

`RES: TLEV MAX`	*Set transient level to maximum resistance for the range*

`RES:RANG 1000`
`RES 200; :RES:TLEV 400; :TRAN ON`
	Since TLEV> LEV on the 1k range, no transient switching occurs

`RES: TLEV 100`	*Now TLEV < LEV and input switches from 100 to 200 ohms*

Query Syntax **RES:TLEV?**

Returned Parameters **<NR3>** Transient resistance value for present range. If the electronic load is not in Resistance Mode, the value will still be set, even if it is less than the presently programmed input resistance.

Related Commands See **TRAN**sient Subsystem

STATus Subsystem

Channel-Specific & Channel-Independent Status Commands/Queries

Description The electronic load has the following four groups of device-dependent status registers:

Register Group	Registers
Channel Status	Condition, Enable, Event
Channel Summary	Enable, Event
Questionable Status	Condition, Enable, Event
Operation Status	Condition, Enable, Event, Transition Filters

Note See *Chapter 5 - Status Reporting* for details concerning the functions of these four groups of registers.

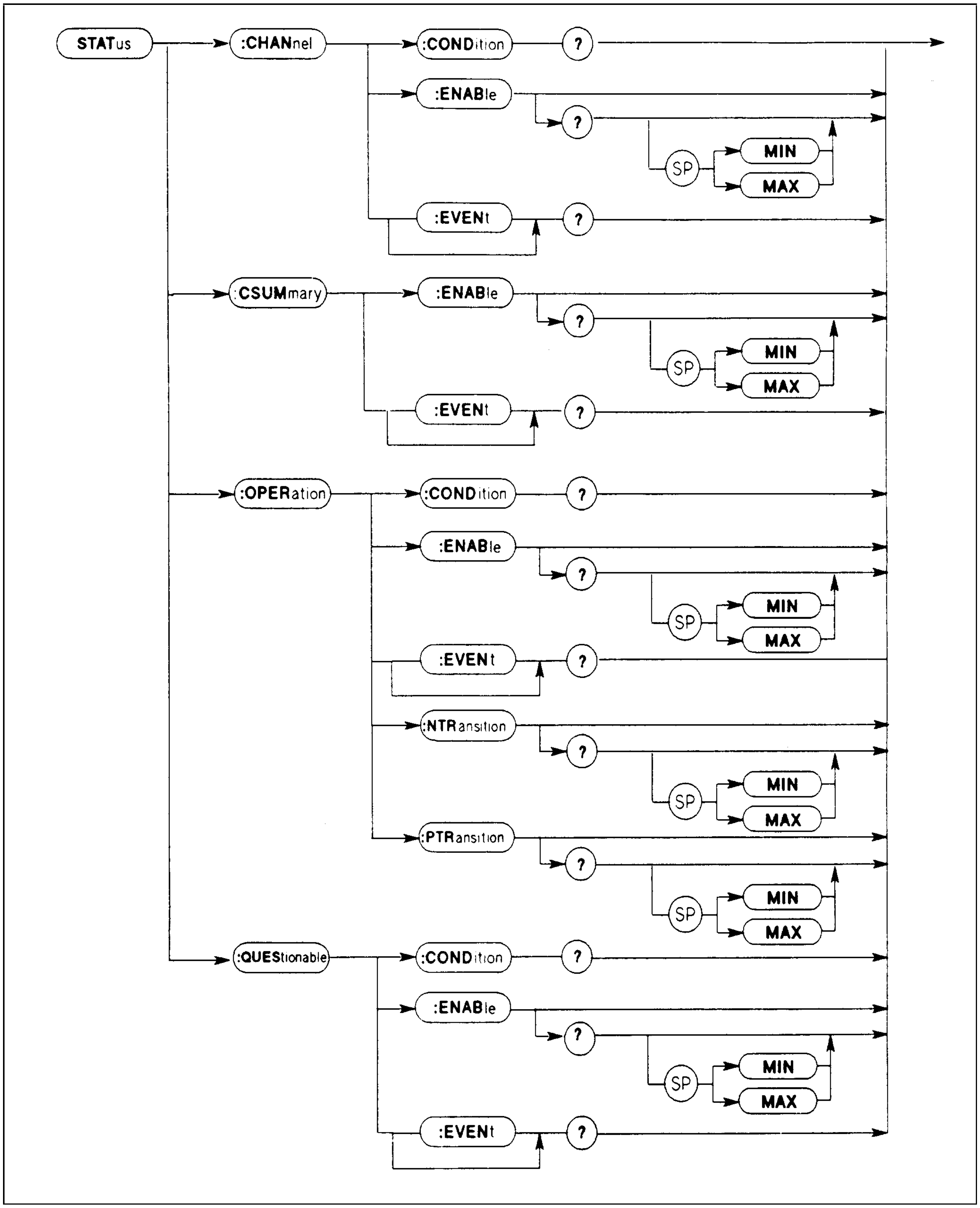
There are four groups of status command/queries corresponding to the above registers. The groups are placed in alphabetical order in the rest of this subsystem. The keywords for all groups are summarized below.

Keywords

STATus:**CHAN**nel	**:COND**ition?
STATus:**CHAN**nel	**:ENAB**le **:ENAB**le? **:ENAB**le? **MAX** **ENAB**le? **MIN**
STATus:**CHAN**nel	**[:EVEN**t**]**?
STATus:**CSUMmary**	**:ENAB**le **:ENAB**le? **:ENAB**le? **MAX** **ENAB**le? **MIN**
STATus:**CSUM**mary	**:EVEN**t?
STATus:**OPER**ation	**:COND**ition?
STATus:**OPER**ation	**:ENAB**le **:ENAB**le? **:ENAB**le? **MAX** **ENAB**le? **MIN**
STATus:**OPER**ation	**[:EVEN**t**]**?
STATus:**OPER**ation	**:NTR**ansition **:NTR**ansition? **:NTR**ansition? **MAX** **:NTR**ansition? **MIN**
STATus:**OPER**ation	**:PTR**ansition **:PTR**ansition? **:PTR**ansition? **MAX** **:PTR**ansition? **MIN**
STATus:**QUES**tionable	**:COND**ition?
STATus:**QUES**tionable	**:ENAB**le **:ENAB**le? **:ENAB**le? **MAX** **ENAB**le? **MIN**
STATus:**QUES**tionable	**[:EVEN**t**]**?

Related Commands/Queries ***CLS *ESE *ESR? *SRE *STB**

Syntax Diagram



Status Subsystem Syntax Diagram

STAT:CHANnel *Channel-Specific* Channel Status Command/Queries

Description The Channel Status group consists of a set of registers for each channel. Each channel register set monitors all events for that channel and sums them into a corresponding summary bit in the Event register of the Channel Summary group.

The following Channel Status registers are associated with each channel:

Condition	Real-time ("live") channel status.
Event	Records all channel events that occurred since the last time the register was read.
Enable	Mask for selecting which bits in the Event register are allowed to be summed into the corresponding channel bit of the Channel Summary Event register.

Remember The **STAT:CHAN** commands are channel specific. For the multiple electronic load first send a **CHAN**nel command to select the desired channel.

Bit Configuration of Channel Status Registers[1]

Bit Position	15,14	13	12	11	10	9	8-5	4	3	2	1	0
Condition	[2]NU	PS	OV	RV	UNR	EPU	NU	OT	OP	NU	OC	VE
Bit Weight		8192	4096	2048	1024	512		16	8	4	2	1

[1]See Chapter 5 -Status Reporting for explanations of the bit mnemonics. [2]NU = Always zero.

Command/Query Syntax

Command/Query	*Function*
STATus:**CHAN**nel:**COND**ition?	Returns the binary value of the Channel Status Condition register, which represents the present status. A condition exists if the corresponding bit = *1*.
STATus:**CHAN**nel[:**EVEN**t]?	Returns the binary value of the Channel Event register, which latches *0-to-1* transition of the channel conditions the first time they occur. The Event bit remains *1* even if the condition has since disappeared. *Reading this register resets it to zero.*
STATus:**CHAN**nel:**ENAB**le	A mask that specifies which bits of the Channel Status Event register will be allowed to be summed into the appropriate channel bit of the Channel Summary register. Set the bit to *1* to enable the condition. Program *MAX* to enable all allowable bits or *MIN* to disable them.
STATus:**CHAN**nel:**ENAB**le?	Returns the binary mask value of the Channel Status Enable register.

Returned Parameters **< NR1 >** Register binary value.

Examples `STAT: CHAN?` *Returns the value of the Event register in a single electronic load or of the presently addressed channel Event register in a multiple electronic load.*

STAT:CHAN:ENAB 18 Programs *the Channel Enable register to allow the occurrence of either an OC or an OT condition to set the corresponding bit in the Channel Summary Event register for the present channel.*

STAT:CHAN:EVEN?;COND? *Returns the values of the Event and Condition registers for the present channel. The Condition value is the status as it existed at the moment the Condition register was read.*

CHAN 2;STAT:CHAN:ENAB 19 *Programs the Channel Status Enable register to enable OV, OC. and OT conditions to set the Channel Summary Event bit for Channel 2.*

Related Commands/Queries **STAT:CSUM STAT:QUES *CLS INP:PROT:CLE *CLS INP:PROT:CLE**

STAT:CSUMmary

Channel-Independent Channel Summary Command/Queries

Description The Channel Summary group of registers provides a convenient way to determine if any enabled **CHAN:STAT** Events have occurred on a particular channel. The Channel Summary group is primarily intended for use with multiple electronic loads. However, all commands in this group are valid with single electronic loads. The following registers comprise the Channel Summary group:

Event Register	Indicates all channels on which an enabled STAT:CHAN Event has occurred since the last time the register was read.
Enable Register	Mask for selecting which bits in the Channel Event register are allowed to be summed into the CSUM (Channel Summary) bit of the Status Byte register.

Bit Configuration of Channel Summary Registers[1]

Bit Position	12	11	10	9	8	7	6	5	4	3	2	1	0
Condition	12	11	10	9	8	7	6	5	4	3	2	1	NU
Bit Weight	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

[1]Not all channel bits may be used. NU = Always zero.

Command/Query Syntax

Command/Query	*Function*
STATus:**CSUM**mary[:**EVEN**t]?	Returns the binary value of the Channel Summary Event register. A *1* in the bit position indicates that there is an event from that channel. *Reading this register resets it to zero.*
STATus:**CSUM**mary:**ENAB**le	A mask that specifies which bits of the Channel Summary Event register can be selected to set the Status Byte register CSUM bit. Set the bit to *l* to enable the bit. Program *MAX* to enable all allowable bits or *MIN* to disable them.
STATus:**CSUM**mary:**ENAB**le?	Returns the binary mask value of the Channel Summary Enable register.

Returned Parameters **<NR 1>** Register binary value

Examples **STAT:CSUM?** *Returns the value of the CSUMmary EVENt register*

STAT:CSUM:ENAB 18 *Programs the Channel Summary Enable register to allow the Channel Summary Event register to set the error summary bits for Channels 1 and 4.*

Related Commands/Queries **STAT:QUES** ***CLS**

STAT:OPERation

Channel-Independent Operation Status Command/Queries

Description The Operation Status registers provide information about the following operating conditions:

WTG A *l* indicates that at least one channel of the electronic load is waiting for a trigger to occur.

CAL A *1* indicates that the calibration constants of the channel are presently being recalculated. Unless you are explicitly calibrating the electronic load, this bit is always *0*.

The WTG bit is used to detect if the electronic load is waiting to complete the transfer of a pending trigger level. When the bit is set, it means that the electronic load has processed a trigger level command (e.g., **CURR:TRIG 25**) but is waiting for a trigger in order to execute that command.

Note Refer to the electronic load *Operating Manual* for information concerning Calibration commands.

The following registers are associated with Operation Status commands:

Register	**Function**
Condition	Real-time (“live”) operating status.
PTR/NTR	Programmable filters that determine what type of transition (0 to-1 or 1-to-0) in the Condition register will set the corresponding bit in the Event register.
Event	Records all channel conditions that occurred since the last time the register was read.
Enable	Mask for selecting which bits in the Event register are allowed to be summed into the OPER (Operation Summary) bit of the Status Byte register

Bit Configuration of Operation Status Registers[1]

Bit Position	11-6	5	4-1	0
Condition	[1]NU	WTG	[1]NU	CAL
Bit Weight		32		1
[1]NU = Not Used				

The following command/queries are associated with this register group:

Command/Query Syntax **Register**	
STATus:**OPER**ation:**COND**ition?	Returns the binary value of the Operation Status Condition register, which represents the present status. The condition is occurring whenever the bit is *l.*
STATus:**OPER**ation:**PTR**ansition	A filter that specifies whether or not a *0-to-1* transition in the Condition register will set the corresponding bit in the Event register. Program the bit position to *1* to enable the transition requirement.
STATus:**OPER**ation:**PTR**ansition?	Returns the binary value of the **PTR**ansition filter.
STATus:**OPER**ation:**NTR**ansition	A filter that specifies whether or not a *1*-to-*0* transition in the Condition register will set the corresponding bit in the Event resister. Program the bit position to *1* to enable the transition requirement.
STATus:**OPER**ation:**NTR**ansition?	Returns the binary value of the **NTR**ansition filter.
STATus:**OPER**ation [:**EVEN**t]?	Returns the binary value of the Operation Event register, which latches specified transition in the Operation Condition register the first time they occur. The event bit remains *1* even if the condition has since disappeared. *Reading this register resets it to 0.*
STATus:**OPER**ation:**ENAB**le	A mask that specifies which bits of the Operation Status Event register are allowed to be summed into the **OPER** (Operation) bit of the Status Byte register. Set the bit to *l* to enable the corresponding condition. Program *MAX* to enable all allowable bits or *MIN* to disable them.
STATus:**OPER**ation:**ENAB**le?	Returns the binary mask value of the Operation Status Enable register.

Note To specify a bit to be set on either a 0-to-l or a l-to-0 transition, program both filters (**NTR** and **PTR**) for that bit to *1*. To prevent a bit from being sent under any conditions, program that bit in both filters to *0*.

Returned Parameters **<NR 1>** Register binary value.

Examples

STAT:OPER? *Returns* the *binary value of the Operation Status Event register.*

STAT: OPER: COND? *Returns the status of the electronic load as it existed the instant the register was read.*

STAT:OPER:PTR 32;NTR 32 *Program a recorded event on any transition of the WTG bit.*

Note **ABOR**t can also cause a WTG bit event if the corresponding **NTR**ansition bit is set to *1*.

Related Commands/Queries **ABOR** ***CLS** **TRIG** ***TRG**

STAT:QUEStionable *Channel-Independent* Questionable Status Command/Queries

Description The Questionable Status register group provides information that some data or parameters may be unreliable. The Questionable Status registers monitor the same conditions as the Channel Status group. However, the Questionable Status monitors a specified condition for all channels in the multiple electronic load and sums them into the **QUES** (Questionable Summary) bit of the Status Byte register. This permits the controller to use a single command to determine if the specified condition exists on any of the channels. All commands associated with this group are valid with single electronic loads, but in that case the Questionable and Channel Status registers will hold the same information. The following three Channel Status registers are associated with each channel:

Condition	Real-time ("live") recording of Questionable data.
Event	Records all Questionable conditions that occurred since the last time the register was read.
Enable	Mask for selecting which bits on the Event register are allowed to be summed into the QUES bit of the Status Byte register.

Bit Configuration of Questionable Status Registers[1]

Bit Position	15,14	13	12	11	10	9	8-5	4	3	2	1	0
Condition	[1]NU	PS	OV	RV	UNR	EPU	NU	[2]TE	[2]PE	NU	[2]CE	[2]VE
Bit Weight		8192	4096	2048	1024	512		16	8		2	1

[1]NU = Not Used

[2]All signals are the same as the Channel Status Condition register.
Different mnemonics are required by the HPSL standard.

Note See *Chapter 5 - Status Reporting* for a explanations of the bit mnemonics.

Command/Query Syntax

Query	*Function*
STATus:**QUES**tionable:**COND**ition?	Returns the binary value of the Questionable Status Condition register, which represents real-time status of possible electronic load malfunctions. A condition exists if the corresponding bit = 1.
STATus:**QUES**tionable [:**EVEN**t]?	Returns the binary value of the Questionable Status Event register, which latches each *0-to 1* transition of the condition of the Questionable Condition register the first time it occurs. The bit remains *1* even if the original condition has since disappeared. *Reading this register resets it to zero.*
STATus:**QUES**tionable:**ENAB**le	A mask that specifies which bits of the Questionable Event register can be summed into the QUES bit of the Status Byte register. Set the bit to *l* to enable the corresponding event. Program *MAX* to enable all allowable bits or *MIN* to disable them.
STATus:**QUES**tionable:**ENAB**le?	Returns the binary mask value of the Questionable Status Enable register.

Returned Parameters	**<NR 1>**	Register binary value

Examples

STAT: QUES?	*Returns the binary value of the Questionable Status Event register and clears it.*
STAT:QUES:EVEN?;ENAB?	*Returns the binary values of the Questionable Status Event and Enable registers.*
STAT: QUES: ENAB 19	*Programs the Questionable Status Enable register to enable OV, OC, and OT conditions to set the Status Byte register QUES bit.*

Related Commands/Queries ***CLS** **:INP:PROT:CLE**

SYST:ERRor?

Channel-Independent Error Query

Description This query reads the remote programming error queue of the electronic load. The queue, which operates in a FIFO (first-in, first-out) mode, records only programming errors - not front panel errors. As it is read, each error is removed from the queue. When all errors have been read, a zero is returned.

Negatively numbered errors are generic HPSL errors and positively numbered errors are specific to the electronic load. (See Table 4-2 at the end of this chapter) If the error queue should become full, error *-350* will be returned.

Query Syntax **SYST**em**:ERR**or?

Returned Parameters

<aard or crd>	Gives the error number and a short description of the error.
Error Number Range	-32768 to 32767

Note For an error summary list, see Table 4-2 at the end of this chapter.

Examples **SYST: ERR?** *Returns the oldest (first) error* currently in *the system error queue*

Related Queries None

TRANsient Subsystem

Channel-Specific Transient Programming Function

Description The Transient Subsystem may be used with Constant-Current, Constant- Resistance, or Constant-Voltage load operation. The transient function provides an alternate input level (**TLEV**el) that occurs periodically under programmed control.

The three transient modes are:

CONTinuous	A continuous pulse train that alternates between **LEV**el and **TLEV**el under control of an internal oscillator.

PULSe	A one-shot pulse that alternates between **LEV**el and **TLEV**el upon occurrence of an explicit trigger.
TOGGle	Each explicit trigger causes the input to alternate between **LEV**el and **TLEV**el.

The input levels (**LEV**el and **TLEV**el) are programmed from the Current, Resistance, or Voltage Subsystems. So is **SLEW**, which determines the rate at which the input changes from one level to the other.

Note For a detailed description of transient subsystem operation, see *Operation Overview* in the Electronic Load *Operating Manual.*

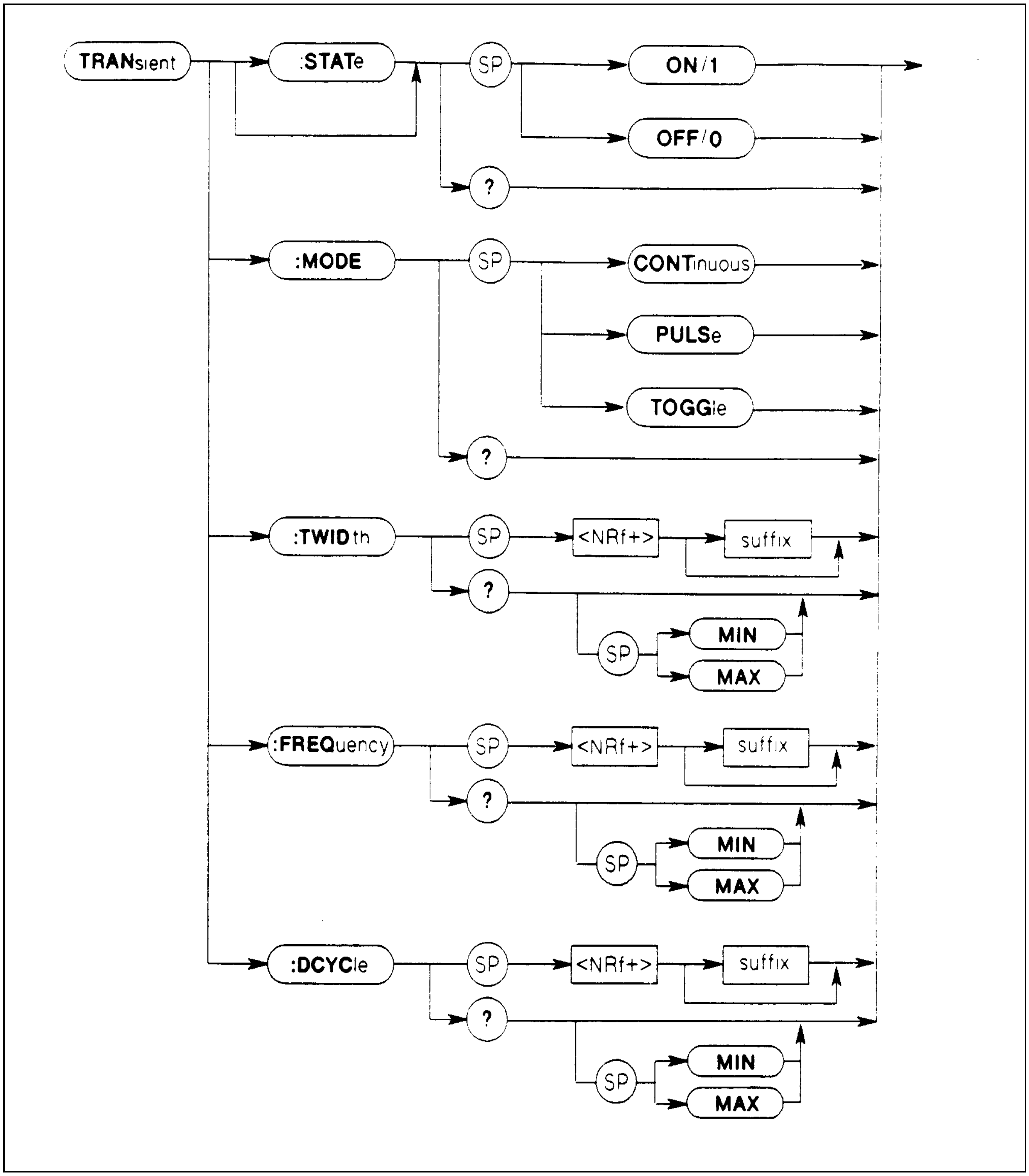
Keywords

Command	*Function*
TRANsient[:**STAT**e] **ON**|1	Enable Transient function.
TRANsient[:**STAT**e] **OFF**|0	Disable Transient function.
TRANsient:**MODE CONT**inuous	Select continuous pulse train varying between **LEV**el and **TLEV**el at **FREQ**uency.
TRANsient:**MODE PULS**e	Select single triggered pubes of duration **TWID**th.
TRANsient:**MODE TOGG**le	Select pulses that switch between **LEV**el and **TLEV**el on alternate triggers.
TRANsient:**DCYC**le	Set pulse duty cycle of **CONT**inuous mode.
TRANsient:**FREQ**uency	Set pulse frequency of **CONT**inuous mode.
TRANsient:**TWID**th	Set pulse duration of **PULS**e mode.

Related Commands **LEV**el <GET> ***TRG** **SLEW** **TLEV**el

Related Subsystem **TRIG**ger Subsystem

Syntax Diagram



Transient Subsystem Syntax Diagram

TRAN:DCYCle

Channel-Specific Transient Command/Query

Description **DCYC**le specifies the duty cycle of **TLEV**, as a percent of the total cycle, when the electronic load is in the **CONT**inuous mode. If programmed when the electronic load is in the **PULS**e or **TOGG**le mode, **DCYC**le is stored until the next time the electronic load is in **CONT**inuous mode. If a value is specified outside the parameter limits, an error is generated (See Table 4-2 at the end of this chapter) .

Command Syntax **TRAN**sient**:DCYC**le **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`TRAN: DCYC 10`	*Set Continuos output to **TLEV**el for 10% of the cycle.*
`TRAN: MODE CONT; FREQ 500; DCYC 10; STAT 1`	*Set Transient Generator to continuous mode, 500 Hz at 10% duty cycle and enable the generator.*

Query Syntax **TRAN:DCYC?** **TRAN:DCYC? MAX** **TRAN:DCYC? MIN**

Returned Parameters **<NR3>** **TRAN:DCYC?** returns present value of Continuous Mode duty cycle in percent.
TRAN:DCYC? MAX and **TRAN:DCYC? MIN** return the maximum and minimum programmable duty cycle values.

Related Commands **TRAN:FREQ**

TRAN:FREQuency

Channel-Specific Transient Command/Query

Description **FREQ**uency specifies the pulse frequency of the **CONT**inous mode. If specified in a **PULS**e or **TOGG**le mode**, FREQ**uency will be stored for the next time the electronic load is in the **CONT**inuous mode. If a value is specified outside the parameter limits, an error is generated (See Table 4-2 at the end of this chapter).

Command Syntax **TRAN**sient:**FREQ**uency **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`TRAN:FREQ 100`	*Set Transient Generator frequency to 100 Hz.*
`TRAN:FREQ 10 KHZ`	*Set Transient Generator frequency to 10 kHz.*
`TRAN:MODE CONT;FREQ 560;DCYC 50;STAT 1`	*Set Transient Generator to continuous mode, frequency to 560 Hz, duty cycle to 50% and enable the generator.*
`TRAN: STAT OFF; FREQ 2E4; STAT ON`	*Turn off Transient Generator change frequency to 20 kHz, and turn generator on.*

Query Syntax **TRAN:FREQ?** **TRAN:FREQ? MAX** **TRAN:FREQ? MIN**

Returned Parameters <NR3> **TRAN:FREQ?** returns the value representing continuous mode **FREQ**uency in Hz.
TRAN:FREQ? MAX and **TRAN:FREQ? MIN?** return the maximum and minimum programmable frequency values.

Related Commands **TRAN:DCYC**

TRAN:MODE

Channel-Specific Transient Command/Query

Description Selects the type of operation provided by the transient generator for each channel.

- **CONT**inous mode provides periodic pulses of programmable frequency and duty cycle, which are not related to **TWID**th in any way. Each channel has its own oscillator that runs asynchronously with respect to oscillators of other channels.

- **PULS**e mode provides a triggered one-shot pulse with a programmable pulse duration. When a trigger occurs, the input goes to **TLEV**el for the duration of **TWlD**th (which is not related to **FREQu**ency or **DCYC**le in any way) and then returns to **LEV**el. This mode is not retriggerable; triggers that occur while the input is at **TLEV**el are ignored. In multiple electronic loads, the trigger is applied to all channels simultaneously and **TRIG:SOUR TIM**er can be used to produce synchronized load waveforms (See **TRIG**er Subsystem).

- **TOGG**le mode causes the input to switch between **LEV**el and **TLEV**el on alternate triggers. Odd triggers set **TLEV**; even triggers set **LEV, FREQ, DCYC**, and **TWID** have no effect on this mode. Triggers are applied to all channels simultaneously in the multiple electronic load.

Note The Operation Status Condition register WTG bit (See *Chapter 5 - Status Reporting*) is not affected by setting either the **PULS**e or **TOGG**le modes.

Command Syntax **TRAN**sient**:MODE <aard>**

Parameters

Value Range	*RST Default
CONT, PULS, TOGG	*CONT*

Examples

`TRAN:MODE PULS`	*Set Transient Generator to pulse mode.*
`TRAN:STAT 0;MODE CONT;STAT 1`	*Turn off Transient Generator, change mode to continuos, and turn generator on*

Query Syntax **TRAN:MODE?**

Returned Parameters Character string *CONT, PULS,* or *TOGG*

Related Commands See **TRIG**ger Subsystem

TRAN[:STATe]

Channel-Specific Transient Command/Query

Description **STAT**e is an implied keyword that enables or disables the Transient Subsystem. When the subsystem is disabled, **TLEV**el inputs cannot occur. However, they may be programmed while the subsystem is disabled and will take effect when the subsystem is enabled. The enable or disable state may be programmed with either characters or equivalent numerics.

Command Syntax **TRAN**sient[:**STAT**e] **<NRf+>**

Parameters

Value Range	*RST Default
ON or *1;OFF or 0*	*0*

Examples

`TRAN ON` *Enable Transient Generator function.*

`TRAN:STAT l;MODE TOGG` *Enable Transient Generator function and set to toggle mode.*

`TRAN:STAT 0;MODE PULS;TWID l0E-3;STAT ON` *Disable Transient Generator function, set generator to Pulse Mode with a 10-millisecond pulse, and re-enable thegenerator function.*

Query Syntax **TRAN:STAT?**

Returned Parameters **<NRl>** **Value:** **0** for *OFF*, **1** for *ON*

Related Commands None

TRAN:TWIDth

Channel-Specific Transient Command/Query

Description **TWID**th specifies the duration of **TLEV**el during the **PULS**e mode. If specified when the electronic load is in the **CONT**inuous or **TOGG**le mode, **TWID**th is stored until the next use of **PULS**e mode. If a value is entered outside the specified parameters, an error is generated (See Table 4-2 at the end of this chapter).

Command Syntax **TRAN**sient:**TWID**th **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`TRAN: TWID 5E-5` *Set Transient Generator pulse duration to 50 microseconds.*

`TRAN:STAT l;MODE PULS;TWID .500` *Enable Transient Generator function, set mode to pulse, and program pulse width to 500 milliseconds.*

Query Syntax **TRAN:TWID?** **TRAN:TWID? MAX** **TRAN:TWID? MIN**

Returned Parameters **<NR3>** **TRAN:TWID?** returns value representing duration of **TLEV** in seconds. **TRAN:TWID? MAX** and **TRAN:TWID? MIN** return the maximum and minimum programmable duration values.

Related Commands None

TRIGger Subsystem

Channel-Independent Trigger Programming Function

Description In a multiple electronic load, triggering can be initiated from the controller, the a-c line, the external **Trigger** jack, or the mainframe timer. single electronic loads may be triggered from their external **Trigger** jack, or from the controller. Any trigger signal generated by a single electronic load or multiple electronic load also exists at the **Trigger** output jack and may be used as a trigger source for any other electronic load.

Note For a detailed explanation of the effects of triggers, see *Operation Overview* in the electronic load *Operating Manual.*

Keywords

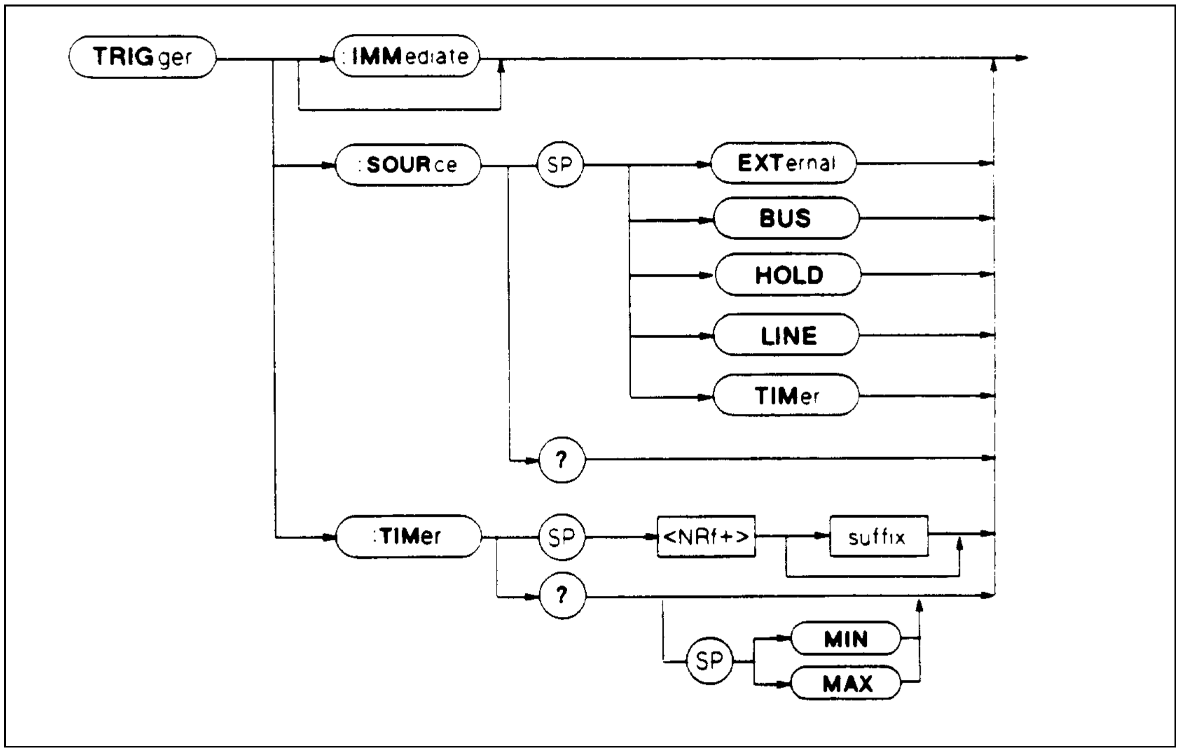
Command	*Function*
TRIGger[:**IMM**ediate]	Trigger-true input from the controller.
TRIGger:**SOUR**ce **EXT**ernal	Selects trigger source at external **Trigger** jack.
TRIGger:**SOUR**ce **BUS**	Selects GPIB ***TRG** or <GET> as trigger source.
TRIGger:**SOUR**ce **HOLD**	Disables all trigger inputs except the **TRlG:IMM** command.
[1]**TRIG**ger:**SOUR**ce **LINE**	Selects the internal a-c line frequency detector as the trigger source.
[1]**TRIG**ger:**SOUR**ce **TIM**er	Selects internal trigger oscillator as the trigger source.

[1] These commands apply only to multiple electronic loads

Related Commands ***TRG** <GET>

Related Subsystems **TRAN**sient Subsystem

Syntax Diagram



Trigger Subsystem Syntax Diagram

TRIG[:IMMediate]

Implied *Channel-Independent* Trigger Command

Description This implied keyword generates a trigger signal to the electronic load, regardless of which trigger **SOUR**ce is currently in effect. This is the only command that overrides **TRIG:SOUR HOLD**. When **TRIG:IMM** is executed, all pending triggered levels are transferred to the electronic load's input. The input also is affected if the transient generator is enabled in either the **PULS**e or **TOGG**le mode. In a multiple electronic load, a trigger is applied simultaneously to all channels.

Note When **TRIG:IMM** is executed in a single electronic load there is brief period (≤100 µs) during which the electronic load can also respond to either an external trigger or a <GET> command. This situation does not occur with multiple electronic loads.

TRIG:IMM affects the WTG bit of the Operation Status Event register (See *Chapter 5 - Status Reporting*).

Command Syntax **TRIG**ger

Parameters None

Examples

TRIG	*Immediate trigger commands*
TRIG:IMM	

Related Commands None

TRIG:SOURce

Channel-Independent Trigger Command/Query

Description

TRIGger**:SOUR**ce selects the electronic load trigger source as follows:

BUS Accepts a GPIB <GET> signal or ***TRG** command as the trigger source. This mode guarantees that all previous commands will be completed before the trigger is executed.

EXTernal Selects the electronic load's external **Trigger** jack as the trigger source. The **EXT**ernal trigger is processed asynchronously with respect to other commands. The user must ensure that the trigger occurs at a valid time with respect to commands that are already being processed or are pending.

Note

When **TRIG:SOUR BUS** is specified for a single electronic load there is brief period (100 µs) during which the electronic load can also respond to an external trigger. This situation does not occur with multiple electronic loads.

HOLD Only the **TRIG**ger:**IMM**ediate command causes a trigger in this mode. All others, including ***TRG**, are ignored.
[1]**LINE** This generates triggers in synchronization with the a-c line frequency.
[1]**TIM**er Selects the multiple electronic load's internal trigger oscillator as the trigger source. The oscillator begins running as soon as this command is executed. The oscillator period is selected with the **TRIG:TIM** command. **TRIG:SOUR TIM** can be used with **TRAN:MODE PULS** to generate synchronous, continuous transient waveforms on multiple channels.

[1]These modes exist only in the multiple electronic load. Programming them in a single electronic load will generate an error.

Command Syntax **TRIG**ger:**SOUR**ce

Parameters

[1]Value Range	Units	*RST Default
BUS, EXTernal, HOLD LINE, TIMer	None	*HOLD*

[1] The long or short form may be used.

Examples

```
TRIG                     Send immediate bus trigger.
TRIG:IMM

TRIG:SOUR HOLD;IMM       Disable all sources and send immediate bus trigger.
```

Query Syntax **TRIG:SOUR?**

Returned Parameters *BUS EXT HOLD LINE TIM*

Related Commands &410 <GET> ***TRG**

TRIG:TIMer

Channel-Independent Trigger Command/Query

Description This command determines the period of the trigger pulses generated by the multiple electronic load's internal trigger oscillator. The trigger oscillator begins running as soon as this command is executed. If the command is executed when the **TRIG:SOUR** is not **TIM**er, the programmed period will take effect when the next **TRIG:SOUR:TIM** statement is executed. If the **TIM** parameter is out of the specified limits, an error is generated (See Table 4-2 at the end of this chapter).

Note Do not confuse this command with **TRIG:SOUR TIM**. In that command, **TIM** is a parameter for **SOUR**ce. This **TIM** is on the same tree level as **SOUR**ce and has its own parameter.

Command Syntax **TRIG**ger:**TIM**er **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

TRIG:SOUR TIM;TIM .001	*Select time mode with frequency = 1 kHz.*
TRIG: TIM 5E-5	*Change trigger frequency to 20 kHz.*

Query Syntax **TRIG:TIM?**

Returned Parameters **<NR3>** Value representing period in seconds.

Related Commands None

VOLTage Subsystem

Channel-Specific Voltage Programming Function

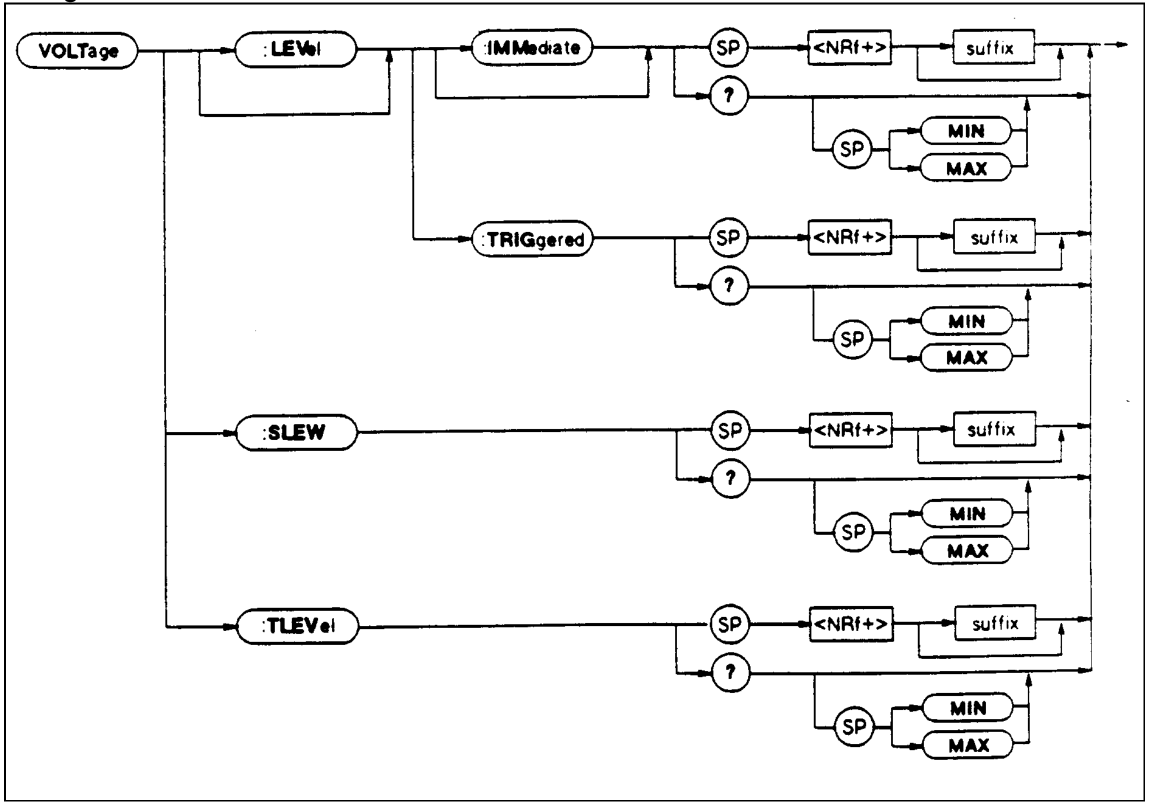
Description This subsystem programs the voltage-mode function of a single electronic load or a single channel of a multiple electronic load.

Keywords

Command	*Function*
VOLTage[:**LEV**el][:**IMM**ediate]	Specify main level for CV Mode.
VOLTage[:**LEV**el]:**TRIG**gered	Preset voltage level pending trigger occurrence.
VOLTage:**SLEW**	Specify voltage level rate of change.
VOLTage:**TLEV**el	Specify input voltage level used by **TRAN**sient subsystem.

Related Subsystems **CURR**ent **RES**istance

Syntax Diagram



Voltage Subsystem Syntax Diagram

VOLT[:LEVel]

Channel Specific Voltage Command/Query

Description This implied keyword specifies the value of the programmed voltage level and whether that level is to be applied immediately or on occurrence of a trigger. If the specified channel is in the Voltage Mode, an **IMM**ediate voltage level is transferred to the input as soon as the command is executed. A **TRIG**gered level is stored and transferred to the input when a trigger occurs. At that time, the change to the input level occurs at the slew rate presently in effect. Following the trigger event, subsequent triggers will not affect the input level unless the electronic load has been sent another **TRIG**gered level command.

If the electronic load is not in the CV (Constant-Voltage) Mode when an **IMM**ediate or **TRIG**gered level command is sent, the programmed levels are saved for the next time the electronic load is placed in the CV Mode. Triggered levels are processed by the Voltage Subsystem even when the electronic load is not in the CV Mode. Thus, the **TRIG**ered level becomes a stored **IMM**ediate level that takes effect when the electronic load is again in the CV Mode.

Note Setting an **IMM** voltage level to the same value as the most recent **TRIG** voltage level will not deactivate a pending **TRIG** level. You must use **ABOR**t to deactivate it.

The present voltage level changes to the pending level on any of the following conditions:

- On a **TRIG**[**:IMM**] command (always).
- On receipt of an external trigger signal (if **TRIG:SOUR** is set to **EXT**).
- On the next line voltage cycle (if **TRIG:SOUR** is set to **LINE**).
- On receipt of ***TRG** (unless **TRIG:SOUR** is set to **HOLD**).
- On receipt of a GPIB <GET> (if **TRIG:SOUR** is set to BUS).
- On the next trigger timer pulse (if multiple electronic load is set to **TRIG:SOUR TIM**).

Command Syntax **VOLT**age[:**LEV**el][**:IMM**ediate] **<NRf+>**
VOLTage[:**LEV**el]:**TRIG**gered **<NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Status and Errors **TRIG**gered level commands affect the **WTG** bit in the Operation Condition register and the **OPC** bit in the Standard Event Status Event register (refer to *Chapter 5 - Status Reporting).*

Examples

VOLT 25 **VOLTAGE: LEVEL 25**	I*mmediate commands for 25-volt input.*
VOLT:TRIG 25MV **VOLTAGE:LEVEL; TRIGGERED 25E-3**	*Commands for 25 mV input on occurrence of trigger.*
VOLT 12; :VOLT:TRIG MIN	*Set input to 12 volts now and minimum voltage when trigger occurs.*

Query Syntax **VOLT?** **VOLT? MAX** **VOLT? MIN**
VOLT:TRIG? **VOLT:TRIG? MAX** **VOLT:TRIG? MIN**

Returned Parameters **<NR3>** **VOLT?** and **VOLT:TRIG?** return the presently programmed voltage levels. After a trigger or **ABOR**t, **VOLT:TRIG?** returns the same value as **VOLT?.**

VOLT? MAX, VOLT? MIN, VOLT:TRIG? MAX,and **VOLT:TRIG? MIN** return the maximum and minimum programmable **LEV**el and **TLEV**el *values* for the present range.

Related Commands **ABOR** See also **TRAN**sient Subsystem

VOLT:SLEW

Channel-Specific Voltage Command/Query

Description This command sets the voltage programming slew rate and the resistance programming slew rate for lowest CR range. The programmed slew rate remains in effect for all programmed voltage changes except **INP**ut **ON** or **OFF**. The hardware implements discrete slew rates (refer to the electronic load *Operating Manual)* and automatically selects the one that is closest to the programmed value. To determine the actual value, use the query **VOLT:SLEW?**.

Command Syntax **VOLT**age**:SLEW < NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`VOLT:SLEW 5E6`	*Set slew rate for 5,000,000 Volts/Sec.*
`VOLTAGE:SLEW  MAX`	
`VOLT:TRIG 60MV; :VOLT:SLEW 1E4`	*Triggered input of 60 mV to be slewed at 10000 volts/second.*

Note Programming the slew rate value greater than *MAX* sets the slew rate to maximum without generating an error message.

Query Syntax
VOLT SLEW?
VOLT:SLEW? MAX
VOLT:SLEW? MIN

Returned Parameters **<NR3>** **VOLT:SLEW?** returns the internally selected slew rate (in V/s) that was chosen as closest to the programmed value within the permissible range. **VOLT:SLEW? MAX** and **VOLT:SLEW? MIN** return the maximum and minimum programmable slew rates for the present range.

Related Commands None

VOLT:TLEVel

Channel-Specific Voltage Command/Query

Description This command specifies the value of the programmed voltage level for the **TRA**Nsient input when the electronic load is in the CV (constant-voltage) Mode. When the Transient Subsystem is on, the electronic load input voltage will switch (under control of the Transient Subsystem) between the main level and **TLEV**el at a rate determined by the present value of **SLEW**. In order for input voltage level switching to occur, **TLEV**el must be greater than the main level. If **TLEV**el is set to a value below the main level, no error is generated but switching will not occur until the main level is subsequently programmed below the value of **TLEV**el. If **TLEV**el is programmed outside the specified range, an error is generated (See Table 4-2 at the end of this chapter).

Command Syntax **VOLT**age**:TLEV**el **< NRf+>**

Parameters See Table 4-1 and the Operating Manual of the electronic load model.

Examples

`VOLT:TLEV MAX`	*Set transient level to maximum voltage.*
`VOLT 40:TLEV 60`	*Set main voltage level to 40 V and transient level to 60 V.*

Query Syntax
VOLT:TLEV?
VOLT:TLEV? MAX
VOLT:TLEV? MIN

Returned Parameters	<NR3>	**VOLT:TLEV?** returns the transient voltage level for present range. If the electronic load is not in CV Mode, the level will still be set, even if it is less than the presently programmed input level. **VOLT:TLEV? MAX** and **VOLT:TLEV? MIN** return the maximum and minimum programmable values for the present range.
Related Commands	**TRAN**sient Subsystem	

Command and Parameters Summary

Table 4-1 lists all electronic load common commands in alphabetical order, followed by all root-level commands in alphabetical order. For the numerical parameters of a specific electronic load model, refer to the ’’Programming Ranges’’ table in that model’s Operating Manual. See Chapter 2 in this manual if you are not familiar with the representations used as data types.

Error Messages

Table 4-2 lists the error numbers and associated error messages that apply to the electronic load. The error number is the value that is placed in the electronic load’s error queue. This can be read back using the **SYST:ERR?** query, which returns the error number into a variable, and both the error number and error message into a string. Information inside the brackets is not part of the standard error message, but is included for clarification. Command errors (-100 through -199) set bit 5 in the Standard Event Status register. Execution errors (-200 through -299) set bit 4 in the Standard Event Status register. Device-dependent errors (-300 through -399) set bit 3 in the Standard Event Status register. Query errors (400 through 499) set bit 2 in the Standard Event Status register. See *Chapter 5 - Status Reporting* for a complete description of the Standard Event Status register.

Hardware Errors During Turn-On Selftest

If a GPIB failure occurs during selftest, the electronic load may or may not be able to communicate with the controller. If it can, error -330 (Self Test Error) is placed in the error queue. If an electronic load input error occurs during selftest, error -330 and error -240 (Hardware Error) are placed in the queue. If input errors occur, the electronic load will generate one or more -240 errors for each command it cannot process. Use the **SYST:ERR?** query to read the error queue after turn-on to ensure that no -330 or -240 errors are present. Any electronic load value returned by a query (such as **CURR?**) should be assumed to be invalid.

Hardware Errors During Operation

If an error does not occur during selftest but after the electronic load has been operating correctly for a time, error -240 is placed in the error queue. Most subsequent commands will not be executed and will also cause one or more -240 errors to be placed in the queue after each attempt. If your application requires the controller to be signaled if the electronic load fails, program the status registers to allow Execution errors or Device-dependent errors to generate an SRQ to the controller. See *Chapter 5 - Status Reporting* for more details.

Table 4-1. Summary of Commands and Parameters
(For numerical parameters, refer to the Operating Manual of the specific electronic load model.)

Command	*Parameters*	*Command*	*Parameters*	*Type*[1]
		Common Commands		
***CLS**	(none)	***RCL**	(space)**<NRf>**	CI
***ESE**	(space)**<NRf>**	***RDT?**	(none)	
***ESE?**	(none)	***RST?**	(none)	
***ESR?**	(none)	***SAV**	(space)**<NRf>**	
***IDN?**	(none)	***SRE**	(space)**<NRf>**	
***OPC**	(none)	***SRE??**	(none)	
***OPC?**	(none)	***STB?**	(none)	
***OPT?**	(none)	***TRG**	(none)	
***PSC**	(space)**<NRf>**	***TST?**	(none)	
***PSC?**	(none)	***WAI**	(none)	

Command	*Parameters*	*Type*[1]
	Root-Level *Commands*	
ABOR	(none)	CI
CHAN[:LOAD]	(space)**<NRf+>**	CI
CHAN[:LOAD]?	(none) or (space)**MIN** or (space)**MAX**	
NOTE: **INST** may be used as an alias for **CHAN**		
CURR[:LEV][:IMM]	(space)**<NRf+>**[suffix]	CS
CURR[:LEV][:IMM]?	(none) or (space)**MIN** or (space)**MAX**	
CURR[:LEV]TRIG	(space)**<NRf+>**[suffix]	
CURR[:LEV]TRIG?	(none) or (space)**MIN** or (space)**MAX**	
CURR[:PROT][:LEV]	(space)**<NRf+>**[suffix]	
CURR[:PROT][:LEV]?	(none) or (space)**MIN** or (space)**MAX**	
CURR:PROT:DEL	(space)**<NRf+>**[suffix]	
CURR:PROT:DEL?	(none) or (space)**MIN** or (space)**MAX**	
CURR:PROT:STAT	(space)**OFF** or **0;**(space)**ON** or **1**	
CURR:PROT:STAT?	(none)	
CURR:RANG	(space)**<NRf+>**[suffix]	
CURR:RANG?	(none) or (space)**MIN** or (space)**MAX**	
CURR:SLEW	(space)**<NRf+>**[suffix]	
CURR:SLEW?	(none) or (space)**MIN** or (space)**MAX**	
CURR:TLEV	(space)**<NRf+>**[suffix]	
CURR:TLEV?	(none) or (space)**MIN** or (space)**MAX**	
INP:PROT:CLE	(none)	CS
INP:SHOR[:STAT]	(space)**OFF** or **0;**(space)**ON** or **1**	
INP:SHOR[:STAT]?	(none)	
INP[:STAT]	(space)**OFF** or **0;**(space)**ON** or **1**	
INP[:STAT]?	(none)	
NOTE: **OUTP** may be used as an alias for **INP**		
MEAS:CURR[:DC]	(none)	CS
MEAS:POW[:DC]	(none)	
MEAS:VOLT[:DC]	(none)	
MODE:CURR[:DC]	(none)	CS
MODE:RES	(none)	
MODE:VOLT[:DC]	(none)	
MODE?	(none)	
NOTE: **FUNC** may be used as an alias for **MODE**		
PORT0	(space)**OFF** or **0** (space)**ON** or **1**	CS

Table 4-1. Summary of Commands and Parameters (continued)

Command	*Parameters*	*Type*[1]
RES[:LEV][:IMM]	(space)**<NRf+>**[suffix]	CS
RES[:LEV][:IMM]?	(none) or (space)**MIN** or (space)**MAX**	
RES[:LEV]:TRIG	(space)**<NRf+>**[suffix]	
RES[:LEV]:TRIG?	(none) or (space)**MIN** or (space)**MAX**	
RES:RANG	(space)**<NRf+>**[suffix]	
RES:RANG?	(none) or (space)**MIN** or (space)**MAX**	
RES:TLEV	(space)**<NRf+>**[suffix]	
RES:TLEV?	(none) or (space)**MIN** or (space)**MAX**	
STAT:CHAN:COND?	(none)	CS
STAT:CHAN:ENAB	(space)**<NRf+>**	
STAT:CHAN:ENAB?	(none) or (space)**MIN** or (space)**MAX**	
STAT:CHAN[:EVEN]?	(none)	
STAT:CSUM:ENAB	(space)**<NRf+>**	CI
STAT:CSUM:ENAB?	(none) or (space)**MIN** or (space)**MAX**	
STAT:CSUM[:EVEN]?	(none)	
STAT:OPER:COND?	(none)	
STAT:OPER:ENAB	(space)**<NRf+>**	
STAT:OPER:ENAB?	(none) or (space)**MIN** or (space)**MAX**	
STAT:OPER[:EVEN]?	(none)	
STAT:OPER:NTR	(space)**<NRf+>**	
STAT:OPER:NTR?	(none) or (space)**MIN** or (space)**MAX**	
STAT:OPER:PTR	(space)**<NRf+>**	
STAT:OPER:PTR?	(none) or (space)**MIN** or (space)**MAX**	
STAT:QUES:COND?	(none)	
STAT:QUES:ENAB	(space)**<NRf+>**	
STAT:QUES:ENAB?	(none) or (space)**MIN** or (space)**MAX**	
STAT:QUES[:EVEN]?	(none)	
SYST:ERR?	(none)	CI
TRAN:DCYC	(space)**<NRf+>**[suffix]	CS
TRAN:DCYC?	(none) or (space)**MIN** or (space)**MAX**	
TRAN:FREQ	(space)**<NRf+>**[suffix]	
TRAN:FREQ?	(none) or (space)**MIN** or (space)**MAX**	
TRAN:MODE	(space)**CONT** or (space)**PULS** or (space)**TOGG**	
TRAN:MODE?	(none)	
TRAN:[:STAT]	(space)**OFF** or **0;**(space)**ON** or **1**	
TRAN:[:STAT]?	(none)	
TRAN:TWID	(space)**<NRf+>**[suffix]	
TRAN:TWID?	(none) or (space)**MIN;**or (space)**MAX**	
TRIG[:IMM]	(none)	CI
TRIG:SOUR	(space)**BUS** or (space)**EXT** or (space)**HOLD** or (space)**LINE** or (space)**TIM**	
TRIG:SOUR?	(none)	
TRIG:TIM	(space)**<NRf+>**	
TRIG:TIM?	(none) or (space)**MIN** or (space)**MAX**	
NOTE: LINE and TIM not valid for Single Electronic Loads		
VOLT[:LEV][:IMM]	(space)**<NRf+>**[suffix]	CS
VOLT[:LEV][:IMM]?	(none) or (space)**MIN** or (space)**MAX**	
VOLT[:LEV]TRIG	(space)**<NRf+>**[suffix]	
VOLT[:LEV]TRIG?	(none) or (space)**MIN** or (space)**MAX**	
VOLT:SLEW	(space)**<NRf+>**[suffix]	
VOLT:SLEW?	(none) or (space)**MIN** or (space)**MAX**	
VOLT:TLEV	(space)**<NRf+>**[suffix]	
VOLT:TLEV?	(none) or (space)**MIN** or (space)**MAX**	
[1] CI = channel independent	CS = channel specific	

Table 4-2. Summary of Error Messages

Error Number	Error String (Description/Explanation/Examples)
-100	Command error [generic]
-101	Invalid character
-102	Syntax error [unrecognized command or data type]
-103	Invalid separator
-104	Data type error [e.g., "numeric or string expected, got block data’’]
-105	GET not allowed
-108	Parameter not allowed [too many parameters]
-109	Missing parameter [too few parameters]
-112	Program mnemonic too long [maximum 12 characters]
-113	Undefined header [operation not allowed]
-121	Invalid character in number [includes "9" in octal data "#q", etc.]
-123	Exponent too large [numeric overflow; exponent magnitude >32 k]
-124	Too many digits [number too long; more than 255 digits received]
-128	Numeric data not allowed
-131	Invalid suffix [unrecognized units, or units not appropriate]
-138	Suffix not allowed
-141	Invalid character data [bad character, or unrecognized]
-144	Character data too long [maximum length is 12 characters]
-148	Character data not allowed
-150	String data error
-151	Invalid string data [e.g., END received before close quote]
-158	String data not allowed
-160	Block data error
-161	Invalid block data [e.g., END received before length satisfied]
-168	Block data not allowed
-170	Expression error
-171	Invalid expression [e.g., illegal character in expression]
-178	Expression data not allowed
-180	Macro error
-181	Invalid outside macro definition [e.g., ’$1’ outside macro definition]
-183	Invalid inside macro definition
-200	Execution error [generic]
-220	Parameter error
-221	Settings conflict [uncoupled parameters]
-222	Data out of range [e.g., frequency too high for this instrument]
-223	Too much data [out of memory; block, string, or expression too long]
-240	Hardware error
-310	System error
-313	Calibration memory lost [out of cal due to memory failure]
-330	Self-test failed [more specific data after ";"]
-350	Too many errors [error queue overflow]
-400	Query error
-410	Query INTERRUPTED [query followed by DAB or GET before response complete]
-420	Query UNTERMINATED [addressed to talk, incomplete program message received]
-430	Query DEADLOCKED [input buffer and output buffer full; can’t continue]
-440	Query UNTERMINATED after indefinite response [indefinite response request not the last request in message unit]

5

Status Reporting

This chapter discusses the status data structure of the electronic loads as shown in Figure 5-1. The Standard Event Status register group, the Output Queue, and the Status Byte and Service Request Enable registers perform standard GPIB functions and are defined in *IEEE 488. 2 Standard Digital Interface for Programmable Instrumentation.* Other status register groups implement the status reporting requirements of the electronic load. The Channel Status and Channel Summary groups are primarily used by multiple electronic loads. This is because each channel in a multiple electronic load has its own Status register to provided status information for that channel.

General Register Model

The Condition register represents the present or "live" state of various electronic load signals. Reading the Condition register does not change the state of its bits. Only changes in electronic load conditions change the contents of this register. Not all status register groups have a Condition register. In some cases, such as the Standard Event Status registers, conditions are directly input into an Event register. In other cases, such as the Channel Summary registers, summary information from other registers is directly input into an Event register.

The Event register captures changes in conditions. Each bit in an Event register either corresponds to a condition bit in a Condition register, or to a specific condition in the electronic load. An event becomes true when the associated condition makes one of the following electronic load-defined transitions:

- *Positive TRansition* (0-to-1)
- *Negative TRansition* (1-to-0)
- *Positive or Negative TRansition* (1-to-0 or 0-to-1)

The PTR/NTR filters determine what type of condition transitions set the bits in the Event register. Only the operation Status registers allow transitions to be programmed. All other register groups use an implied 0-to-1 condition transition to set bits in the Event register. Reading an Event register clears the register (all bits set to zero). When the electronic load is turned on, Event registers are set to zero, and the PTR/NTR filters are set to their firmware assigned states.

The Enable register selects which bits in the corresponding Event register are logically-ORed into the Summary bit. At turn-on, Enable registers are set to zero. However, the Standard Event Enable register and the Service Request Enable register are not set to zero if the ***PSC** command is programmed. These registers are set to the most recent values saved in non-volatile memory before the Electronic Load was last turned off.

Channel Status

The Channel Status registers inform you that one or more channel status conditions, which indicate the presence of certain errors or faults, have occurred on a specific channel. Table 5-1 describes the channel status conditions that apply to the electronic load.

The Channel Status Condition register represents the present status of a channel; the bits are set when the indicated condition is true.

The Channel Status Event register records all of the channel conditions that have occurred since the last time this register was read. A condition transition from 0-to-l on a bit in the Channel Status Condition register will set the corresponding bit in the Channel Status Event register. Reading the Channel Status Event register resets it to zero.

The Channel Status Enable register can be programmed to specify which channel status event bits are logically-ORed to become the corresponding channel bit in the Channel Summary Event register.

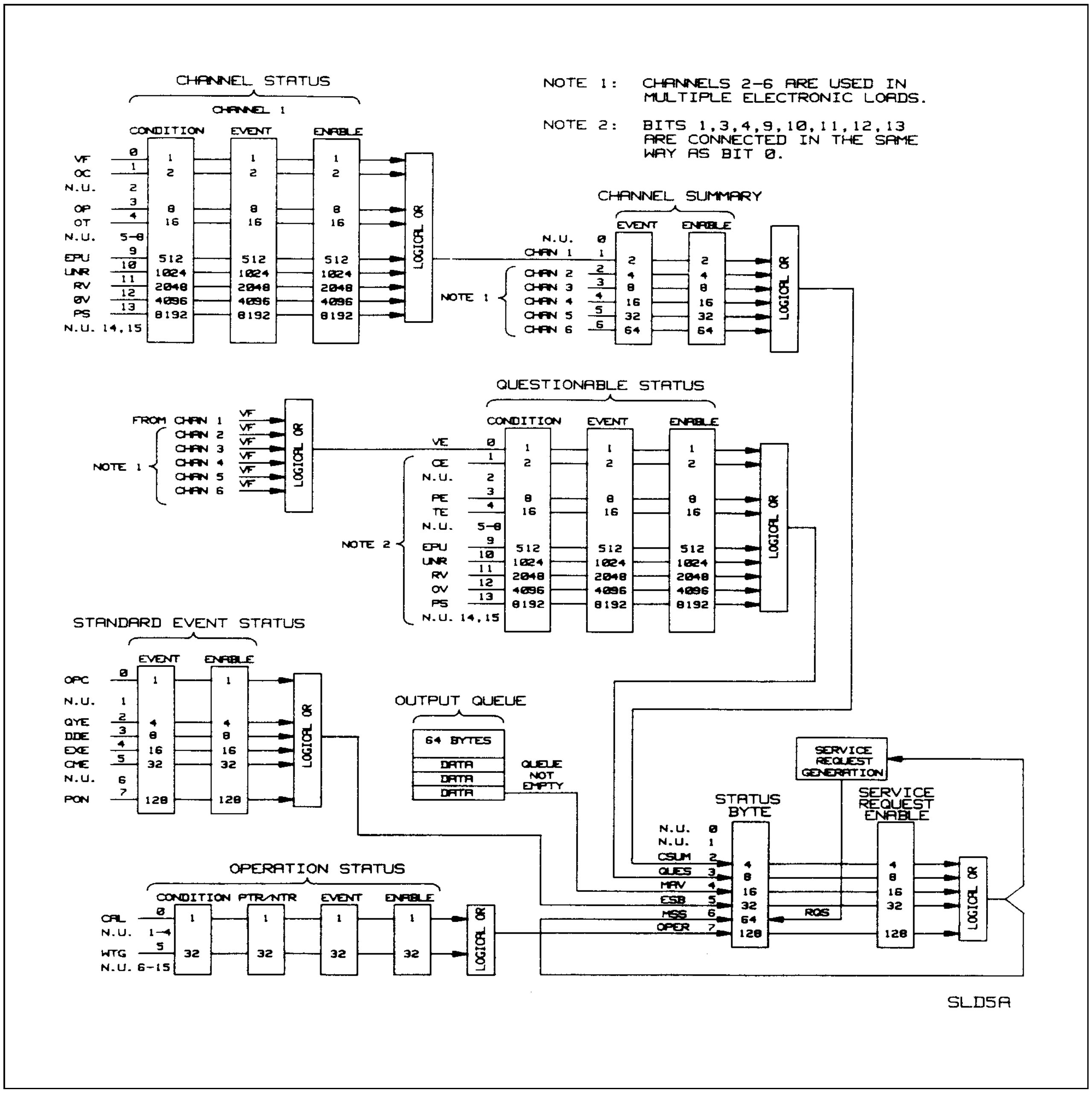


Figure 5-1. Electronic Load Status Registers

Channel Summary

The Channel Summary registers can summarize the channel status conditions of up to six channels. The channel/bit assignments in the Channel Summary registers are as follows:

Channel	Bit	Value
Channel 1	1	2
Channel 2	2	4
Channel 3	3	8
Channel 4	4	16
Channel 5	5	32
Channel 6	6	64

When an enabled bit in the Channel Status Event register is set, it causes the corresponding channel bit in the Channel Summary Event register to be set. Reading the Channel Summary Event register resets it to zero.

The Channel Summary Enable register can be programmed to specify which channel summary event bits from the existing channels are logically-ORed to become Bit 2 (CSUM bit) in the Status Byte register. For single electronic loads, only Channel exists.

Table 5-1. Channel Status Bit Description

Mnemonic	Bit[1]	Value	Meaning
VF	0	1	*Voltage Fault*. Either an overvoltage or a reverse voltage condition has occurred on a channel. When either of these conditions occur, Bit 0 is set and remains set until **INP:PROT:CLE** is programmed. Note that this bit reflects the active state of the **Flt** pin on the back of the unit.
OC	1	2	*Overcurrent.* An overcurrent condition has occurred on a channel. This condition sets Bit 1 if the current exceeds 102% of the rated current, or if the current exceeds the user-programmed current protection level. If the overcurrent condition is removed, Bit 1 is cleared. However, if the user-programmed overcurrent condition persists beyond the user-programmed current protection delay time, Bit 13 is also set and the channel is turned off. In this case, Bits 1 and 13 remain set until the overcurrent condition is removed and **INP:PROT:CLE** is programmed.
OP	3	8	*Overpower* An overpower condition has occurred on a channel. This condition sets Bit 3 when the internal overpower protection circuit is limiting the input power. This occurs if the unit exceeds the rated power of a channel. However, if an overpower condition occurs and persists for more than 3 seconds. Bit 13 (PS bit) is also set and the channel is turned off. In this case, Bits 3 and 13 remain set until the overpower condition is removed and **INP:PROT:CLE** is programmed.

Table 5-1. Channel Status Bit Description (continued)

Mnemonic	Bit[1]	Value	Meaning
OT	4	16	*Overtemperature.* An overtemperature condition has occurred on a channel. When this occurs, both Bit 4 and Bit 13 (PS bit) are set and the channel is turned off. Bits 4 and 13 remain set until the channel (or unit) has cooled down well below the overtemperature trip point and **INP:PROT:CLE** is programmed.
EPU	9	512	*Extended Power Unavailable.* This bit has no significance in later "A" version and in all ’’B’’ version electronic loads.
UNR	10	1024	*Unregulated Input.* A channel is unregulated. This condition sets Bit 10. When the load becomes regulated, Bit l0 is cleared. Unregulated input does not occur in CV mode or in the 1 ohm range of CR mode.
RV	11	2048	*Reverse Voltage on input.* A channel has a reverse voltage applied to it. When this occurs, both Bit 11 and Bit 0 (VF bit) are set. When the reverse voltage is removed, Bit 11 is cleared. However, Bit 0 remains set until **INP:PROT:CLE** is programmed.
OV	12	4096	*Overvoltage.* An overvoltage condition has occurred on a channel. When this occurs, both Bit 12 and Bit 0 (VF bit) are set and the FETs are turned on as hard as possible to lower the voltage. Bits 12 and 0 remain set until the overvoltage condition is removed and **INP:PROT:CLE** is programmed.
PS	13	8192	*Protection Shutdown.* A channel has turned off because of an overcurrent, overpower, or overtemperature condition. When any of these three conditions occur, Bit 13 is set and remains set until **INP:PROT:CLE** is programmed.

[1]Bits 2, 5-8, 14 and 15 are not used by the electronic load.

Questionable Status

The Questionable Status registers inform you that one or more questionable status conditions, which indicate the presence of certain errors or faults, have occurred on at least one channel. This lets you check for specific errors or faults that have occurred without having to poll each channel individually. Table 5-2 lists the questionable status conditions that apply to the electronic load. These conditions are the same as the channel status conditions. Refer to Table 5-1 for a complete description.

The Questionable Status Condition register represents the present status of all channel conditions; the bits are set when the indicated condition is true.

The Questionable Status Event register represents all of the conditions that have occurred since the last time this register was read. A condition transition from 0-to-1 on a bit in the Questionable Status Condition register will set the corresponding bit in the Questionable Status Event register. Reading the Questionable Status Event register resets it to zero.

The Questionable Status Enable register can be programmed to specify which questionable status event bits are logically-ORed to become Bit 3 (QUES bit) in the Status Byte register.

Table 5-2. Questionable Status Bit Description

Mnemonic	Bit[1]	Value	Meaning
VE/VF	0	1	Voltage Error (Voltage Fault)
CE/OC	1	2	Current Error (Overcurrent)
PE/OP	3	8	Power Error (Overpower)
TE/OT	4	16	Temperature Error (Overtemperature)
EPU	9	512	Extended Power Unavailable
UNR	10	l024	Unregulated input
RV	11	2048	Reverse Voltage on input
OV	12	4096	Overvoltage
PS	13	8192	Protection Shutdown

[1]Bits 2, 5-8, 14, and 15 are not used

Output Queue

The Output Queue is a data structure that stores output messages until they are read from the electronic load. The Output Queue stores messages sequentially on a FIFO (first-in, first-out) basis. When there is data in the queue, it sets Bit 4 (MAV bit) in the Status Byte register.

Standard Event Status

The function of the Standard Event Status register is standard on all IEEE 488.2 devices. Table 5-3 describes the standard events that apply to the electronic load. Note that all programming errors that have occurred will set one or more of the error bits in the Standard Event Status register. Programming errors are listed in Table 4-2.

The Standard Event Status register represents all of the standard events that have occurred since the last time this register was read. Reading the Standard Event Status register resets it to zero.

The Standard Event Enable register can be programmed to specify which standard event bits are logically-ORed to become Bit 5 (ESB bit) in the Status Byte register.

Note The present settings of the Standard Event Enable register can be saved in non-volatile memory if ***PSC** is programmed to zero. The next time the unit is turned on, the Standard Event Enable register will be programmed according to the saved settings.

Table 5-3. Standard Event Status Bit Description

Mnemonic	Bit[1]	Value	Meaning
OPC	*0*	*1*	*Operation Complete.* The electronic load has completed all pending operations. Programming *OPC causes this bit to be set when the electronic load completes all pending operations
QYE	2		*Query Error.* The output queue was read when no data was present or the data in the queue was lost. Errors in the range of -499 thru -400 can set this bit.
DDE	3	8	*Device Dependent Error.* Memory was lost, or self-test failed. Errors in the range of -399 thru -300 can set this bit.
EXE	4	16	*Execution Error*. A command parameter was outside the legal range or inconsistent with the electronic load's operation, or the command could not be executed due to some operating condition. Errors in the range of -299 thru -200 can set this bit.

Table 5-3. Standard Event Status Bit Description (continued)

Mnemonic	Bit[1]	Value	Meaning
CME	5	32	*Command Error*. A syntax or semantic error has occurred or the electronic load received a < GET > within a program message. Errors in the range of -199 thru -l00 can set this bit.
PON	7	128	*Power On.* The electronic load has been turned on or off since the last time this register was read. This bit is always set when the electronic load is turned on.
[1] Bits 1 and 7 are not used by the electronic load.			

Operation Status

The Operation Status registers let you determine whether either of the operation conditions described in Table 5-4 presently exist on the electronic load.

The Operation Condition register represents the present status of the electronic load; the bits are set when the indicated condition is true.

The PTR/NTR filter determines what type of transition in the Operation Condition register will set the bit in the corresponding Operation Event register.

- Programming a bit in the PTR filter causes a 0-to-1 transition in the Operation Condition register to set the corresponding bit in the Operation Event register.
- Programming the NTR filter causes a 1-to-0 transition to set the bit.
- Programming both filters causes either transition to set the bit.

If the transition filters are not programmed, the bit is disabled.

Table 5-4. Operation Status Bit Description

Mnemonic	Bit[1]	Value	Meaning
CAL	0	1	*Calibrating*. A calibration calculation is in progress. (Refer to the O*perating Manual* for details of calibration commands.)
WTG	5	32	*Waiting for trigger.* At least one channel is waiting for a trigger to occur. Any **TRIG** command for any mode on any channel sets this bit. When a trigger is received, the bit is reset. **ABORt** also resets this bit.
[1] Bits 1-4, and 6-15 are not used by the electronic load.			

Note When the unit is turned on, the PTR filter is programmed on for the CAL bit, and the NTR filter is programmed on for the WTG bit.

The Operation Event register represents all of the filtered operation conditions that have occurred since the last time this register was read. Reading the register resets it to zero.

The Operation Enable register can be programmed to specify which operation event bits are logically-ORed to become Bit 7 (OPER bit) in the Status Byte register.

Note Refer to the *Electronic Load Operating Manual* for information about calibration.

Status Byte Register

The Status Byte register summarizes all of the status events from all status registers. Table 5-5 describes the status events that apply to the electronic load.

The Status Byte register can be read with a serial poll or ***STB?** query. When a serial poll is sent in response to a service request, Bit 6 of the Status Byte register will contain the RQS bit. The RQS bit is the only bit that is automatically cleared after a serial poll. The other bits in the Status Byte register (including the MSS bit) are unaffected by a serial poll.

When the Status Byte register is read with a ***STB?** query, Bit 6 of the Status Byte register will contain the MSS bit. The MSS bit indicates that the load has at least one reason for requesting service. It is the inclusive-OR of the enabled bits (excluding bit 6) of the Status Byte register. ***STB?** does not affect the status byte. The Status Byte register is cleared when a ***CLS** command clears all of the associated status registers.

Service Request Enable Register

The Service Request Enable register can be programmed to specify which bits in the Status Byte register will generate service requests. All bits except Bit 6 (RQS/MSS) can be enabled to generate service requests. In addition to generating a service request, the enabled bits in the Service Request Enable register are logically-ORed to become the MSS bit in the Status Byte register.

Note The present settings of the Service Request Enable register can be saved in non-volatile memory if ***PSC** is programmed to zero. The next time the unit is turned on, the Service Request Enable register will be programmed according to the saved settings.

Table 5-5. Status Byte Bit Description

Mnemonic	Bit[1]	Value	Meaning
CSUM	2	4	*Channel Summary*. Indicates if an enabled channel event has occurred. Affected by Channel Condition, Channel Event, and Channel Summary Event registers.
QUES	3	8	*Questionable*. Indicates if an enabled questionable event has occurred. Affected by Questionable Condition and Questionable Event registers.
MAV	4	16	*Message Available*. Indicates if the Output Queue contains data.
ESB	5	32	*Event Status Bit*. Indicates if an enabled standard event has occurred. Affected by Standard Event register.
RQS/MSS	6	64	During a serial poll, RQS *(Request Service))* is returned and cleared. For an ***STB?** query, MSS (*Master Summary Status*) is returned without being cleared.
OPER	7	128	*Operation*. Indicates if an enabled operation event has occurred. Affected by Operation Condition and Operation Event registers.

[1] Bits 0 and 1 are not used by the electronic load. They will be read back as zeroes.

Index

Index (continued)

Index (continued)

Index (continued)

Index (continued)

Agilent Sales and Support Offices

For more information about Agilent Technologies test and measurement products, applications, services, and for a current sales office listing, visit our web site: http://www.agilent.com/find/tmdir

You can also contact one of the following centers and ask for a test and measurement sales representative.

United States:
Agilent Technologies
Test and Measurement Call Center
P.O. Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

Canada:
Agilent Technologies Canada Inc.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

Europe:
Agilent Technologies
Test & Measurement European Marketing Organisation
P.O. Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

Japan:
Agilent Technologies Japan Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Latin America:
Agilent Technologies
Latin American Region Headquarters
5200 Blue Lagoon Drive, Suite #950
Miami, Florida 33126
U.S.A.
(tel) (305) 267 4245
(fax) (305) 267 4286

Australia/New Zealand:
Agilent Technologies Australia Pty Ltd
347 Burwood Highway
Forest Hill, Victoria 3131
(tel) 1-800 629 485 (Australia)
(fax) (61 3) 9272 0749
(tel) 0 800 738 378 (New Zealand)
(fax) (64 4) 802 6881

Asia Pacific:
Agilent Technologies
24/F, Cityplaza One, 1111 King’s Road,
Taikoo Shing, Hong Kong
tel: (852)-3197-7777
fax: (852)-2506-9284

Technical data is subject to change.

Manual Updates

The following updates have been made to this manual since the print revision indicated on the title page.

4/15/00

All references to HP have been changed to Agilent.
All references to HP-IB have been changed to GPIB.