EEK
EK

EK-101 and EK-102

Programmers
Quick
Reference

**Tektronix·**
COMMITTED TO EXCELLENCE

# Contents

# Setting Up the 11401/11402

You must perform the following steps to set up the 11401/11402 to send and receive remote commands:

1. Touch the UTILITY MENU key to the right of the crt/display area. The Utility Menu will appear at the bottom of the screen.

2. For GPIB, touch the **GPIB Parameters** label and set the parameters. The following list is an example of usable settings:

   > **Mode:** TalkListen
   > **Address:** between 0 and 30
   >  (1 is the factory default setting)
   > **Terminator:** EOI or EOI/LF
   > **Debug:** Off.

3. For RS-232-C, touch the **RS232C Parameters** label and set the parameters. The following list is an example of usable settings:

   > **Baud Rate:** the baud rate used by your controller
   > **Parity:** None
   > **EOL String:** CR/LF
   > **Echo:** Off
   > **Flagging:** Soft (DC1/DC3)
   > **Verbose:** Off
   > **Stop Bits:** 1
   > **Delay:** 0.0
   > **Debug:** Off

For either interface, selecting **Debug: ON** enables front-panel viewing of interface input operations.

# General Information

This Programmers Quick Reference Manual is based on the GPIB and RS-232-C section of the 11401/11402 User's Reference Manual. Refer to the main manual for full details of commands and syntax.

In this Quick Reference, "remote commands" and "ASCII ports" refers to both GPIB and RS-232-C unless otherwise noted.

The 11401/11402 command set includes both set and query commands. Set commands cause the instrument to change a setting or mode. Query commands return a setting, mode, waveform, measurement, or instrument status to the controller.

## How the Manual is Arranged

The 11401/11402 Programmers Quick Reference Manual contains the following sections:

**Interface Considerations** — summarizes basic GPIB information.

**Syntax** — describes the structure of set and query commands, explains how the syntax is shown in the command section, and lists the defined syntactical elements.

**Commands** — lists commands alphabetically within functional groups. (The functional groups are the same as those in the 11401/11402 User's Reference, and are presented alphabetically.) Each group is located by a header at the top of the page.

**Error and Warning Messages** — defines the status byte and lists error and warning messages numerically in tables.

**Index** — contains all command headers and links arranged alphabetically.

**Command Summaries** — two one-page foldout summaries of all commands and their syntax—one arranged by functional group and the other alphabetically.

# Interface Considerations

The 11401/11402 supports remote commands via both IEEE Standard 488-1978 (GPIB), and EIA Standard RS-232-C. The same commands are used with both interfaces unless otherwise noted. See the 11401/11402 User's Reference for detailed information about the interfaces.

## GPIB Addressing

Each device in a GPIB network has a unique primary address between 0 and 30. A device can be a Listener (receiving data) or a Talker (sending data), depending on its setup. The 11401/ 11402 can be both a Talker and Listener but not a Controller.

## GPIB Interface Functions

Table 1-1 shows the interface function subsets supported by the 11401/11402.

Table 1-1. GPIB Functions

| Function Name | Subset |
|---|---|
| Acceptor Handshake | AH1 |
| Controller | C0 |
| Device Clear | DC1 |
| Device Trigger | DT0 |
| Electrical Interface | E2 |
| Listener | L4 |
| Parallel Poll | PP0 |
| Remote/Local | RL0 |
| Service Request | SR1* |
| Source Handshake | SH1 |
| Talker | T6 |

* STByte? provides this function for RS-232-C.

## GPIB and RS-232-C Interface Message Implementation

Table 1-2 shows the interface messages implemented on the GPIB and RS-232-C ports.

**Table 1-2.
Interface Message Implementation**

| Message | Implemented GPIB? | Implemented RS-232-C? |
|---|---|---|
| DCL | Yes | Yes* |
| GET | No | No |
| GTL | No** | No** |
| LLO | No** | No** |
| PPC | No | No |
| PPU | No | No |
| SDC | Yes | No |
| SPD | Yes | No |
| SPE | Yes | No |
| UNL | Yes | No |
| UNT | Yes | No |
| Listen Addresses | Yes | No |
| Talk Addresses | Yes | No |

* Function provided by BREAK character
** Function provided by FPAnel command

## Talked With Nothing To Say (TWNTS)

If the GPIB port is made a talker while the 11401/11402 isn't currently processing a GPIB command and its input and output buffers are empty, it will return the "TWNTS" (Talked With Nothing To Say) message (0FFh < EOI >) at the GPIB port.

The RS-232-C interface does not have a "TWNTS" message. If an external device attempts to read data from the RS-232-C port while the 11401/11402 has no pending query responses or buffered query responses to process, the interface will "hang," preventing further input/output operations.

# Command Syntax

## Set Command Structure

Set commands are of two types: header-link-argument, or header-argument. Figures 1-1 and 1-2 show the structure of the two types of set command. A header is always followed by a space delimiter. A link and its argument are delimited by a colon. Argument-only commands have no links and consist of a header and argument separated by a space; the colon delimiter is omitted.

**Note:** To conserve space, most examples in this book show the short form (LONgform OFF) of the header, link(s), and argument(s); both the long and short forms are shown in the command descriptions. However, the power-on default is LONgform ON, which returns the full spellings of headers, links and arguments even if the short form was entered.

Header  Link  Argument

**TRM TIH:50E-6**

Delimiter      Delimiter
(space)        (colon)
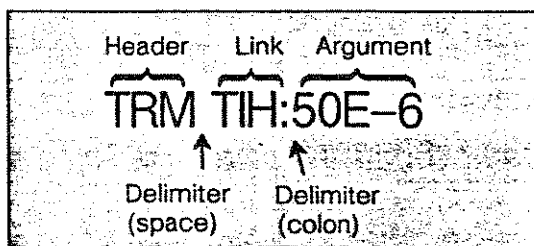
Figure 1-1.  Header-Link-Argument Set
Command Structure

Header  Argument

**MAINP -1E-5**

Delimiter
(space)
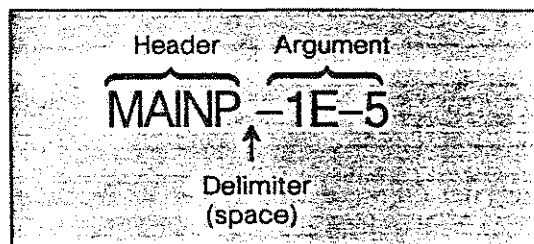
Figure 1-2.  Header-Argument Set
Command Structure

## Query Command Structure

Figure 1-3 shows the query command structure. Query commands are formed by adding a question mark to the header; in this manual, a header shown only with a question mark is query-only. (Headers that are set-only are noted in the text.) For most queries, links can be specified optionally and some links are query-only.

*Note:* For most set/query commands, only the set form is shown. The query form returns the current value(s) of the argument(s). Any special query information is given in a "query note" following the main command information.
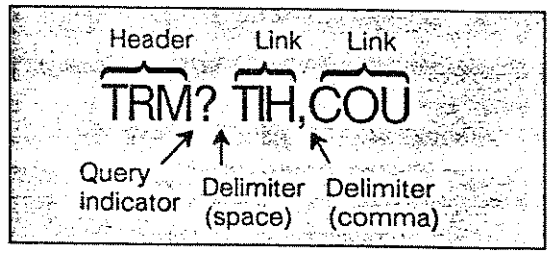


Figure 1-3. Query Command Structure

## Examples

Examples are shown in shaded boxes, like this:

**Examples of text you enter are in bold.**
Responses are shown in regular type.

Figure 1-4 shows two examples of a query response in this manual. The first example queries specific links of the command, and the second displays the parameters of all the links.

**TRM? TIH,COU <EOI>**
TRM TIH:25E-3,COU:DCN

**TRM? <EOI>**
TRM MOD:AUTOL,ALE:25,ANL:-2,VOL,
COU:DCN,SLO:PLU,SOU:"L1+L2",
STA:TRG,TIH:25E-3

Figure 1-4. Query Command Responses

*<EOI> Note:* The <EOI> symbol shown in examples represents the message terminator. The 11401/11402 requires a message terminator to process a command. (Most programs define the message terminator at the start of the program.) See the Syntax Definitions later in this section for a list of allowable <EOI> delimiters.

## Syntax Description

Figure 1-5 illustrates a syntax block in this manual. Headers are left-justified; the rest of the command information is indented. The full form of the header is given (with the minimum spelling in **CAPs**), followed by the required syntax. <Link> parameters are shown in the left-hand space of the open-sided box; <arg> parameters are shown in the center. The right-hand space summarizes the range of values for <arg>, if applicable. Examples (in the shaded boxes) are included for most headers and links.

---

**TRMain** <link>:<arg>

Establishes main time trigger values.

| TIHoldoff | <NRx> | 490e-9 to 10 secs |
|---|---|---|

Sets main trigger Time HOLDOFF in seconds.

**TRM TIH:25e-3** <EOI>

---

Figure 1-5. Command Syntax Block

Query-only links use a two-space syntax box showing the link and its response. Figure 1-6 shows a query-only link syntax block.

---

| STAtus | TRG | NOTrg |
|---|---|

**Query only.** Returns trigger status of main time base. TRG indicates main time base is triggered; NOTrg, that it is not.

**TRM? STA** <EOI>
TRM STA:TRG

---

Figure 1-6. Query-Only Link Syntax Block

## Formula Boxes

When a formula is needed to calculate a parameter value, the formula is enclosed in a box. (In the formula box only, commands and their links or arguments are shown for clarity connected with an underscore.) Figure 1-7 shows a formula box with explanatory text.

The duration of the main time base is calculated by:

TBM_XINcr * (TBM_LENgth -1)

Figure 1-7. Formula Box Example

# Syntax Definitions

The following BNF (Backus-Naur-Form) symbols are used in this manual:

| | |
|---|---|
| < > | Defined element |
| ::= | Is defined as |
| { } | Group — one element required |
| [ ] | Optional elements, may be omitted |
| \| | Exclusive or |
| ... | Previous element(s) may be repeated |

## Globally-Defined Syntax Elements

The following elements are globally defined:

| | | |
|---|---|---|
| <bblock> | ::= | Tektronix "Codes & Formats" binary data block. Used only by CURVe and SET commands |
| <EOI> | ::= | Message terminator selected in the front-panel Utility menu. For GPIB, <EOI> can be the IEEE-488 EOI interface signal or LF (linefeed). (The 11401/11402 always accepts the EOI interface signal as an input message terminator.) For RS-232C, the output terminator (EOL string) can be CR, LF, CR/LF, or LF/CR. (The 11401/11402 accepts any of these as an input message terminator.) |
| <slot> | ::= | {L\|C\|R}. Left, Center, or Right plug-in compartment |

### Numeric Arguments

The following numeric arguments are used in this manual:

$<ui>$      Unsigned integer value; no leading white space permitted

$<NR1>$      Signed integer value

$<NR2>$      Floating point value, no exponent

$<NR3>$      Floating point value with exponent

$<NRx>$      $\{<NR1>|<NR2>|<NR3>\}$ Range: 1e-300 to 1e+300, inclusive, to 15 significant digits

### Quoted Strings ($<qstring>$)

Quoted strings must follow these rules:

- Unless otherwise noted, maximum length is 127 characters, excluding delimiters.

- String delimiters are either an apostrophe (') or quotation mark ("). The same delimiter that opens the string must close the string.

- To include a delimiter within a string, double it, as in: "here is a "" mark".

- A $<qstring>$ may not be terminated with an IEEE-488 EOI interface signal, nor may it include an embedded ASCII NULL (0). However, carriage returns and linefeeds can be embedded within the $<qstring>$; they do not terminate the string.

### Command Processing Conventions

The following items are command processing conventions:

- Any combination of set and/or query commands can be concatenated together with a semicolon (;).

- The 11401/11402 is not case-sensitive; it accepts both upper- and lowercase input. It returns only UPPERCASE alphabetic data to its ASCII ports.

# Command Descriptions

This section describes the remote commands available through the GPIB and RS-232-C interfaces. For more detailed information, see the 11401/11402 User's Reference Manual.

## Acquisition Commands

The acquisition commands control waveform digitizing.

### AUTOSet [<link>:] <arg>

Enables or disables vertical and horizontal automatic ranging of input signals.

| HORiz | PERiod | OFF | |
|-------|--------|-----|---|

HORiz:PERiod enables AUTOSet on the horizontal component of the trace; HORiz:OFF disables AUTOSet on the horizontal component.

**AUTOS HOR:PER <EOI>**

| | STARt | |
|---|-------|---|

**Set Only.** Begins autosetting on the selected trace. If no trace is selected, the 11401/11402 samples all plug-in units and autosets the first signal it encounters.

**AUTOS STAR <EOI>**

| | UNDO | |
|---|------|---|

**Set Only.** Cancels a previous autoset and returns the selected trace to its original horizontal and vertical settings.

**AUTOS UNDO <EOI>**

# AUTOSet  [<link>:] <arg>          (cont.)

| VERt | ECL | PP | TTL | OFF | |
|------|-----|-----|-----|-----|---|

VERt:ECL autosets the vertical components referenced to ECL levels. VERt:PP autosets to peak-to-peak levels. VERt:TTL autosets to TTL levels. VERt:OFF disables autoset on the vertical components.

> AUTOS VER:PP <EOl>

# CONDacq <link>:<arg>

Sets the conditions of conditional waveform acquisition.

| FILl | <NRx> | 1 to 100 percent |
|------|-------|------------------|

Sets the percentage of waveform completion when CONDacq TYPe is FILl.

> COND FIL:60 <EOl>

| REMAining | <NR1> |
|-----------|-------|

Query Only. Returns a value indicating how much of the selected acquisition type must be acquired to complete acquisition.

| | |
|-----|-----|
| AVG | Averages remaining to completion |
| BOTh | Averages & Envelopes remaining |
| CONTInuous | Always 0; REMA is not meaningful |
| ENV | Envelopes remaining |
| FILl | Percent of fill remaining |
| SINgle | Always 0; REMA is not meaningful |

> COND? REMA <EOl>
> COND REMA:22

## CONDacq \<link\>:\<arg\>  (cont.)

| TYPe | AVG \| BOTh \| CONTInuous \| ENV \| FILI \| SINgle |
|---|---|

Selects the acquisition type, as follows:

| | |
|---|---|
| AVG | Acquires until NAVg number of averages, for all traces that include AVG in their descriptions. |
| BOTh | Acquires both AVG and ENV types. |
| CONTInuous | Acquires continually until halted. |
| ENV | Acquires until NENV number of envelopes, for all traces that include ENV in their descriptions. |
| FILI | Acquires to the percentage set by COND FIL. |
| SINgle | Acquires a single trigger from the main time base. |

**COND TYP:CONTI \<EOI\>**

## DIGitizer { RUN | STOP }

Starts and stops waveform digitization. At least one trace must be currently defined with at least one component that is being digitized.

| | RUN \| STOP | |
|---|---|---|

**DIG RUN \<EOI\>**

## Calibration Commands

The calibration commands initiate and report on the 11401/11402's self-calibration features.

## CALStatus?

**Query Only.** Returns the calibration (accuracy) status of the mainframe and plug-in units. Responses are: ENHanced | NENhanced | NEWconfig.

> CALS? <EOI>
> CALS ENH

## CCAlconstants <ui>:<NRx>

Sets or queries the center plug-in unit calibration constants. Can only be set when an internal jumper is enabled by qualified service personnel.

| <ui> | <NRx> | Range depends on unit |
|------|-------|------------------------|

where < ui > specifies the constant and < NRx > is the value of the constant.

> CCA 12:800 <EOI>

## LCAlconstants <ui>:<NRx>

Sets or queries the left plug-in unit calibration constants. Can only be set when an internal jumper is enabled by qualified service personnel.

| <ui> | <NRx> | Range depends on unit |
|------|-------|------------------------|

where < ui > specifies the constant and < NRx > is the value of the constant.

> LCA 12:800 <EOI>

## MCAlconstants <ui>:<NRx>

Sets or queries mainframe calibration constants. Can be set only when an internal jumper is enabled by qualified service personnel.

| <ui> | <NRx> | (See below for range) |
|------|-------|------------------------|

where <ui> specifies the constant and <NRx> is the value of the constant. Range of <ui> is 1 to "x", where "x" depends on the current firmware. Range of <NRx> is $-2^{31}$ to $2^{31}-1$.

**MCA 9:48 <EOl>**

## RCAlconstants <ui>:<NRx>

Sets or queries the right plug-in unit calibration constants. Can be set only when an internal jumper is enabled by qualified service personnel.

| <ui> | <NRx> | Range depends on unit |
|------|-------|------------------------|

where <ui> specifies the constant and <NRx> is the value of the constant.

**RCA 12:800 <EOl>**

## SELFcal [<link>:] <arg>

Forces calibration to occur or selects the mode of calibration.

| | FORce | |
|--|-------|--|

**Set Only.** Forces an immediate self-calibration.

**SELF FOR <EOl>**

| MODe | AUTO | MANual | |
|------|------|--------|--|

Selects whether self-calibration is performed automatically when due or performed manually using the FORce option.

**SELF MOD:MAN <EOl>**

## Channel Commands

The channel commands set and query the vertical amplifier plug-in units. The links that apply to all amplifiers are listed first, followed by the links specific to nondifferential amplifiers, then links specific to differential amplifiers, and finally the links that affect bandwidth.

### CH<slot><ui> <link>:<arg>

Sets vertical plug-in parameters.

The following three links affect **all amplifiers**:

| IMPedance | <NRx> | 50Ω, 1MΩ, 1GΩ |
|---|---|---|

Sets the channel input impedance in ohms. Out-of-range values are coerced to acceptable values; no warning message is returned.

    CHR1 IMP:1E6 <EOI>

| SENsitivity | <NRx> | 1e-3 to 10 volts* |
|---|---|---|
| | | 10e-3 to 1 volt** |

Sets the channel vertical gain.

* Range for 11A32, 11A33, 11A34, 11A52 amplifiers. See the plug-in unit user reference supplement for the resolution (step size) for different parts of the range.

** Range for 11A71 amplifier in 1-2-5 step intervals.

    CHL1 SEN:5E-1 <EOI>

| UNIts | <qstring> |
|---|---|

**Query Only.** Returns the units of the channel.

    CHC1? UNI <EOI>
    CHC1 UNI:"VOLTS"

# CH<slot><ui> <link>:<arg>     (cont.)

The following three links affect **nondifferential amplifiers** only:

| COUpling | AC \| DC \| OFF | |
|---|---|---|

Sets the input coupling.

### CHC2 COU:DC <EOI>

| OFFSet | <NRx> | (See below for range) |
|---|---|---|

Sets the voltage subtracted from the input signal. Vertically positions signal on the display.

Range for 11A32, 11A34, and 11A52 plug-in units:

| Sensitivity | Range |
|---|---|
| 1 mV to 99.5 mV | +/- 1 V |
| 100 mV to 995 mV | +/- 10 V |
| 1 V to 10 V | +/- 100 V |

Range for 11A71 plug-in unit:

| +/- 10 * CH_SEN |
|---|

### CHL2 OFFS:0.5 <EOI>

| PROBe | <qstring> |
|---|---|

**Query Only.** Returns probe type: "NONE", "Level 1", or "Level 2/<probe_type>/<serial_num>" currently connected to channel input.

### CHL1? PROB <EOI>
### CHL1 PROB:"LEVEL 1"

## CH<slot><ui> <link>:<arg>     (cont.)

The following 10 links affect **differential amplifiers** only:

| AMPoffset | <NRx> | |
|---|---|---|

AMPoffset is affected by the <arg> of the differential OFFSet link, depending on the current coupling and probes. See the plug-in unit's manual for more information.

**CHC1 AMP:1.0 <EOI>**

| MNSCoupling | AC \| DC \| VC \| OFF | |
|---|---|---|

Sets the channel minus differential input probe coupling. When OFF or VC, the input is disconnected from its external signal source. (See VCOffset.)

**CHR1 MNSC:AC <EOI>**

| MNSOffset | <NRx> | |
|---|---|---|

Sets the probe offset voltage subtracted from the minus input of the specified channel. Offset-type probe is required; the MNSOffset value is saved until an appropriate probe is connected.

**CHR1 MNSO:0 <EOI>**

| MNSProbe | <qstring> | |
|---|---|---|

**Query Only.** Returns the probe type: "Level 1", "Level 2/<probe_type>/<serial_num>" or "NONE" currently connected to minus input of the channel. See the nondifferential PROBe link for an example.

| OFFSet | <NRx> | |
|---|---|---|

The OFFSet <arg> modifies the links AMPoffset, MNSOffset, PLSOffset, or VCOffset, depending on current coupling and probes. See the plug-in supplements for more information.

**CHC1 OFFS:5E-1 <EOI>**

# CH<slot><ui> <link>:<arg>        (cont.)

The following links affect **differential amplifiers** only:

| PLSCoupling | AC | DC | VC | OFF |
|---|---|---|---|---|

Sets the channel plus differential input probe coupling. When OFF or VC, input is disconnected from its external signal source. (See VCOffset link in this subgroup.)

**CHR1 PLSC:VC <EOI>**

| PLSOffset | < NRx > | |
|---|---|---|

Sets the plus input probe offset voltage for the specified channel.

**CHR1 PLSO:3 <EOI>**

| PLSProbe | < qstring > | |
|---|---|---|

**Query Only.** Returns probe type: "NONE", "Level 1", or "Level 2/ < probe_type > / < serial_num >" currently connected to the plus input of the channel. See the nondifferential PROBe link for an example.

| PROTect | ON | OFF | |
|---|---|---|---|

PROTect:ON restricts minimum SENsitivity settings for differential amps. For the 11A33 with PROT:ON, SENsitivity range is restricted to 100 mV to 10 V and IMPedance is restricted to 50 ohms (active probe) or 1M ohm (passive probe).

**CHL3 PROT:ON <EOI>**

| VCOffset | < NRx > | |
|---|---|---|

Sets an internal comparison voltage when either the PLSCoupling or MNSCoupling link is set to VC (VCOffset has no effect otherwise).

**CHC1 VCO:4.5 <EOI>**

# CH<slot><ui> <link>:<arg>     (cont.)

The following three links affect the **bandwidth parameter**, which is determined by the plug-in unit and mainframe:

| BW | <NRx> | See below for ranges |
|---|---|---|

Sets the channel bandwidth. Out-of-range values are coerced to acceptable maximum or minimum values; no warning message is returned. Ranges are:

11A32 — 20e6, 100e6, 350e6/400e6*
11A33 — 20e6, 100e6, 150e6
11A34 — 20e6, 100e6, 300e6
11A52 — 20e6, 100e6, 500e6/600e6*
11A71 — 500e6/1000e6*

Note: * "/" values are 11401/11402 differences.

**CHL3 BW:100E6 <EOI>**

| BWHi | <NRx> | |
|---|---|---|

Sets a channel's high bandwidth. Only valid for plug-in units with BWHi function.

| BWLo | <NRx> | |
|---|---|---|

Sets a channel's low bandwidth. Only valid for plug-in units with BWLo function.

*CH < slot > < ui >? Note:* For the indicated channel, returns links (all or only those specified) with their current arguments. Differential and nondifferential amplifiers return the link values associated with that type of amp.

Query of nondifferential amplifier:

**CHC2?** <EOI>
CHC2 COU:DC,OFFS:2.5E+1,
BW:2.0E+7,IMP:1.0E+6,
PROB:"LEVEL 1",SEN:5.0E−1,
UNI:"VOLTS"

## CH<slot><ui> <link>:<arg>       (cont.)

Query of differential amplifier:

**CHR1?** <EOI>
CHR1 MNSC:AC,PLSC:VC,PROT:ON,
OFFS:5.0E−1,AMP:1.5E+0,BW:1.0E+8,
IMP:1.0E+6,MNSO:0.0E+0,
MNSP:"LEVEL 1",PLSO:3.0E+0,
PLSP:"LEVEL 1",SEN:2.0E+0,
UNI:"VOLTS",VCO:4.5E+1

*CH[<slot>]? Note:* CH<slot>? returns
the arguments of all links for each input
channel in the slot, in low-to-high channel
order. CH? returns the same information as
CH<slot>? for all 11000-Series plug-ins in
low-to-high channel order, in left-center-right
slot order.

**CHC?** <EOI>
CHC1 <link>:<arg>,<link>:<arg>...;
...
CHCn <link>:<arg>,<link>:<arg> ...

**CH?** <EOI>
CHL1 <link>:<arg>,<link>:<arg> ... ;
...
CHLn <link>:<arg>,<link>:<arg> ... ;
CHC1 <link>:<arg>,<link>:<arg> ... ;
...
CHCn <link>:<arg>,<link>:<arg> ... ;
CHR1 <link>:<arg>,<link>:<arg> ... ;
...
CHRn <link>:<arg>,<link>:<arg> ...

## Cursor Commands

The cursor commands create and place cursors on waveforms.

## CURSor <link>:<arg>

Selects cursor operating characteristics. Query returns all links (or only those specified) and their current arguments.

| REAdout | ON | OFF | |
|---------|---------|---|

Enables or disables the front-panel cursors and their corresponding knob readouts. Cursors can be manipulated with remote commands regardless of the state of REAdout. However, with REAdout OFF, the display is not updated to show cursor changes made remotely.

**CURS REA:ON <EOI>**

| REFErence | TRAce <ui> | 1 to 8 |
|-----------|------------|--------|

Selects the reference trace for SPLit cursors.

**CURS REFE:TRA2 <EOI>**

| TYPe | PAIred | SPLit | VBArs | HBArs |
|------|--------|-------|-------|-------|

Selects the cursor type. SPLit cursors are not permitted on XY traces.

**CURS TYP:PAI <EOI>**

| XUNit | AMPS | DIVS | OHMs | SEConds | VOLts | WATs |
|-------|----------------------------------------|

Query Only. Returns the horizontal units of the selected trace.

**CURS? XUN <EOI>**
**CURS XUN:SEC**

## CURSor <link>:<arg>                    (cont.)

| YUNit | AMPS \| DIVS \| OHMs \| VOLts \| WATs |
|-------|---------------------------------------|

**Query Only.** Returns the vertical units of the selected trace.

**CURS? YUN <EOI>**
CURS YUN:VOL

## DOT1Abs <link>:<arg>;
## DOT2Abs <link>:<arg>

Sets or queries absolute positions (with respect to the waveform record) for split or paired (dot) cursors. DOT1A and DOT2A have the same parameters. For YT traces, PCT, XCO, or XDI may be used to position the cursors. For XY traces, only PCT may be used.

| PCTg | <NRx> | 0 to 100 percent |
|------|-------|------------------|

Positions the first or second dot cursor as a percentage of the waveform record.

**DOT2A PCT:80  <EOI>**

| XCOord | <NRx> | See below for range |
|--------|-------|---------------------|

Positions the first or second dot cursor with respect to horizontal units of the selected trace.

Range when the selected trace's record is MAIN:

| MAINP to ( MAINP + duration ) |
|-------------------------------|

Range when the selected trace's record is WIN1:

| WIN1P to ( WIN1P + duration ) |
|-------------------------------|

Range when the selected trace's record is WIN2:

| WIN2P to ( WIN2P + duration ) |
|-------------------------------|

(See the Time Base commands for duration.)

**DOT1A XCO:150E-6  <EOI>**

Cursor Commands

# DOT1Abs; DOT2Abs <link>:<arg>

(cont.)

| XDIv | <NRx> | −5.12 to + 3.07* |
| --- | --- | --- |
| | | −5.12 to + 5.10** |

Ranges shown are valid only when PANzoom is OFF and the selected trace is acquired.

\* Maximum XDIv of + 3.07 for record lengths of 4096 or 8192.

\*\* Maximum XDIv of + 5.10 for any record length except 4096 or 8192.

**DOT2A XDI:−4.65  <EOI>**

| XQUal | EQ \| LT \| GT \| UN |
| --- | --- |

**Query Only.** Indicates the accuracy of XCO or XDI positioning information. EQ means the position and the response are equal. LT means the position is lower than the response (i.e., the cursor is offscreen low). GT means the position is greater than the response (i.e., the cursor is offscreen high). UN means the position is uncertain (i.e., on an unacquired waveform point.)

**DOT2A? XQU  <EOI>**
DOT2A  XQU:EQ

| YCOord | <NR3> |
| --- | --- |

**Query Only.** Returns the vertical position of the first or second dot cursor, in units of the selected trace.

**DOT1A? YCO <EOI>**
DOT1A  YCO:5e−6

| YDIv | <NR3> |
| --- | --- |

**Query Only.** Returns the vertical position of the first or second dot cursor in graticule divisions.

**DOT2A? YDI  <EOI>**
DOT2A  YDI:1.0

# DOT1Abs; DOT2Abs <link>:<arg>

(cont.)

| YQUal | EQ \| LT \| GT \| UN |
| --- | --- |

**Query Only.** Indicates the accuracy of YCO or YDI positioning information. EQ means the position and the response are equal. LT means the position is lower than the response (i.e., the cursor is offscreen low). GT means the position is greater than the response (i.e., the cursor is offscreen high). UN means the position is uncertain (i.e., on an unacquired waveform point.)

> DOT1A? YQU  <EOI>
> DOT1A YQU:EQ

# DOT1Rel <link>:<arg>
# DOT2Rel <link>:<arg>

**Set Only.** Sets the paired or split (dot) cursor position offset (relative) to the current cursor (absolute) location. DOT1R and DOT2R have the same parameters.

| PCTg | <NRx> | |
| --- | --- | --- |

**Set Only.** Positions the first or second dot cursor as a percentage of the waveform record.

> DOT1R PCT:20  <EOI>

| XCOord | <NRx> | |
| --- | --- | --- |

**Set Only.** Positions the first or second dot cursor with respect to the units of the selected trace.

> DOT2R XCO:3.1e-3  <EOI>

| XDIv | <NRx> | |
| --- | --- | --- |

**Set Only.** Positions the first or second dot cursor in graticule divisions.

> DOT1R XDI:2.25  <EOI>

Cursor Commands

# H1Bar <link>:<arg>;
# H2Bar <link>:<arg>

Sets or queries the absolute horizontal bar cursor position. H1Bar and H2Bar have the same parameters.

| YCOord | <NRx> | See below for range |
|--------|-------|---------------------|

Positions the first or second horizontal bar cursor with respect to the units of the selected trace.

Range for the selected trace created in integer mode (TRAce WFMCalc:FASt):

| (-5.12 * CH_SEN + CH_OFFS) to |
| (5.10 * CH_SEN + CH_OFFS) |

Range for the selected trace created in floating-point mode (TRAce WFMCalc:HIPrec):

| (-5.12 * ADJ_VSI + ADJ_VPO) to |
| (5.10 * ADJ_VSI + ADJ_VPO) |

**H2B YCO:−3.25 <EOI>**

| YDIv | <NRx> | −5.12 to + 5.10 |
|------|-------|-----------------|

Positions the first or second horizontal bar cursor in graticule divisions.

**H2B YDI:1.5 <EOI>**

## V1Bar <link>:<arg>;
## V2Bar <link>:<arg>

Sets or queries the absolute vertical bar cursor position. V1Bar and V2Bar have the same parameters.

| XCOord | <NRx> | See below for range |
|--------|-------|---------------------|

Positions the first or second vertical bar cursor with respect to the units of the selected trace, with PANzoom OFF.

Range when the selected trace's reference record is MAIN:

| MAINP to (10.22 * TBM_TIMe + MAINP) |
|-------------------------------------|

Range when the selected trace's reference record is WIN1:

| WIN1P to (10.22 * TBW_TIMe + WIN1P) |
|-------------------------------------|

Range when the selected trace's reference record is WIN2:

| WIN2P to (10.22 * TBW_TIMe + WIN2P) |
|-------------------------------------|

V1B XCO:-1E-6 <EOI>

| XDIv | <NRx> | -5.12 to +5.10 |
|------|-------|----------------|

Positions the first or second vertical bar cursor in graticule divisions.

V1B XDI:-4.15 <EOI>

## Data Transfer Commands

The data transfer commands transfer wave-
forms and settings between the 11401/11402
and its external interfaces.

### ABBwfmpre { ON | OFF }

Enables or disables abbreviation of the
response to a WFMpre? query. When ABB is
ON, the WFM? response is:

> WFM? <EOI>
> WFM ACS:ENH,NR.:2048,PT.:Y,
> XIN:5.0E-7,XMU:5.0E-1,XZE:0.0,
> YMU:1.5625E-4,YZE:0.0

When ABB is OFF, the response to WFM?
includes all 17 links of the WFMpre command.
The power-up default is ABB OFF.

> **ABB ON** <EOI>

### BYT.or { LSB | MSB }

Determines whether the least significant byte
(LSB) or most significant byte (MSB) of
waveform data is transmitted first during a
binary (<bblock>) CURVe transfer. Power-
up default is BYT MSB; however, BYT LSB has
a faster data transfer rate.

> **BYT LSB** <EOI>

Data Transfer Commands

# CURVe {<bblock>|(<NRx>[,<NRx>]...)}

Sends or accepts unscaled waveform data in binary or ASCII format via the interfaces. The WFMpre command defines the waveform characteristics (for example, NR.pt is the number of data points transmitted).

The set form sends data to the 11401/11402 from the controller. The storage location for the data is specified by the INPut command.

The query form retrieves data from the 11401/11402. The waveform source is specified by the OUTput command.

| <bblock> | %<byte_cnt><bin_pt>...<checksum> |
|---|---|

**Binary Transfer.** <Byte_cnt> is a 16-bit integer (MSB) giving the length in bytes of the remainder of the binary block, including checksum; <bin_pt> is a binary data point in the range –32768 to +32767 (transmission order is set by the BYT.or command); and <checksum> is an 8-bit, 2's complement of the modulo 256 sum of <byte_cnt> and all <bin_pt> data.

**CURV %**(see <bblock> above)

| ( <NRx> [ ,<NRx> ] ... ) | –32768 to +32767 |
|---|---|

**ASCII Transfer.** <NRx> values are data points within the range.

**CURV 16392,16464,...,–9248<EOI>**

*CURVe? Data Values:* The following data point values are predefined for CURVe?:

| | |
|---|---|
| +32767 | Vertical Overrange. Data points are high off-screen and are not displayed. |
| –32767 | Vertical Underrange. Data points are low off-screen and not displayed. |
| –32768 | Null Data. Data points that have not been acquired. |

11401/11402 Quick Reference          2–19

Data Transfer Commands

# CURVe {<bblock>|(<NRx>[,<NRx>]...)}

(cont.)

**Waveform Scaling:** The following formulas are used to scale the coordinate values of each waveform point for YT waveforms:

```
X(n) = WFM_XZE + WFM_XIN * n
Y(n) = WFM_YZE + WFM_YMU * data_pt_n
```

where X(n) is the scaled horizontal coordinate of the nth data point in XUNits; Y(n) is the scaled vertical coordinate of the nth data point in YUNits; "n" is the sequence number of the nth retrieved data point (range is 0 to WFM_NR.pt−1); and data_pt_n is the value of the nth unscaled point (as retrieved by CURVe?).

The following formulas are used to scale the coordinate values of each X,Y point-pair for XY waveforms:

```
X(n) = WFM_XZE + WFM_XMU * data_pt_nx
Y(n) = WFM_YZE + WFM_YMU * data_pt_ny
```

where X(n) is the scaled X-coordinate of the nth unscaled X,Y pair in XUNits; Y(n) is the scaled Y-coordinate of the nth unscaled X,Y pair in YUNits; data_pt_nx is the value of the nth unscaled X-coordinate (as retrieved by CURVe?); and data_pt_ny is the value of the nth unscaled Y-coordinate.

**Sending "Preamble-less" Waveforms:** If a stored waveform exists at the INPut STO location, it is overwritten and its preamble is used with the new waveform. If no stored waveform exists at the INPut STO location, the following default preamble is used with the new waveform:

| | |
|---|---|
| ACState | ENHanced |
| NR.pt | 1024 |
| PT.FMT | Y |
| XINcr | 5.0e−7 |
| XZEro | 0.0 |
| YMUlt | 1.5625e−4 |
| YUNit | VOLts |
| YZEro | 0.0 |

These are the power-up defaults. If any of these links have been modified, the modified values are used.

## ENCdg <link>:<arg>

Selects the data encoding returned by CURVe?, WAVfrm?, and SET? queries.

| SET | ASCii \| BINary | |
|-----|----------------|--|

Sets the encoding for front-panel setting (FPS) transfers.

**ENC SET:ASC** <EOI>

| WAVfrm | ASCii \| BINary | |
|--------|----------------|--|

Sets the encoding for waveform transfers.

**ENC WAV:BIN** <EOI>

## INPut STO<ui>

Selects the destination for preamble and waveform data input by WFMpre and CURVe.

| STO < ui > | 1 to 256 |
|------------|----------|

The power-up default INPut location is STO1.

**INP STO36** <EOI>

## OUTput { STO<ui> | TRAce<ui> }

Selects the source for the preamble and waveform data output by the WFMpre?, CURVe?, or WAV? queries.

| STO < ui > | 1 to 256 |
|------------|----------|
| TRAce < ui > | 1 to 8 |

STO < ui > is a stored waveform; TRAce < ui > is a displayed waveform. The power-up default is STO1.

**OUT TRA8** <EOI>
**OUT STO6** <EOI>

Data Transfer Commands

## SET <bblock>

Sends a front-panel setting (FPS) to the 11401/11402 from the controller. The FPS was acquired by an earlier SET? command, and must not be modified by the controller. Also, the FPS cannot be sent to another scope having a different firmware version.

| <bblock> | % <byte_cnt> <settings> ... <checksum> |
|----------|------------------------------------------|

**Set Only.** <Byte_cnt> is a 16-bit integer (MSB) giving the length in bytes of the remainder of the binary block, including checksum; <settings> are binary-encoded data; and <checksum> is an 8-bit, 2's complement of the modulo 256 sum of <byte_cnt> and all <settings> data.

**SET %** *(see <bblock> above)*

## SET?

Returns the front-panel settings (FPS) as either ASCII strings or in binary <bblock> format, depending on the ENC SET argument.

**SET?** <EOI>
*{<string>[,<string>...]}*

where <string> represents an ASCII string. *Note:* The ASCII string response does *not* return the header "SET" before the data. The binary response does return the header "SET" before the data.

## WAVfrm?

**Query Only.** The WAV? command is equivalent to entering WFM?;CURV?.

**WAV?** <EOI>
WFM <link>:<arg> ...;CURV <arg>

## WFMpre <link>:<arg>

Either sends the waveform preamble for the waveform last selected by OUTput, or accepts a waveform preamble for the waveform last selected by INPut.

# WFMpre &lt;link&gt;:&lt;arg&gt; (cont.)

| ACState | ENHanced \| NENHanced |
|---------|----------------------|

Specifies whether the OUTput waveform was created with enhanced or non-enhanced calibration accuracy.

WFM ACS:ENH &lt;EOI&gt;

| BIT/nr | 16 |
|--------|-----|

Query Only. Returns the number of bits per binary waveform point, which is always 16.

WFM? BIT &lt;EOI&gt;
WFM BIT:16

| BN.fmt | RI |
|--------|-----|

Query Only. Returns the TEK Codes and Formats binary number format, which is always RI — right-justified, two's complement integers.

WFM? BN. &lt;EOI&gt;
WFM BN.:RI

| BYT/nr | 2 |
|--------|-----|

Query Only. Returns the binary data field width, which is always two bytes per binary waveform point.

WFM? BYT/ &lt;EOI&gt;
WFM BYT/:2

| BYT.or | LSB \| MSB |
|--------|-----------|

Query Only. Returns the binary data transmission order.

WFM? BYT &lt;EOI&gt;
WFM BYT:MSB

# WFMpre <link>:<arg>                    (cont.)

| CRVchk | CHKsm0 \| NONe \| NULl |
|--------|------------------------|

Query Only. Returns the type of checksum appended to waveform data returned via CURVe? CHKsm0 is a standard TEK Codes and Formats checksum, and is returned when ENCdg WAVfrm is BINary and OUTput is STO<ui>. NONe is no checksum and is returned when ENCdg WAVfrm is ASCii. NULl is an arbitrary zero checksum and is returned when ENCdg WAVfrm is BINary and OUTput is TRAce<ui>.

**WFM? CRV <EOI>**
WFM CRV:NUL

| ENCdg | ASCii \| BINary |
|-------|-----------------|

Query Only. Returns the encoding set with the ENCdg command.

**WFM? ENC <EOI>**
WFM ENC:BIN

| NR.pt | <NRx> | 512\|1024\|2048\|4096 5120 \| 8192 \| 10240 |
|-------|-------|------------------------------------------|

Specifies the number of points in the waveform record to be transferred.

**WFM NR.:4096 <EOI>**

| PT.fmt | ENV \| Y \| XY | |
|--------|---------------|---|

Selects the point format of the waveform data. "ENV" applies to YT waveforms that are transmitted as maximum-minimum point-pairs, with the maximum point transmitted first. "Y" indicates a YT waveform, with one data point returned for each waveform point. "XY" is an XY waveform, with an X,Y point-pair returned for each waveform point.

**WFM? PT.:Y <EOI>**

## WFMpre \<link\>:\<arg\> (cont.)

| WFId | STO \<ui\> \| TRAce \<ui\> |
|---|---|

**Query Only.** Returns the waveform ID, which is the source selected with the OUTput command.

> WFM? WFI \<EOI\>
> WFM WFI:STO36

| XINcr | \<NRx\> | $\geq$ 10e–12 secs / pt |
|---|---|---|

Specifies the horizontal sample interval of YT waveforms. Range begins at 10 ps per point.

> WFM XIN:20e–9 \<EOI\>

| XMUlt | \<NR3\> |
|---|---|

**Query Only.** Returns the vertical scale factor, in XUNits per division, of the horizontal component of an XY waveform.

> WFM? XMU \<EOI\>
> WFM XMU:1.5625E–4

| XUNit | AMPS \| DIVS \| OHMs \| SEConds \| VOLts \| WATs |
|---|---|

**Query Only.** Returns the horizontal units of the specified waveform at the time of its creation. For XY waveforms, this will be the vertical units of the horizontal component.

> WFM? XUN \<EOI\>
> WFM XUN:OHM

| XZEro | \<NRx\> | 1e–15 to 1e15 |
|---|---|---|

For YT waveforms, XZEro specifies the number of seconds of pre- or post-trigger. For XY waveforms, XZEro specifies the vertical offset of the horizontal component.

> WFM XZE:1.0 \<EOI\>

## WFMpre \<link>:\<arg>                    (cont.)

| YMUlt | \<NRx> | 1e-15 to 1e15 |
|-------|--------|---------------|

For YT waveforms, YMUlt specifies the vertical scale factor, in YUNits per division. For XY waveforms, YMUlt specifies the vertical scale factor, in YUNits per division, of the vertical component.

**WFM YMU:1.5625E-4** \<EOl>

| YUNit | AMPS \| DIVS \| OHMs \| VOLts \| WATs | |
|-------|--------------------------------------|--|

Specifies the vertical units of the waveform data at the time of waveform creation.

**WFM YUN:VOL** \<EOl>

| YZEro | \<NRx> | 1e-15 to 1e15 |
|-------|--------|---------------|

For YT waveforms, YZEro specifies the vertical offset of the waveform. For XY waveforms, YZEro specifies the vertical offset of the vertical component.

**WFM YZE:250E-3** \<EOl>

**WFM? Note:** WFM? returns links (all or only those specified) with their current arguments. (See also the ABBwfmpre command.)

```
WFM? <EOl>
WFM ACS:ENH,BIT:16,BN.:RI,BYT/:2,
BYT:LSB,CRV:CHK,ENC:BIN,NR.:1024,
PT.:Y,WFI:STO12,XIN:5.0E-7,
XMU:5.0E-1,XUN:SEC,XZE:0.0,
YMU:1.5625E-4,YUN:VOL,YZE:0.0
```

## Diagnostic Commands

The diagnostic commands invoke self-test or extended-test diagnostics.

## DIAg?

**Query Only.** Returns pass/fail information from self-test or extended-test diagnostics.

| PASsed | NONE | <omitted> |
|--------|------|-----------|

"NONE" means no tests were omitted. <Omitted> is a comma-delimited list of tests that were omitted.

    **DIAG?** <EOI>
    DIAG PAS:"NONE"

| FAIled | <failed_test> | <omitted> |
|--------|---------------|-----------|

See the 11401/11402 Service Reference Manual for the <failed_test> definitions. <Omitted> is a comma-delimited list of tests that were omitted. The DIAG? FAIled response can include both failed and omitted tests.

    **DIAG?** <EOI>
    DIAG FAI:"DI62X,DI22X,R????"

In the preceding example, "DI62X" and "DI22X" are failed tests and "R????" is an omitted test.

## TESt [ XTND ]

**Set Only.** Invokes either a self-test (TESt) or extended-test (TESt XTND) diagnostic sequence.

*Note:* TESt destroys all stored waveforms and user-defined expansion strings, resets the TEXt X:, Y: coordinates to 0,0, and removes user-text from the display.

    **TES XTND** <EOI>

## Display Commands

The display commands select the mode, number of graticules, and intensity of the display, and place text on the screen.

## DISPlay <link>:<arg>

Sets the display mode, number of graticules, and intensity.

| MODe | DOTs \| VECtors | |
|------|-----------------|---|

Enables either dot or vector type display.

**DISP MOD:VEC <EOI>**

| GRAticule | DUAl \| SINgle | |
|-----------|----------------|---|

Enables dual or single display graticules.

**DISP GRA:DUA <EOI>**

| INTensity | <NRx> | 0 to 100 percent |
|-----------|-------|------------------|

Sets the display intensity as a percentage value.

**DISP INT:65 <EOI>**

## TEXt [<link>:] <arg>

**Set Only.** Writes desired text to the specified area of the screen.

| | CLEar | |
|---|-------|---|

**Set Only.** Removes all user-defined text from the screen.

**TEX CLE <EOI>**

# TEXt [<link>:] <arg>          (cont.)

| STRing | < qstring > | |
|--------|-------------|--|

**Set Only.** < qstring > is the text to be displayed on the graticule, at the X: and Y: coordinates.

**TEX STR:'Touch here'** <EOI>

| X | <NRx> | 0 to 49 |
|---|-------|---------|

**Set Only.** The X coordinate of the text. 0 is the left edge of the graticule, 49 the right edge.

**TEX X:8** <EOI>

| Y | <NRx> | 0 to 31 |
|---|-------|---------|

**Set Only.** The Y coordinate of the text. 0 is the top edge of the graticule, 31 the bottom edge.

**TEX X:10,Y:7,STR:'ENTER'** <EOI>

In the above example, commas delimit the three links of the TEXt command.

## External I/O Commands

The external I/O commands manipulate the 11401/11402's GPIB, RS-232-C, and Centronics I/O ports.

## COPy  [<link>:] <arg>

Copies the front-panel display to the printer (Centronics) port.

|  | ABOrt |  |
|---|---|---|

**Set Only.** Aborts the current copy operation and clears the queue of copy requests.

COP ABO <EOI>

| FORMat | DRAft | HIRes | REDuced |
|---|---|

Selects the output format. DRAft and HIRes are approximately 8.5" by 11"; HIRes shows front-panel intensified regions; DRAft does not. REDuced format is a smaller size version of DRAft.

COP FORM:DRA <EOI>

| PRInter | PIN8 | PIN24 |  |
|---|---|---|

Selects the target printer type. PIN8 supports the standard Epson command set. PIN24 supports the extended Epson command set.

COP PRI:PIN8 <EOI>

|  | STARt |  |
|---|---|---|

**Set Only.** Initiates the front-panel copy. Entering "COPy" with no links or arguments is the same as entering "COPy STARt."

COP STAR <EOI>

## DEBug <link>:<arg>

Copies input data from the selected interface to the front-panel display for program development troubleshooting. The incoming ASCII commands are displayed on the top four lines of the screen.

| GPIb | ON | OFF | |
|------|---------|---|

Enables or disables the debug function for GPIB.

**DEB GPI:ON <EOI>**

| RS232 | ON | OFF | |
|-------|---------|---|

Enables or disables the debug function for RS-232-C.

**DEB RS232:OFF <EOI>**

## RS232 <link>:<arg>

**RS-232-C Only.** Sets or queries the RS-232-C parameters.

| BAUd | < NRx > | 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 |
|------|---------|--------------------------------------------------|

Sets the baud rate. Always set the baud rate on the 11401/11402 before setting the baud rate on the controller.

**RS232 BAU:9600 <EOI>**

| DELAy | < NRx > | 0 to 60 seconds |
|-------|---------|-----------------|

Sets the minimum delay from receipt of an RS-232-C query to its response. Range is 0 to 60 seconds in 20 millisecond (20e-3) intervals.

**RS232 DELA:80E-3 <EOI>**

| ECHo | ON | OFF | |
|------|---------|---|

Enables or disables character echo. Echo must be OFF for binary transfers.

**RS232 ECH:OFF <EOI>**

External I/O Commands

# RS232 <link>:<arg>                    (cont.)

| EOL | CR \| CRLf \| LF \| LFCr |
|-----|-------------------------|

Selects the End-Of-Line string (output message delimiter): carriage return (CR), line feed (LF), and combinations of the two.

**RS232 EOL:CRL <EOI>**

| FLAgging | SOFt \| HARd \| OFF |
|----------|---------------------|

Controls the input and output flagging. FLA:SOFt is XON/ XOFF control. FLA:HARd uses DTR and CTS lines. FLA:OFF ignores I/O control flags.

**RS232 FLA:SOF <EOI>**

| PARity | ODD \| EVEN \| NONe |
|--------|---------------------|

Selects a parity type (or NONe) for data transfers.

**RS232 PAR:NON <EOI>**

| STOPBits | <NRx> | 1, 1.5, 2 |
|----------|-------|-----------|

Selects the number of stop bits sent to identify the end of data.

**RS232 STOPB:2 <EOI>**

| VERBose | ON \| OFF |
|---------|-----------|

With VERB:ON, the 11401/11402 returns error and warning messages when they occur. With VERB:OFF, the controller must query the 11401/ 11402 for error and warning messages.

**RS232 VERB:ON <EOI>**

## Measurement Commands

The measurement commands execute and query measurements, and set references.

The following measurements (<meas>) can be performed with remote commands:

| | | | |
|---|---|---|---|
| CROss | DELAy | FALltime | FREq |
| MAX | MEAN | MID | MIN |
| PDElay | PERiod | PP | RISetime |
| RMS | TTRig | WIDth | YTEnergy |
| YTMns_area | YTPls_area | | |

### BASeline <NRx>

Sets the vertical baseline for measurements when MTRack is OFF. Range is any valid <NRx>.

| | <NRx> | 1e-300 to 1e300 |
|---|---|---|

**BAS -150E-3 <EOI>**

### COMpare { ON | OFF }

Enables or disables comparison between a measurement and a reference from the REFSet command.

**COM OFF <EOI>**

### DAInt { WHOle | SINgle }

Sets the data interval for measurements. Affects the MEAN, RMS, YTEnergy, YTMns_area, and YTPls_area <meas>.

| | WHOle | SINgle | |
|---|---|---|---|

WHOle sets the data interval to the whole measurement interval. SINgle sets the data interval as a single period.

**DAI WHO <EOI>**

Measurement Commands

## DISTal <NRx>

Sets the distal (far from point of origin) reference level for the RISetime and FALltime <meas>.

| | <NRx> | 0 to 100 percent |
|---|---|---|

**DIST 80 <EOI>**

## DLYtrace TRAce<ui>

Selects a delayed trace for the PDElay <meas>. The delayed trace cannot be the current selected trace.

| | TRAce<ui> | 0 to 8 |
|---|---|---|

If no delayed trace is selected, DLY? returns DLY TRA0. Setting DLY TRA0 is ignored (no error).

**DLY TRA3 <EOI>**

## LMZone <NRx>

Sets the left measurement zone as a percentage of the waveform record.

| | <NRx> | 0 to 100 percent |
|---|---|---|

**LMZ 30 <EOI>**

# MEAS?

**Query Only.** Executes the <meas>(s) in the current measurement list (MSLIst) and returns the <meas> value(s) plus qualifier(s). MEAS? returns "EMPTY" if no <meas> were in MSLIst.

Qualifiers indicate how accurately the value returned matches the true measurement. EQ means the true measurement and the response are equal. LT means that the measurement is lower than the response (i.e. offscreen low). GT means that the measurement is greater than the response (i.e., offscreen high). UN means that the true measurement is uncertain (i.e., on an unacquired portion of the waveform). ER indicates an error occurred during measurement.

    **MEAS?** <EOl>
    MEAS PP:4.0E-3,EQ,FRE:5.0E+6,EQ

# <meas>?

**Query Only.** Executes the specified <meas> and returns the <meas> value plus qualifier. (See MEAS? for qualifiers.)

    **WID?** <EOl>
    WID 4.2E-7,UN

# MESial <NRx>

Sets the mesial (middle) reference level of the waveform (i.e., the vertical endpoint of the waveform period) for DELAy, FREq, MEAN, PERiod, RMS and WIDth; and when DAInt is SINgle, for YTEnergy, YTMns_area, and YTPls_area.

| | <NRx> | 0 to 100 percent |
|---|---|---|

    **MES 50** <EOl>

Measurement Commands

# MSLIst { <meas> }

Generates a measurement list of up to six <meas> which the 11401/11402 executes continually while in Measurement major menu mode. MSLI? returns the <meas> in the current list, or "EMPTY" if there are none.

MSLI PP,MAX,RIS,FAL,FRE <EOI>

# MSLOpe { PLUs | MINUs }

Sets the crossing slope for the CROss <meas>.

MSLO PLU <EOI>

# MSNum?

**Query Only.** Returns the number of items in the current MSLIst. Range is 0 to 6.

MSN? <EOI>
MSN 5

# MSYs { ON | OFF }

Activates or deactivates the Measurement major menu in the front-panel display. Use MSYs ON when you are using the front panel in conjunction with remote commands. Use MSYs OFF for fastest GPIB throughput.

MSY ON <EOI>

# MTRack { ON | OFF }

Enables or disables measurement tracking (continuous building of histograms). When MTRack is ON, the 11401/11402 sets the topline and baseline; when MTRack is OFF, the user-defined TOPline and BASeline are used.

MTR ON <EOI>

## PROXimal <NRx>

Sets the proximal (near to point of origin) reference level for the RISetime and FALltime <meas>.

| <NRx> | 0 to 100 percent |
|---|---|

**PROX 20** <EOI>

## REFLevel <NRx>

Sets the signal reference level for CROss, YTEnergy, YTMns_area, and YTPls_area <meas>. Range is any valid <NRx>.

| <NRx> | 1e-300 to 1e300 |
|---|---|

**REFL 100.0** <EOI>

## REFSet <link>:<arg>

Sets the reference value for the <meas> used in comparison mode.

| CURRent | <meas> | |
|---|---|---|

**Set Only.** Executes the <meas> and stores its value as the reference.

**REFS CURR:PP** <EOI>

| <meas> | <NRx> | -5.0e20 to 5.0e20 |
|---|---|---|

Sets or queries the value of <meas>.

**REFS PP:4.7** <EOI>

## RMZone <NRx>

Sets the right measurement zone as a percentage of the waveform record.

| | <NRx> | 0 to 100 percent |
|---|---|---|

**RMZ 70** <EOI>

## SNRatio <NRx>

Sets the signal-to-noise ratio.

| | <NRx> | 1 to 99 |
|---|---|---|

**SNR 30** <EOI>

## TOPline <NRx>

Sets the top vertical level of the selected waveform when MTRack is OFF. Range is any valid <NRx>.

| | <NRx> | 1e-300 to 1e300 |
|---|---|---|

**TOP 150E-3** <EOI>

## TTAverage <NRx>

Sets the number of averages for the TTRig <meas>. Applies to all traces.

| | <NRx> | 1, 10, 100, 1000 |
|---|---|---|

**TTA 100** <EOI>

## *Miscellaneous Commands*

The miscellaneous commands perform a variety of useful functions.

## ABStouch { CLEar | <NRx>,<NRx> }

Activates a location on the 11401/11402 front panel by giving its X,Y coordinates. ABStouch always works, regardless of front-panel lockout (FPA OFF) or the Touch Panel disable button.

| | CLEar | |
|---|---|---|

Empties the 20-deep FIFO buffer in which front-panel touches are stored.

**ABS CLE <EOI>**

| | <NRx>,<NRx> | X=0 to11; Y=0 to 21 |
|---|---|---|

Touch panel screen coordinates range from 0,0 (upper left) to 10,21 (lower right). Front-panel buttons have the following coordinates:

| Waveform | 11,0 | Touch Panel | 11,5 |
|---|---|---|---|
| Trigger | 11,1 | Digitizer Run/Stop | 11,6 |
| Measure | 11,2 | Autoset | 11,7 |
| Store/Recall | 11,3 | Hardcopy | 11,8 |
| Utility | 11,4 | Enhanced Accuracy | 11,9 |

ABStouch cannot be used to touch a plug-in channel button or plug-in probe ID button.

**ABS 11,0 <EOI>**

*ABS? Note:* Every front-panel touch, whether from ABStouch or the front panel, is stored in a 20-deep FIFO (first-in, first-out) buffer. ABS? returns the touch coordinates from the buffer. If no touches are in the buffer, ABS? returns ABS –1,–1.

**ABS? <EOI>**
**ABS –1,–1**

Miscellaneous Commands

# DATE <qstring>

Sets or queries the date on the internal calendar.

| <qstring> | <dd>-<mon>-<yy> |
|-----------|-----------------|

where <dd> is the day of the month, <mon> is the first three letters of the month, and <yy> is the last two digits of the year.

**DATE '12-JAN-88' <EOI>**

# DEF <qstring>, <qstring>

Defines a logical name for an 11401/11402 command combination.

| <qstring> , <qstring> | |
|-----------------------|--|

Set Only. The first < qstring > is the logical name, the second < qstring > is the expansion command string that will be executed.

**DEF 'TB?','TBM?;TBW?' <EOI>**

*Notes:* Logical names cannot be used within a < qstring > . The first character of the logical name must be alphabetic. Case is ignored. An expansion string cannot be null (i.e., ''). The first character of an expansion string cannot be (:), (;), (,), <CR>, <LF>, or <space>.

# DSYmenu?

Query Only. Returns the name of the currently-active major menu on the front-panel display. Possible responses are:

| | |
|--------|--------------|
| CURSor | STORE_Recall |
| MEAS   | WAVfrm       |
| TRIgger| ALL_Wavfrm   |
| UTIlity| NONe         |

*Note:* ALL_Wavfrm is the paged waveform menu and NONe is a blank major menu.

**DSY? <EOI>**
**DSY MEAS**

## FEOi

**Set Only.** Forces the 11401/11402 to output the <EOI> delimiter for any pending query response.

**FEO <EOI>**

## FPAnel { ON | OFF }

Controls front-panel lock-out.

|  | ON | OFF |  |
|---|---|---|

FPA ON enables all front-panel functions (assuming the Touch Panel ON/OFF button is ON). FPA ON functionally mimics the GPIB LOCS state. FPA OFF locks out the front panel, similar to the GPIB RWLS state. The power-on default is FPA ON.

**FPA ON <EOI>**

## FPUpdate { ON | OFF | NEVer }

Determines whether the front-panel display readouts are updated following set command execution. The power-on default is FPU OFF.

|  | ON | OFF | NEVer |  |
|---|---|---|

With FPU ON, the front-panel display is updated after each successful set command.

With FPU OFF, the front-panel display is only updated when:
- the 11401/11402 receives DCL or SDC
- the 11401/11402 receives an incorrect query or set command
- the 11401/11402 input buffer is empty after a successful set execution.

With FPU NEVer, the front-panel display is not updated until FPU is changed to OFF or ON, or power is cycled on the instrument.

**FPU OFF <EOI>**

*Note:* Front-panel controls function with FPU ON or FPU OFF, but do not function with FPU NEVer.

# INIt

**Set Only.** Initializes the 11401/11402 to its factory-assigned default parameters/settings. For both GPIB and RS-232-C, the defaults are:
- DEBug is OFF
- SRQM USEr is OFF
- All pending events except "Power On" are discarded
- All user TEXt is cleared from the display
- For GPIB only, RQS is ON

**INI <EOI>**

# LONgform { ON | OFF }

Enables or disables the longer versions of query responses.

| | ON | OFF | |
|---|---|---|

With LONgform ON, ordinary queries respond with full header and link spellings, and the EVENT? and RS232 VERB:ON commands respond with a descriptive < qstring > as well as the event code. With LONgform OFF, query responses are in abbreviated form, and EVENT? and RS232 VERB:ON responses are only the event codes. The power-on default is LONgform ON. Nearly all examples in this quick reference are shown with LONgform OFF.

**LON OFF <EOI>**

# PATh { ON | OFF }

Enables or disables the return of headers and links to query commands. Default is PATh ON, which is how all examples in this manual are shown. With PATh OFF, only the arguments are returned to a query.

| PATh ON | PATh OFF |
|---|---|
| **CHL3? IMP**<br>CHL3 IMP:1.0E+6 | **CHL3? IMP**<br>1.0E+6 |
| **AUTOS?**<br>AUTOSET HOR:PER, VER:PP | **AUTOS?**<br>PER,PP |

## PATh { ON | OFF }                    (cont.)

### *PATh Notes:*

- PATh does not affect the ASCii or BINary SET? query response. Headers and links are returned regardless of the setting of PATh.
- PATh has no effect at the RS-232-C port when VERBose is ON.
- When PATh is OFF, only the headers of the DIAG? and FPSList? queries are removed. The links of these queries are always returned.
- When PATh is OFF, the query data returned is not acceptable as set command input and will generate error(s) if returned to the instrument.

## POWeron?

**Query Only.** Returns the total number of times the 11401/11402 has been powered up.

> POW? <EOI>
> POWERON 53

## PROBe { NT | NTAuto | SETSeq }

Selects the function performed when an 11000-Series plug-in probe ID button is pressed.

|  | NT | NTAuto | SETSeq |  |
|---|---|---|

PROBe NT either selects a displayed trace that includes the probe input channel, or if no displayed trace includes the probe channel, creates a new trace that contains only the probe channel. PROBe NTAuto is the same as PROBe NT except that the new trace is created using AUTOSet. PROBe SETSeq makes a probe button press recall the next set of stored front-panel settings from memory. Repeated button presses will sequentially recall all stored settings.

> **PROB NTA** <EOI>

## SPEaker { ON | OFF }

Enables or disables the audio response of the 11401/11402.

**SPE OFF** <EOI>

## TIMe <qstring>

Sets or queries the time of day on the internal clock.

| | |
|---|---|
| <qstring> | < hh > : < mm > : < ss > |

where < hh > is the hour in 24-hour format; < mm > is the minutes and < ss > the seconds.

**TIM '15:21:05'** <EOI>

## UNDEF { <qstring> | ALL }

**Set Only.** Undefines (removes) ALL or the specified logical name(s) defined with the DEF command.

| | |
|---|---|
| <qstring> | ALL | |

< qstring > removes the specified logical name; ALL removes all logical names.

**UNDEF "TB?"** <EOI>

## UPTime?

**Query Only.** Returns the total number of hours the 11401/11402 has been powered up.

**UPT?** <EOI>
UPTIME 4.257E2

## Record Position Commands

The record position commands control the main and window (delayed) sweep positioning.

## MAINPos <NRx>

Sets or queries the position of the Main acquisition record with respect to the Main trigger.

| <NRx> | – (main_duration) to 0 secs |
|-------|-----------------------------|

See the Time Base commands for duration calculation.

MAINP –6.0E–8 <EOI>

## WIN1Pos <NRx>;
## WIN2Pos <NRx>

WIN1P and WIN2P set or query the position of the Window 1 or Window 2 acquisition records with respect to the window trigger.

| <NRx> | (For range, see below) |
|-------|------------------------|

Range when WTMode = TIHoldoff:

| – ( TRW_TIH – MAINP + win_duration) to (main_duration + MAINP – TRW_TIH) |
|---|

Range when WTMode $\neq$ TIHoldoff:

| (MAINP – win_duration)   to (main_duration + MAINP) |
|---|

(See the Time Base commands for duration calculation.)

WIN1P 0; WIN2P –3.0E–6 <EOI>

## *Status and Event Commands*

The status and event commands control the reporting of the operating status of the 11401/11402 to the external controller or device. See also the Events section at the end of this manual.

# CONFig?

**Query Only.** Returns identifying information about plug-in unit names.

    CONF? <EOI>
    CONF LEF:"11A32",CEN:"11A71",
    RIG:"N/7K"

> *Note:* If a compartment is empty, its name is returned as "N/7K".

# EVENT?

**Query Only.** Returns the event code <NR1> if LONgform is OFF, or the event code plus descriptive <qstring> if LONgform is ON.

    EVENT? <EOI>
    EVENT 269,"NO SUCH TRACE"

> *Note:* LONgform is ON for the above example.

# ID?

**Query Only.** Returns the following comma-delimited information:
- Instrument model number
- TEK Codes & Formats version number
- Digitizer Processor (DIG) software version number
- Display Processor (DSY) software version number
- Executive Processor (EXP) software version number

    ID? <EOI>
    ID TEK/11402,V81.1,DIG/2.1,
    DSY/2.1,EXP/2.1

# IDProbe?

**Query Only.** Returns the plug-in unit and channel number (<slot> <ui>) of the last probe ID button pressed by the front-panel operator. Returns "L0" if no probe ID button was pressed.

    IDP? <EOI>
    IDP C2

# PIVersion?

**Query Only.** Returns identifying information about each plug-in unit software/firmware version number.

    PIV? <EOI>
    PIV LEF:"2.6",CEN:"2.6", RIG:"N/7K"

> *Note:* If a compartment is empty, its version number is returned as "N/7K".

# RQS { ON | OFF }

When RQS is ON, the 11401/11402 asserts SRQ after an event. When RQS is OFF, the 11401/11402 does not assert SRQ. The power-on default is RQS ON for GPIB and RQS OFF for RS-232-C (RQS OFF is the only allowable setting for RS-232-C).

    **RQS OFF** <EOI>

# STByte?

**RS-232-C Only. Query Only.** Permits the RS-232-C controller to read the status byte. See the Events section for more information.

    **STB?** <EOI>
    STB 37

> *Note:* "37" is the event code for an execution warning with RQS OFF.

# SRQMask <link>:<arg>

Controls the reporting of a specified class of events. At power-up, all SRQM <links> are ON except ABStouch, IDProbe, and USEr.

| ABStouch | ON \| OFF | |
|----------|-----------|--|

Enables or disables the reporting of front-panel touches either via the ABStouch command or screen touches (event code 451).

**SRQM ABS:ON <EOI>**

| CALDue | ON \| OFF | |
|--------|-----------|--|

Enables or disables instrument calibration due events (event codes 465 to 472).

**SRQM CALD:OFF <EOI>**

| CMDerr | ON \| OFF | |
|--------|-----------|--|

Enables or disables command errors (event codes 100 to 199).

**SRQM CMD:OFF <EOI>**

| EXErr | ON \| OFF | |
|-------|-----------|--|

Enables or disables execution errors (event codes 200 to 299).

**SRQM EXE:OFF <EOI>**

| EXWarn | ON \| OFF | |
|--------|-----------|--|

Enables or disables execution warnings (event codes 500 to 599).

**SRQM EXW:OFF <EOI>**

## SRQMask \<link>:\<arg>                    (cont.)

| IDProbe | ON \| OFF | |
|---------|-----------|---|

Enables or disables plug-in unit probe ID button press (event code 457).

**SRQM IDP:ON \<EOI>**

| INErr | ON \| OFF | |
|-------|-----------|---|

Enables or disables internal errors (event codes 300 to 399).

**SRQM INE:OFF \<EOI>**

| INWarn | ON \| OFF | |
|--------|-----------|---|

Enables or disables internal warnings (event codes 600 to 699).

**SRQM INW:ON \<EOI>**

| OPCmpl | ON \| OFF | |
|--------|-----------|---|

Enables or disables operation complete events (event codes 458, 460 to 464).

**SRQM OPC:OFF \<EOI>**

| USEr | ON \| OFF | |
|------|-----------|---|

Enables or disables RQS icon touch (event code 403). Also controls whether the RQS icon is displayed.

**SRQM USE:ON \<EOI>**

## UID <link>:<arg>

Queries or sets the serial numbers of the 11401/11402 and its plug-in units. (Setting of serial numbers is disabled by a default factory-set internal jumper; changing this jumper should only be done by qualified service personnel.) However, the UID can be queried regardless of the jumper position.

| CENter | <qstring> | 1 to 10 characters |
|--------|-----------|--------------------|

Queries or sets the serial number of the center plug-in unit.

UID? CEN <EOI>
UID CEN:"B010521"

| LEFt | <qstring> | 1 to 10 characters |
|------|-----------|--------------------|

Queries or sets the serial number of the left plug-in unit.

| MAIn | <qstring> | 1 to 10 characters |
|------|-----------|--------------------|

Queries or sets the serial number of the 11401/11402 mainframe.

| RIGht | <qstring> | 1 to 10 characters |
|-------|-----------|--------------------|

Queries or sets the serial number of the right plug-in unit.

*UID? Note:* If UID? is queried, the response is MAIn plus the LEFt, CENter, and RIGht plug-in units (assuming all are installed—nothing is returned for a slot that is empty).

UID? <EOI>
UID MAIN:"B010400",LEF:"B010562",
CEN:"B010521",RIG:"B010934"

## Time Base Commands

The time base commands select a time base and set acquisition scaling.

### Duration Calculation

The duration of the main time base is calculated by:

```
TBM_XINcr * (TBM_LENgth -1)
```

The duration of the window time base is calculated by:

```
TBW_XINcr * (TBW_LENgth -1)
```

The result is the time base duration in seconds.

## TBMain <link>:<arg>

Establishes main time base scaling.

| TIMe | <NRx> | 500e-12 to 100 secs |
|------|-------|---------------------|

Time per division (1-2-5 step intervals). The shortest TIMe values (500 ps to 5 ns) require small LENgth values:

| TIMe | LENgth |
|------|--------|
| 500 ps | 512 |
| 1 ns | 512, 1024 |
| 2 ns | 512, 1024, 2048 |
| 5 ns | 512, 1024, 2048, 4096, 5120 |

**TBM TIM:2E-2** <EOI>

| LENgth | <NRx> | 512\|1024\|2048\|4096 5120\|8192\|10240 |
|--------|-------|------------------------------------------|

Record length in points per waveform, such that TBM XIN is greater than or equal to TBW XIN.

**TBM LEN:2048** <EOI>

## TBMain <link>:<arg>                    (cont.)

| XINcr | < NR3 > |
|---|---|

**Query Only.** Returns the sample interval of the main time base, in seconds per point.

```
TBM? XIN  <EOI>
TBM XIN:2.0E-9
```

## TBWin <link>:<arg>

Establishes window time base scaling.

| TIMe | <NRx> | 500e-12 to TBM TIM secs |
|---|---|---|

Time per division (1-2-5 step intervals). The shortest TIMe values (500 ps to 5 ns) require small LENgth values:

| TIMe | LENgth |
|---|---|
| 500 ps | 512 |
| 1 ns | 512, 1024 |
| 2 ns | 512, 1024, 2048 |
| 5 ns | 512, 1024, 2048, 4096, 5120 |

Also, the window time base may not exceed the main time base.

```
TBW TIM:.001 <EOI>
```

| LENgth | < NRx > | 512\|1024\|2048\|4096 5120\|8192\|10240 |
|---|---|---|

Record length in points per waveform, such that TBW XIN is less than or equal to TBM XIN.

```
TBW LEN:1024 <EOI>
```

| XINcr | < NR3 > |
|---|---|

**Query Only.** Returns the sample interval of the window time base, in seconds per point.

```
TBW? XIN  <EOI>
TBW XIN:2.0E-8
```

## Triggering Commands

The triggering commands select and define the triggering system.

## TR?

**Query Only.** The TR? query is equivalent to entering: **TRMain?;TRWin?**. The response is:

```
TR? <EOI>
TRM <link>:<arg>,<link>:<arg>...;
TRW <link>:<arg>,<link>:<arg>...
```

## TRMain <link>:<arg>

Establishes main time base trigger values.

| ALEvel | <NRx> | 20 to 80 percent |
|--------|-------|------------------|

When trigger MODe is AUTOLevel, ALE sets the trigger level to a percentage value.

**TRM ALE:25 <EOI>**

| ANLevel | <NRx>,{DIVS\|VOLts} | (See ↓) |
|---------|---------------------|---------|

When trigger MODe is AUTO or NORmal, ANL sets the trigger level to the specified value in DIVS (AC coupled) or VOLts (DC coupled). [Be sure to set MODe, COUpling, and SOUrce before setting ANLevel.] DIVS range is –5 to +5; volts range is:

```
(–5 * CH_SEN + CH_OFFS)  to
(5 * CH_SEN + CH_OFFS)
```

**TRM ANL:150E–3,VOL <EOI>**

| COUpling | AC \| ACLf \| ACHf \| DCHf |
|----------|----------------------------|
|          | ACNoise \| DCNoise \| DC   |

Sets trigger coupling.

**TRM COU:DCN <EOI>**

# TRMain <link>:<arg>    (cont.)

| MODe | AUTO \| AUTOLevel \| NORmal |
|------|-----------------------------|

Sets triggering mode.

**TRM MOD:AUTO <EOI>**

| SLOpe | PLUs \| MINUs | |
|-------|---------------|--|

Sets trigger slope.

**TRM SLO:MINU <EOI>**

| SOUrce | <qstring> | LINE \| <exp> |
|--------|-----------|---------------|

Sets the trigger source to LINE or to the specified trigger expression, <exp>. Full expression syntax is found in the 11401/11402 User's Reference, but in brief, L and C channels can be added or subtracted together, but R channels can only combine with other Rs.

**TRM SOU:'L1' <EOI>**

| STAtus | TRG \| NOTrg |
|--------|--------------|

**Query Only.** Returns the trigger status of the main time base. TRG indicates the main time base is triggered; NOTrg, that it is not.

**TRM? STA <EOI>**
TRM STA:TRG

| TIHoldoff | <NRx> | 490e–9 to 10 secs |
|-----------|-------|-------------------|

Sets the main trigger TIme HOLDOFF in seconds.

**TRM TIH:25e–3 <EOI>**

# TRWin <link>:<arg>

Establishes window time base trigger values.

| ALEvel | <NRx> | 20 to 80 percent |
|---|---|---|

When trigger mode is AUTOLevel, ALE sets the trigger level to a percentage value.

**TRW ALE:40 <EOI>**

| COUpling | AC \| ACLf \| ACHf \| DCHf |
|---|---|
| | ACNoise \| DCNoise \| DC |

Sets trigger coupling.

**TRW COU:ACH <EOI>**

| EVHoldoff | <NRx> | 1 to 1e9 |
|---|---|---|

Sets the trigger EVent HOLDOFF to <NRx> number of events.

**TRW EVH:600 <EOI>**

| MODe | AUTOLevel \| NORmal | |
|---|---|---|

Sets the window triggering mode.

**TRW MOD:NOR <EOI>**

| NLEvel | <NRx>,{DIVS\|VOLts} | (See ↓) |
|---|---|---|

When trigger MODe is NORmal, NLE sets the trigger level to the specified value in DIVS (AC coupled) or VOLts (DC coupled). [Be sure to set MODe, COUpling, and SOUrce before setting NLEvel.] DIVS range is –5 to +5; volts range is:

```
(–5 * CH_SEN + CH_OFFS)  to
(5 * CH_SEN + CH_OFFS)
```

**TRW NLE:–2.6e–3,VOL <EOI>**

## TRWin \<link\>:\<arg\>                    (cont.)

| SLOpe | PLUs \| MINUs |
|---|---|

Sets trigger slope.

**TRW SLO:PLU  \<EOI\>**

| SOUrce | \<qstring\> | \<exp\> |
|---|---|---|

Sets the trigger source to the specified trigger
expression, \<exp\>. Full expression syntax is
found in the 11401/11402User's Reference, but in
brief, L and C channels can be added or
subtracted together, but R channels can only
combine with other Rs.

**TRW SOU:'C1+C2' \<EOI\>**

| STAtus | TRG \| NOTrg |
|---|---|

**Query Only.** Returns the trigger status of the
window time base. TRG indicates the window
time base is triggered; NOTrg, that it is not.

**TRW? STA  \<EOI\>**
TRW STA:NOT

| TIHoldoff | \<NRx\> | 20e-9 to 10* secs |
|---|---|---|

Sets the window trigger Time HOLDOFF in
seconds. *Upper limit is bounded by the main
time base duration; 10 seconds maximum.

**TRW TIH:20E-9 \<EOI\>**

# WTMode { MAIn | EVHoldoff | TIHoldoff }

Sets window triggering mode.

| MAIn | |
|---|---|

Window trigger coincides with the main trigger; the window trigger is not held off.

WTM MAI <EOI>

| EVHoldoff | |
|---|---|

Window time base trigger is held off the number of events specified by TRW EVH.

WTM EVH <EOI>

| TIHoldoff | |
|---|---|

Window time base trigger is held off for a time equal to the trigger holdoff (TRW TIH).

WTM TIH <EOI>

## Waveform and Settings Commands

The waveform and settings commands select, store, remove, and specify waveforms and front-panel settings (FPS).

### ADJtrace<ui> <link>:<arg>

Adjusts the position of the specified trace without modifying the time base or plug-in unit parameters. Range of $<ui>$ is 1 to 8.

| HMAg | <NRx> | 1, 2, 2.5, 4, 5, 10, 20 |
|---|---|---|

Sets the trace's horizontal magnification factor when PANzoom is ON. Maximum HMAg value depends on record LENgth of TBM or TBW.

**ADJ1 HMA:2 <EOI>**

| HPOsition | <NRx> | 0 to 9728 |
|---|---|---|

Sets the trace's horizontal position when PAN-zoom is ON. HPOsition range is in waveform points.

**ADJ3 HPO:900 <EOI>**

| HVPosition | <NRx> | −1e15 to 1e15 |
|---|---|---|

Sets the graphical position of the horizontal component of the specified XY trace, only if the XY trace was created in floating-point mode (TRAce WFMCalc:HIPrec).

**ADJ6 HVP:−1E3 <EOI>**

| HVSize | <NRx> | 1e−15 to 1e15 |
|---|---|---|

Sets the graphical size of the horizontal component of the specified XY trace, only if the trace was created in floating-point mode (TRAce WFMCalc: HIPrec).

**ADJ7 HVS:2e−3 <EOI>**

## ADJtrace<ui> <link>:<arg>     (cont.)

| PANzoom | ON \| OFF | |
|---------|-----------|--|

Enables or disables pan/zoom mode. PANzoom is always enabled for stored or scalar traces. However, XY traces cannot have PAN enabled.

**ADJ4 PAN:ON   <EOI>**

| TRSep | <NRx> | −5.0 to +5.0 |
|-------|-------|--------------|

Sets the window trace separation in graticule divisions only if the trace is not XY and was created:
- on the WIN1 or WIN2 timebase
- in integer mode ( TRAce WFMCalc:FASt).

**ADJ2 TRS:−2.35   <EOI>**

| VPOsition | <NRx> | −1e15 to 1e15 |
|-----------|-------|---------------|

Sets the trace's vertical graphical position only if the trace was created in floating-point mode (TRAce WFMCalc:HIPrec).

**ADJ5 VPO:1.7e2   <EOI>**

| VSIze | <NRx> | 1e−15 to 1e15 |
|-------|-------|---------------|

Sets the trace's vertical graphical size only if the trace was created in floating-point mode (TRAce WFMCalc:HIPrec).

**ADJ7 VSI:2e−3   <EOI>**

*ADJ[<ui>]? Note:* ADJ<ui>? returns links (all or only those specified) with their current argument(s). ADJ? returns all links for all currently-defined traces, in low-to-high trace order.

    ADJ?   <EOI>
    ADJ1 <link>:<arg>,<link>:<arg>...;
    ...
    ADJ8 <link>:<arg>,<link>:<arg>...

# ADJtrace<ui> <link>:<arg>     (cont.)

***ADJ? Restrictions:*** Several ADJtrace links can only be set under restricted conditions, but can be queried at any time. These links return the following predefined values if queried while they cannot be set:

| | |
|---|---|
| HMAg | -1 |
| HPOsition | 1e16 |
| HVPosition | 1e16 |
| HVSize | -1 |
| TRSep | 1e16 |
| VPOsition | 1e16 |
| VSIze | -1 |

# AVG { ON | OFF }

Turns averaging on or off a vertical expression component < Y_exp > of a trace description. (See also TRAce and ENV commands.)

| | ON | OFF | |
|---|---|---|

When < Y_exp > is *not* enclosed with ENV and AVG is set ON, < Y_exp > is enclosed with AVG().

When < Y_exp > is enclosed with ENV and AVG is set ON , AVG() replaces ENV().

When < Y_exp > is enclosed with AVG() and AVG is set OFF, the enclosing AVG() is removed.

| <u>"Before"</u> | <u>Command</u> | <u>"After"</u> |
|---|---|---|
| R2 | AVG ON | AVG(R2) |
| ENV(L1–L2) | AVG ON | AVG(L1–L2) |
| AVG(L3) | AVG OFF | L3 |
| AVG(L3) | AVG ON | AVG(AVG(L3)) |

***AVG? Note:*** AVG? returns the current argument. AVG ON means the entire vertical description is enclosed by AVG. AVG OFF means the entire vertical description isn't enclosed, although the AVG() function may be embedded within the description.

# CLEar { ALL | TRAce<ui> }

**Set Only.** Clears (sets to null values) one or all displayed traces. (See also the REMove command.)

| | ALL<br>TRAce <ui> | 1 to 8 |
|---|---|---|

**Set Only.** Clears all or the specified displayed trace(s). Not an error if CLEar ALL is issued when no traces are defined.

**CLE TRA:5**  <EOI>

# DELete { [ALL:] STO<ui> | FPS<ui> }

**Set Only.** Deletes a stored waveform or front-panel setting. Note that deleting a stored waveform which is the only component of an active trace will remove the trace from the display.

| ALL | STO | FPS | |
|---|---|---|

**Set Only.** ALL:STO deletes all stored waveforms; ALL:FPS deletes all stored front-panel settings. Not an error if DEL ALL:xxx is issued when no waveforms or settings are stored.

**DEL ALL:FPS**  <EOI>

| | STO<ui> | FPS<ui> | 1 to 256 (STO)<br>1 to 10 (FPS) |
|---|---|---|

**Set Only.** Deletes the specified stored waveform or front-panel setting.

**DEL STO:201**  <EOI>

## ENV { ON | OFF }

Turns enveloping on or off a vertical expression component $<Y\_exp>$ of a trace description. (See also the TRAce and AVG commands.)

| ON | OFF |
|---|---|
|  |  |

When $<Y\_exp>$ is *not* enclosed with AVG and ENV is set ON, $<Y\_exp>$ is enclosed with ENV().

When $<Y\_exP>$ is enclosed with AVG and ENV is set ON , ENV() replaces AVG().

When $<Y\_exp>$ is enclosed with ENV() and ENV is set OFF, the enclosing ENV() is removed.

| "Before" | Command | "After" |
|---|---|---|
| R2 | ENV ON | ENV(R2) |
| AVG(L1–L2) | ENV ON | ENV(L1–L2) |
| ENV(L3) | ENV OFF | L3 |
| ENV(L3) | ENV ON | ENV(ENV(L3)) |

*ENV? Note:* ENV? returns the current argument. ENV ON means the entire vertical description is enclosed by ENV. ENV OFF means the entire vertical description isn't enclosed, although the ENV() function may be embedded within the description.

## FPSList?

**Query Only.** Returns a list of all front-panel settings stored in nonvolitile RAM or "EMPTY" if there are none.

| EMPTY | (FPS $<ui>$ : $<seq>$ , $<len>$ )... |
|---|
|  |

The FPSL? query returns the FPS number (1 to 10), followed by the sequence number of that setting (1 to 10) and the setting's byte length (an unsigned integer).

FPSL? <EOI>
FPSL FPS2:1,1056,FPS5:2,979

# FPSNum?

**Query Only.** Returns the number of front-panel settings (FPS) stored in nonvolitile RAM, in <NR1> form. Range is 0 to 10.

> **FPSN?** <EOI>
> FPSN 2

# NAVg <NRx>

Sets or queries the number of waveform samples to be averaged when conditional acquisition is set to averaging. (See the CONDacq command.)

| <NRx> | 2 to 4096 |
|---|---|

**NAV 1000** <EOI>

# NENV <NRx>

Sets or queries the number of waveform samples to be enveloped when conditional acquisition is set to envelope complete. (See the CONDacq command.)

| <NRx> | 2 to 4096 |
|---|---|

**NENV 100** <EOI>

# NVRam?

**Query Only.** Returns the number of bytes of unallocated nonvolitile RAM available for storing front-panel settings, in <NR1> form.

> **NVR?** <EOI>
> NVR 1195

## RECall { FPS<ui> | FPSNext }

**Set Only.** Recalls a stored front-panel setting (FPS) from memory.

| | | |
|---|---|---|
| | FPS<ui><br>FPSNext | 1 to 10 |

**Set Only.** Recalls the specified FPS, or the next FPS in sequence if SETSeq is ON.

**REC FPS9** <EOl>

## REMove { ALL | TRAce<ui> }

**Set Only.** Removes traces from the screen.

| | | |
|---|---|---|
| | ALL<br>TRAce<ui> | 1 to 8 |

**Set Only.** Removes from the screen ALL or the specified displayed trace(s). If the trace is stored in memory, the stored waveform is not removed. Not an error if REM ALL is issued when no traces are defined.

**REM ALL** <EOl>

## SELect TRA<ui>

Defines the "selected trace" used by autoset, measurement, and cursor commands.

| | | |
|---|---|---|
| | TRAce<ui> | 0 to 8 |

TRA0 is returned by SEL? when there is no selected trace. Setting SEL TRA0 is meaningless and is ignored (no error.)

**SEL TRA5** <EOl>

## SETSeq { ON | OFF }

Enables or disables settings sequencing. If SETS is ON and all stored settings are deleted, SETS is set OFF. If SETS is OFF and PRObe SETSeq is issued, SETS is set ON.

**SETS ON** <EOl>

## STOre { FPS<ui> | TRA<ui>:STO<ui> }

**Set Only.** Saves the current front-panel setting (FPS) in nonvolitile RAM or saves a specified trace in memory.

| | FPS < ui > | 1 to 10 |
|---|---|---|

**Set Only.** Saves the current front-panel setting as the indicated FPS number.

**STO FPS3** <EOI>

| TRAce < ui > | STO < ui > | 1 to 8 (TRA) 1 to 256 (STO) |
|---|---|---|

**Set Only.** Stores the indicated trace at the specified STO<ui> location. XY traces cannot be stored. An existing STO<ui> location can be over-written if the record lengths of the new and stored traces are the same. If the old stored trace was a component of an active trace, the active trace changes to include the newly-stored waveform.

**STO TRA2:STO50** <EOI>

## STOList?

**Query Only.** Returns a list of all stored waveforms in the form STO<ui>, or "EMPTY" if none are stored.

**STOL?** <EOI>
STOL STO1,STO2,STO50,STO201

## STONum?

**Query Only.** Returns the number of waveforms currently stored in memory.

**STON?** <EOI>
STON 4

# TRAce<ui> <link>:<arg>

Defines the characteristics of a trace. Range of trace numbers is 1 to 8.

| ACCumulate | ON | OFF |
|---|---|---|

Enables or disables point accumulate (PA). PA cannot be defined for an XY trace, nor enabled for a stored or scalar trace, nor enabled when the record LENgth is greater than 2048.

**TRA5 ACC:OFF  <EOI>**

| ACState | ENHanced | NENHanced |
|---|---|

**Query Only.** Indicates whether the specified trace was created in enhanced accuracy mode or not.

**TRA7? ACS  <EOI>**
**TRA7 ACS:ENH**

| DEScription | < qstring > | 55 chars max |
|---|---|---|

Describes the source of the waveform. The general format is:

<Y_exp> [vs <X_exp>] [on <time_base>]

where expressions <Y_exp> and <X_exp> can range from a single source to a combination of inputs, stored traces, and waveform functions. (See the 11401/11402 User's Reference for complete syntax.) If [vs <X_exp>] is omitted, the trace is YT; if included, the trace is XY. The <time_base> can be MAIN|WIN1|WIN2; if [on <time_base>] is omitted, the default is MAIN.

**TRA6 DES:'L1'  <EOI>**
**TRA4 DES:'ENV(R1 + STO26)'<EOI>**
**TRA5 DES:'C2-C1 ON WIN2'  <EOI>**

| GRLocation | UPPer | LOWer |
|---|---|---|

Positions the selected trace to the upper or lower graticule pair.

**TRA5 GRL:LOW  <EOI>**

# TRAce<ui> <link>:<arg>     (cont.)

| GRType | LINear | |
|--------|--------|--|

Sets the selected trace's graticule type to linear.
(Linear is the only option currently available).

     TRA8 GRT:LIN   <EOI>

| WFMCalc | FASt | HIPrec |
|---------|--------|--------|

**Query Only.** Indicates whether a trace was
created in integer (FASt) mode or floating-point
(HIPrec) mode. A trace cannot be changed from
the mode in which it was created.

       TRA2? WFMC   <EOI>
       TRA2 WFMC:FAS

| XUNit | AMPS \| DIVS \| OHMs \| SEConds \| VOLts \| WATs |
|-------|---------------------------------------------|

**Query Only.** Returns the horizontal units of the
specified trace.

       TRA3? XUN    <EOI>
       TRA3 XUN:OHM

| YUNit | AMPS \| DIVS \| OHMs \| VOLts \| WATs |
|-------|--------------------------------------|

**Query Only.** Returns the vertical units of the
specified trace.

       TRA3 YUN   <EOI>
       TRA3 YUN:VOL

*TRA[<ui>]? Note:* TRAce<ui>? returns
links (all or only those specified) with their
current arguments. TRA? returns all links for
all currently-defined traces, in low-to-high
trace order.

       TRA2?  <EOI>
       TRA2 DES:"L3",ACC:OFF,ACS:ENH,
       GRL:UPP,GRT:LIN,WFMC:FAS,
       XUN:SEC,YUN:VOL

# TRANUm?

**Query Only.** Returns the number of traces displayed on the screen. Range is 0 to 8, in <NR1> form.

> **TRANU?** <EOI>
> TRANU 7

# WFMScaling { FORce | OPTional }

Determines whether a new trace is created in floating-point mode (FORce), or in integer mode when possible (OPTional). (Traces created in integer mode have faster display update rates.) Certain trace types require floating-point or integer mode, regardless of the WFMS setting.

| FORce | Floating-point mode |
|---------|---------------------|
| OPTional | Integer mode |

> **WFMS OPT** <EOI>

# Event Reporting

## Status Byte Codes

The 11401/11402 reports 10 status conditions in its status byte — five normal events and five error or warning conditions.

Table 3-1 gives the status byte codes in binary and in decimal with both RQS ON and RQS OFF. The bits in the status byte are active high. Bits 1 through 4 are system status bits. Bit 5 is the busy bit and is asserted only during diagnostics. Bit 6 is the error bit. Bit 7 (shown as "R" in the table) indicates whether RQS is ON or OFF. (Bit 7 is always low until specifically enabled with the RQS command.) Bit 8 is always low.

**Table 3-1. Status Byte Codes**

| Condition | BINARY Status Bits 8765 4321 | | DECIMAL RQS ON | RQS OFF |
|---|---|---|---|---|
| *Normal:* | | | | |
| No Status To Report | 0000 | 0000 | 0 | 0 |
| Power On | 0R00 | 0001 | 65 | 1 |
| Operation Complete | 0R00 | 0010 | 66 | 2 |
| User Request | 0R00 | 0011 | 67 | 3 |
| Calibration Due | 0R00 | 0110 | 70 | 6 |
| *Abnormal:* | | | | |
| Command Error | 0R10 | 0001 | 97 | 33 |
| Execution Error | 0R10 | 0010 | 98 | 34 |
| Internal Error | 0R10 | 0011 | 99 | 35 |
| Execution Warning | 0R10 | 0101 | 101 | 37 |
| Internal Warning | 0R10 | 0110 | 102 | 38 |

## Event Code Reporting

GPIB and RS-232-C controllers read event codes via the EVENT? query command. The query response depends on whether LONgform is ON or OFF. If LONgform is OFF, the event query returns:

```
EVENT <NR1>
```

Event Reporting

If LONgform is ON, the event query returns:

```
EVENT <NR1>,<qstring>
```

where <NR1> is the event code and <qstring> is the description from the event table. In some cases, the text in the tables contains formatting codes which expand as follows:

%a  Channel number or unsigned integer

%A  Argument name

%b  Plug-in slot: L, R, or C

%B  Plug-in slot: LEFT, RIGHT, or CENTER

%M  Mainframe calibration fault string. If no error occurred, %M is replaced by "Pass"; otherwise %M is replaced by a short descriptive of what caused the mainframe failure, for example, "Main Fine Holdoff".

%P  Plug-in calibration fault list. If mainframe calibration fails, %P is replaced with "NA". If mainframe calibration passes and plug-in calibration passes, %P is replaced with "Pass". Otherwise %P is replaced with a comma-delimited list of plug-in slots in the format: <slot><dd>, where <slot> is L, C, or R, and <dd> is a two-digit hex number whose first digit is always 0 and whose second digit is binary weighted to encode which channel failed. (Bits are weighted 8, 4, 2, 1 to represent channels 4, 3, 2, and 1, respectively.)

%t  Trigger calibration fault list. If mainframe calibration fails, %t is replaced with "NA". If mainframe calibration passes and both main and window trigger calibration passes for each plug-in slot, %t is replaced with "Pass". Otherwise, %t is replaced with a comma-delimited list of failures in the format: <trig><slot>, where <trig> is M (main) or W (window) and <slot> is L, C, or R.

%T  Time, as "X minutes and Y seconds." If X is 0, then "X minutes" is omitted. If Y is 0, then "Y seconds" is omitted

%?  Event code value

%? Event code value

For example, the following set command causes an execution warning, event code 550:

**CHL1 OFFSET: 5000** <EOI>

Event code 550 has this entry in Table 3-8, Execution Warnings:

| Code | Event Description |
|------|-------------------|
| 550 | %A out of range — limit set |

If LONgform is OFF, the event is reported:

**EVENT?** <EOI>
EVENT 550

If LONgform is ON, the response is:

**EVENT?** <EOI>
EVENT 550,"Offset out of range — limit set"

# *Command Errors*

Table 3-2 lists the error codes and descriptions for all command errors. Command errors can be masked with SRQMask CMDerr:OFF. The status byte for all command errors is **97** with RQS ON and **33** with RQS OFF.

### Table 3-2. Command Errors

| Code | Event Description |
| --- | --- |
| 108 | Checksum error in binary block transfer |
| 109 | Illegal byte count value on a binary block transfer |
| 154 | Invalid number input |
| 155 | Invalid string input |
| 156 | Symbol not found |
| 157 | Syntax error |
| 160 | Expression too complex |
| 161 | Excessive number of points in binary CURVE data input |
| 162 | Excessive number of points in ASCII CURVE data input |
| 163 | No input terminator seen |
| 164 | Binary block input not allowed with ECHO ON |
| 167 | Insufficient data to satisfy binary block byte count |
| 168 | Unsupported constant |
| 169 | Unsupported function |

# Execution Errors

Table 3-3 lists the error codes and descriptions for all execution errors. Execution errors can be masked with SRQMask EXErr:OFF. The status byte for all execution errors is **98** with RQS ON and **34** with RQS OFF.

### Table 3-3. Execution Errors

| Code | Event Description |
| --- | --- |
| 203 | I/O buffers full |
| 205 | %A out of range—value ignored |
| 224 | Function not available in selected plugin range |
| 231 | Autoset—not functional with this waveform type |
| 232 | That XY waveform has incompatible components |
| 233 | Delayed trace must not be the selected trace |
| 239 | Improper version number |
| 240 | Can't accumulate nonacquired waveform |
| 241 | Too many acquisitions |
| 242 | Enhanced accuracy available after %T |
| 243 | That function is disabled by a hardware strap |
| 244 | %B plugin channel(s) used differently in main and window sources |
| 246 | Can't sequence settings |
| 247 | No settings defined |
| 248 | Misuse of AVG/ENV function |
| 249 | Illegal use of trace positioning function |
| 250 | No traces defined |
| 251 | Illegal trace number |
| 252 | Illegal stored settings number |
| 255 | Out of memory |
| 257 | Illegal stored waveform number |
| 263 | Illegal channel number |
| 264 | No further XY waveforms may be defined |
| 265 | Illegal DATE/TIME |
| 266 | DEF expansion overflow |
| 267 | Illegal DEF string |
| 268 | Illegal DEF recursion |

Event Reporting

The status byte for all execution errors is **98** with RQS ON and **34** with RQS OFF.

### Table 3-3. Execution Errors (cont.)

| Code | Event Description |
|------|-------------------|
| 269 | No such trace |
| 270 | No such stored waveform |
| 271 | No such DEF |
| 272 | That function is not supported by this plugin |
| 273 | No such FPS |
| 274 | No appropriate 11K plugins loaded |
| 275 | %B slot not loaded with appropriate 11K plugin |
| 278 | Plugin channel used more than once in trigger source |
| 279 | Line trigger not available for window trigger source |
| 280 | Invalid smooth argument |
| 281 | Can't delete active stored waveform |
| 282 | Can't store trace |
| 283 | Can't clear nonacquired waveform |
| 284 | Requested coupling for channel %a not available on %B plugin |
| 285 | Requested input impedance for channel %a not available on %B plugin |
| 286 | Too many measurements specified |
| 287 | Hardcopy absent or off line |
| 288 | Inappropriate trigger level units |
| 289 | Split cursors not permitted on XY trace |
| 290 | Current reference measurement failed |
| 291 | TEXT not permitted when acquired XY trace is active |
| 292 | %B slot not loaded with 11K plugin unit |
| 293 | Misuse of 11K plugin unit |
| 294 | Dual graticules not permitted with XY trace |
| 295 | Record length too long for Point Accumulate waveform |
| 296 | Point Accumulate and XY waveforms are mutually exclusive |
| 297 | Panzoom may not be enabled |
| 298 | Panzoom may not be disabled |
| 299 | CONDACQ function not available |

## Internal Errors

Table 3-4 lists the error codes and descriptions for all internal errors. Internal errors can be masked with SRQMask INErr:OFF. The status byte for all internal errors is **99** with RQS ON and **35** with RQS OFF.

### Table 3-4. Internal Errors

| Code | Event Description |
|------|-------------------|
| 308 | Bad level 2 probe checksum on channel %b%a |
| 327 | DIG probe compensation failed |
| 328 | DIG plugin calibration failed |
| 329 | DIG deskew failed |
| 330 | Enhanced accuracy failed. Mainframe: %M. Plugin: %P. Trigger: %t. |
| 394 | Test completed and failed |
| 395 | General DIG failure detected (code = %a) |
| 396 | %B plugin communication failure |
| 397 | Internal DAC overflow on channel %a of %B plugin |
| 398 | Invalid DIG table ID detected |
| 399 | Invalid DIG field ID detected |

## System Events

Table 3-5 lists the event codes, SRQMask links, status byte values (for RQS ON and RQS OFF), and descriptions for four normal system events.

### Table 3-5. System Events

| Code | SRQMask | Status Bytes | Event Description |
|------|---------|--------------|-------------------|
| 400 | — — | 0, 0 | System function normal |
| 401 | — — | 65, 1 | Power on |
| 403 | USEr | 67, 3 | Front panel RQS icon selected |
| 451 | ABStouch | 67, 3 | Abstouch |
| 457 | IDProbe | 67, 3 | Probe %a ID button pressed on %B plugin |

## Operation Complete Events

Table 3-6 lists the codes and descriptions for all operation complete events. These events can be masked with SRQMask OPCmpl:OFF. The status byte for all operation complete events is 66 with RQS ON and 2 with RQS OFF.

Table 3-6. Operation Complete Events

| Code | Event Description |
|------|-------------------|
| 450 | Conditional acquire completed |
| 458 | Hardcopy aborted |
| 460 | Test completed and passed |
| 461 | Enhanced accuracy complete and passed |
| 462 | Hardcopy complete |
| 463 | Measurements complete |
| 464 | Autoset complete |

## Calibration Due Events

Table 3-7 lists the codes and descriptions for all calibration due events. These events can be masked with SRQMask CALDue:OFF. The status byte for all calibration due events is 70 with RQS ON and 6 with RQS OFF.

Table 3-7. Calibration Due Events

| Code | Event Description |
|------|-------------------|
| 465 | Warmup complete — calibration due |
| 466 | New configuration — partial enhanced accuracy occurring |
| 467 | Warmup complete with new configuration — calibration due |
| 468 | Warmup complete with new configuration — automatic enhanced accuracy occurring |
| 469 | Temperature change — automatic enhanced accuracy occurring |
| 470 | Temperature change — calibration due |
| 471 | Warmup complete — enhanced accuracy in effect |
| 472 | Warmup complete — automatic enhanced accuracy occurring |

## *Execution Warnings*

Table 3-8 lists the codes and descriptions for all execution warnings. Execution warnings can be masked with SRQMask EXWarn:OFF. The status byte for all execution warnings is **101** with RQS ON and **37** with RQS OFF.

**Table 3-8. Execution Warnings**

| Code | Event Description |
|------|-------------------|
| 550 | %A out of range — limit set |
| 551 | Insufficient data to satisfy binary block byte count |
| 552 | Checksum error in binary block transfer |
| 553 | Window trigger source set equal to main trigger source |
| 554 | Autoset — no signal detected |
| 555 | Binary curve odd data byte discarded |
| 556 | No active acquisitions — digitizer remains stopped |
| 557 | Hardcopy aborted |
| 558 | Nothing to abort |
| 559 | XY PT.FMT not permitted — Y assumed |
| 561 | Autoset — trigger search failed |
| 562 | Autoset — horizontal search failed |
| 563 | Autoset — ac signal too large |
| 564 | Autoset — dc signal too large |

Index

## Tektronix 11401/11402 Functional Command Summary

### Key

| | | |
|---|---|---|
| < > | ::= | Defined item |
| { } | ::= | One item from group required |
| [ ] | ::= | Optional item(s) |
| ( ) | ::= | Grouped items |
| ... | ::= | Group can be repeated |
| \| | ::= | Exclusive or |
| FPS | ::= | Front Panel Setting |
| <NR1> | ::= | Signed integer |
| <NR2> | ::= | Floating point, no exponent |
| <NR3> | ::= | Floating point with exponent |
| <NRx> | ::= | { <NR1> \| <NR2> \| <NR3> } |
| <ui> | ::= | Unsigned integer |
| <slot> | ::= | {L\|C\|R} |
| <qstring> | ::= | Quoted string |
| <bblock> | ::= | Tek "Codes&Formats" binary block data |
| TYPe | | Spelling of <u>header</u> or <u>link</u>; minimum spelling in **CAPs** |
| SINgle | | Spelling of <u>argument</u>; minimum spelling in CAPs |
| ? | | Query-only header or link |

Command headers are flush-left, followed by their syntax. Links are indented.
Query-only links are indicated with a leading **?**; the "argument" after the colon shows the form of the response.

## Acquisition Commands

AUTOSet [<link>:] <arg>
    HORiz: {PERiod|OFF}
    STARt                          *(Set-only)*
    UNDO                           *(Set-only)*
    VERt: {ECL|PP|TTL|OFF}
CONDacq <link>:<arg>
    FILI: <NRx>
    ? REMAining .<NR1>
    TYPe: {FILI|AVG|ENV|BOTh|SINgle| CONTInuous}
DIGitizer {RUN|STOP}

## Calibration Commands

CCAIconstants <ui>:<NRx>
CALStatus?
LCAIconstants <ui>:<NRx>
MCAIconstants <ui>:<NRx>
RCAIconstants <ui>:<NRx>
SELFcal [<link>:] <arg>
    FORce                          *(Set-only)*
    MODe: {AUTO|MANual}

## Channel Commands

CH<slot> <ui> <link>:<arg>
    AMPoffset: <NRx>
    BW: <NRx>
    COUpling: {AC|DC|OFF}
    IMPedance: <NRx>
    MNSCoupling: {AC|DC|VC|OFF}
    MNSOffset: <NRx>
    ? MNSProbe: <qstring>
    OFFSet: <NRx>
    PLSCoupling: {AC|DC|VC|OFF}
    PLSOffset: <NRx>
    ? PLSProbe: <qstring>
    ? PROBe: <qstring>
    PROTect: ON|OFF
    SENsitivity: <NRx>
    ? UNIts: <qstring>
    VCOffset: <NRx>
CH[<slot>]?

## Cursor Commands

CURSor <link>:<arg>
    REAdout: {ON|OFF}
    REFErence: TRAce<ui>
    TYPe: {PAIred|SPLit|VBArs|HBArs}
    ? XUNit: {AMPS|DIVS|OHMs|SEConds| VOLts|WATs}
    ? YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}
DOT1Abs, DOT2Abs <link>:<arg>
    PCTg: <NRx>
    XCOord: <NRx>
    XDIv: <NRx>

DOT1Abs, DOT2Abs <link>:<arg>     (cont.)
  **?** XQUal: {EQ|LT|GT|UN}
  **?** YCOord: <NR3>
  **?** YDIv: <NR3>
  **?** YQUal: {EQ|LT|GT|UN}

DOT1Rel, DOT2Rel <link>:<arg> *(Set-only)*
  PCTg: <NRx>                        *(Set-only)*
  XCOord: <NRx>                    *(Set-only)*
  XDIv: <NRx>                         *(Set-only)*

H1Bar, H2Bar <link>:<arg>
  YCOord: <NRx>
  YDIv: <NRx>

V1Bar, V2Bar <link>:<arg>
  XCOord: <NRx>
  XDIv: <NRx>

## Data Transfer Commands

ABBwfmpre {ON|OFF}

BYT.or {LSB|MSB}

CURVe { <bblock> | (<NRx> [, <NRx>]...)}

ENCdg <link>:<arg>
  SET: {ASCii|BINary}
  WAVfrm: {ASCii|BINary}

INPut STO<ui>

OUTput {STO<ui> | TRAce<ui>}

SET <bblock>                         *(Set-only)*

SET?

WAVfrm?

WFMpre <link>:<arg>
  ACState: {ENHanced|NENhanced}
  **?** BIT/nr: 16
  **?** BN.fmt: RI
  **?** BYT/nr: 2
  **?** BYT.or: {LSB|MSB}
  **?** CRVchk: {CHKsm0|NONe|NULl}
  **?** ENCdg: {ASCii|BINary}
  NR.pt: <NRx>
  PT.fmt: {ENV|Y|XY}
  **?** WFId: {STO<ui> | TRAce<ui>}
  XINcr: <NRx>
  **?** XMUlt: <NR3>

WFMpre <link>:<arg>                   (cont)
  **?** XUNit: {AMPS|DIVS|OHMs|SEConds|
         VOLts|WATs}
  XZEro: <NRx>
  YMUlt: <NRx>
  YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}
  YZEro: <NRx>

## Diagnostics Commands

DIAg?

TESt [XTNd]                         *(Set-only)*

## Display Commands

DISPlay <link>:<arg>
  GRAticle: {DUAl|SINgle}
  INTensity: <NRx>
  MODe: {DOTs|VECtors}

TEXt [<link>:] <arg>                 *(Set-only)*
  CLEar                               *(Set-only)*
  STRing: <qstring>              *(Set-only)*
  X: <NRx>                            *(Set-only)*
  Y: <NRx>                            *(Set-only)*

## External I/O Commands

COPy [<link>:] <arg>
  ABOrt                               *(Set-only)*
  FORMat: {DRAft|HIRes|REDuced}
  PRInter: {PIN8|PIN24}
  STARt                               *(Set-only)*

DEBug <link>:<arg>
  GPIb: {ON|OFF}
  RS232: {ON|OFF}

RS232 <link>:<arg>
  BAUd: <NRx>
  DELAy: <NRx>
  ECHo: {ON|OFF}
  EOL: {CR|CRLf|LF|LFCr}
  FLAgging: {SOFt|HARd|OFF}
  PARity: {ODD|EVEN|NONe}
  STOPBits: <NRx>
  VERBose: {ON|OFF}

## Measurement Commands

```
<meas> ::= {CROss|DELAy|FALltime|FREq
    |MAX|MEAN|MID|MIN|PDElay|PERiod|PP|
    RISetime|RMS|TTRig|WIDth|YTEnergy|
    YTMns_area|YTPls_area}
```

BASeline <NRx>

COMpare {ON|OFF}

DAInt {WHOle|SINgle}

DISTal <NRx>

DLYtrace TRAce<ui>

LMZone <NRx>

MEAS?

<meas>?

MESial <NRx>

MSLIst {<meas>|EMPty}

MSLOpe {PLUs|MINUs}

MSNum?

MSYs {ON|OFF}

MTRack {ON|OFF}

PROXimal <NRx>

REFLevel <NRx>

REFSet <link>:<arg>
    CURRent: <meas>                    *(Set-only)*
    <meas>: <NRx>

RMZone <NRx>

SNRatio <NRx>

TOPline <NRx>

TTAverage <NRx>

## Miscellaneous Commands

ABStouch {<NRx>,<NRx>|CLEar}

DATE <qstring> = "dd-mmm-yy"

DEF <qstring>,<qstring>              *(Set-only)*

DSYmenu?

FEOi                                 *(Set-only)*

FPAnel {ON|OFF}

FPUpdate {ON|OFF|NEVer}

INIt                                 *(Set-only)*

LONgform {ON|OFF}

PATh {ON|OFF}

POWeron? <NR1>

PROBe {NT|NTAuto|SETSeq}

SPEaker {ON|OFF}

TIMe <qstring> = "hh:mm:ss"

UNDEF {<qstring>|ALL}                *(Set-only)*

UPTime?

## Record Position Commands

MAINPos <NRx>

WIN1Pos <NRx>

WIN2Pos <NRx>

## Status and Event Commands

CONFig?

EVENT?

ID?

IDProbe?

PIVersion?

RQS {ON|OFF}

STByte?

SRQMask <link>:<arg>
    ABStouch:{ON|OFF}
    CALDue: {ON|OFF}
    CMDerr: {ON|OFF}
    EXErr: {ON|OFF}
    EXWarn: {ON|OFF}
    IDProbe: {ON|OFF}
    INErr: {ON|OFF}
    INWarn: {ON|OFF}
    OPCmpl: {ON|OFF}
    USEr: {ON|OFF}

UID <link>:<arg>
    CENter: <qstring>
    LEFt: <qstring>
    MAIn: <qstring>
    RIGht: <qstring>

## Timebase Commands

TBMain <link>:<arg>
   LENgth: <NRx>
   TIMe: <NRx>
   ? XINcr: <NR3>

TBWin <link>:<arg>
   LENgth: <NRx>
   TIMe: <NRx>
   ? XINcr: <NR3>

## Triggering Commands

TR?

TRMain <link>:<arg>
   ALEvel: <NRx>
   ANLevel: <NRx>,{VOLts|DIVS}
   COUpling: {AC|ACLf|ACHf|DCHf|ACNoise|
          DCNoise|DC}
   MODe: {AUTO|AUTOLevel|NORmal}
   SLOpe: {PLUs|MINUs}
   SOUrce <qstring>
   ? STAtus {TRG|NOTrg}
   TIHoldoff: <NRx>

TRWin <link>:<arg>
   ALEvel: <NRx>
   COUpling: {AC|ACLf|ACHf|DCHf|ACNoise|
          DCNoise|DC}
   EVHoldoff: <NRx>
   MODe: {AUTOLevel|NORmal}
   NLEvel: <NRx>,{VOLts|DIVS}
   SLOpe: {PLUs|MINUs}
   SOUrce: <qstring>
   ? STAtus: {TRG|NOTrg}
   TIHoldoff: <NRx>

WTMode {MAin|EVHoldoff|TIHoldoff}

## Waveform/Settings Commands

ADJtrace<ui> <link>:<arg>
   HMAg: <NRx>
   HPOsition: <NRx>
   HVPosition: <NRx>
   HVSize: <NRx>
   PANzoom: {ON|OFF}
   TRSep: <NRx>
   VPOsition: <NRx>
   VSIze: <NRx>

ADJtrace[<ui>]?

AVG {ON|OFF}

CLEar {TRAce<ui>|ALL}      (Set-only)

DELete {STO<ui>|FPS<ui>}   (Set-only)
   ALL: {STO|FPS}        (Set-only)

ENV {ON|OFF}

FPSList?

FPSNum?

NAVg <NRx>

NENV <NRx>

NVRam?

RECall {FPS<ui>|FPSNext}   (Set-only)

REMove {TRAce<ui>|ALL}   (Set-only)

SELect TRAce<ui>

SETSeq {ON|OFF}

STORe {FPS<ui> |        (Set-only)
   TRAce<ui>: STO<ui>}  (Set-only)

STOList?

STONum?

TRAce<ui> <link>:<arg>
   ACCumulate: {ON|OFF}
   ? ACState: {ENHanced|NENHanced}
   DEScription: <qstring>
   GRLocation: {UPPer|LOWer}
   GRType: LINear
   ? WFMCalc: {FASt|HIPrec}
   ? XUNit: {AMPS|DIVS|OHMs|SEConds|
          VOLts|WATs}
   ? YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}

TRAce?

TRANUm?

WFMScaling {FORce|OPTional}

## Tektronix 11401/11402
### Alphabetic Command Summary

### Key

| | | |
|---|---|---|
| < > | ::= | Defined item |
| { } | ::= | One item from group required |
| [ ] | ::= | Optional item(s) |
| ( ) | ::= | Grouped items |
| ... | ::= | Group can be repeated |
| \| | ::= | Exclusive or |
| FPS | ::= | Front Panel Settings |
| <NR1> | ::= | signed integer |
| <NR2> | ::= | floating point, no exponent |
| <NR3> | ::= | floating point with exponent |
| <NRx> | ::= | {<NR1>|<NR2>|<NR3>} |
| <ui> | ::= | unsigned integer |
| <slot> | ::= | {L|C|R} |
| <qstring> | ::= | quoted string |
| <bblock> | ::= | Tek "Codes&Formats" binary block data |
| TYPe | | Spelling of header or link; minimum spelling in CAPs |
| SINgle | | Spelling of argument; minimum spelling in CAPs |
| ? | | Query-only header or link |

Command headers are flush-left, followed by their syntax. Links are indented.
Query-only links are indicated with a leading ?; the "argument" after the colon shows the form of the response.

## A – B

ABBwfmpre {ON|OFF}

ABStouch { <NRx>,<NRx> |CLEar}

ADJtrace<ui> <link>:<arg>
  HMAg: <NRx>
  HPOsition: <NRx>
  HVPosition:<NRx>
  HVSize: <NRx>
  PANzoom: {ON|OFF}
  TRSep: <NRx>
  VPOsition: <NRx>
  VSIze: <NRx>
ADJtrace[<ui>]

AUTOSet {[<link>:]<arg>}
  HORiz {PERiod|OFF}
  STARt                          (Set–only)
  UNDO                           (Set–only)
  VERt: {ECL|PP|TTL|OFF}

AVG {ON|OFF}

BASeline <NRx>

BYT.or {LSB|MSB}

## C

CALStatus?

CCAlconstants <ui> <NRx>

CH<slot><ui> <link>:<arg>
  AMPoffset: <NRx>
  BW: <NRx>
  COUpling: {AC|DC|OFF}
  IMPedance: <NRx>
  MNSCoupling: {AC|DC|VC|OFF}
  MNSOffset: <NRx>
  ? MNSProbe: <qstring>
  OFFSet: <NRx>
  PLSCoupling: {AC|DC|VC|OFF}
  PLSOffset: <NRx>
  ? PLSProbe: <qstring>
  ? PROBe: <qstring>
  PROTect: ON|OFF
  SENsitivity: <NRx>
  ? UNIts: <qstring>
  VCOffset: <NRx>

CH[<slot>]?

CLEar {TRAce<ui> |ALL}          (Set–only)

COMpare {ON|OFF}

CONDacq <link>:<arg>
  FILI: <NRx>
  ? REMAining: <NR1>
  TYPe: {FILI|AVG|ENV|BOTh|SINgle|
          CONTinuous}

CONFig?

COPy {[<link>:]<arg>}
  ABOrt                          (Set–only)
  FORMat: {DRAft|HIRes|REDuced}
  PRInter: {PIN8|PIN24}
  STARt                          (Set–only)

CURSor <link>:<arg>
   REAdout: {ON|OFF}
   REFErence  TRAce<ui>
   TYPe: {PAIred|SPLit|VBArs|HBArs}
   ? XUNit: {AMPS|DIVS|OHMs|SEConds|
               VOLts|WATs}
   ? YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}

CURVe { <bblock> | (<NRx>[,<NRx>]...)}

# D

DAInt {WHOle|SINgle}

DATE <qstring>  = "dd-mmm-yy"

DEBug <link>:<arg>
   GPIb: {ON|OFF}
   RS232: {ON|OFF}

DEF <qstring>,<qstring>          *(Set-only)*

DELete {STO<ui>|FPS<ui>}      *(Set-only)*
   ALL: {STO|FPS}                *(Set-only)*

`g?

DIGitizer {RUN|STOP}

DISPlay <link>:<arg>
   GRAticle: {DUAl|SINgle}
   INTensity: <NRx>
   MODe: {DOTs|VECtors}

DISTal <NRx>

DLYtrace TRAce<ui>

DOT1Abs, DOT2Abs <link>:<arg>
   PCTg: <NRx>
   XCOord: <NRx>
   XDIv: <NRx>
   ? XQUal: {EQ|LT|GT|UN}
   ? YCOord: <NR3>
   ? YDIv: <NR3>
   ? YQUal: {EQ|LT|GT|UN}

DOT1Rel, DOT2Rel <link>:<arg> *(Set-only)*
   PCTg: <NRx>              *(Set-only)*
   XCOord: <NRx>          *(Set-only)*
   XDIv: <NRx>              *(Set-only)*

DSYmenu?

# E – F

ENCdg <link>:<arg>
   SET: {ASCii|BINary}
   WAVfrm: {ASCii|BINary}

ENV {ON|OFF}

EVENT?

FEOi                              *(Set-only)*

FPAnel {ON|OFF}

FPSList?

FPSNum?

FPUpdate {ON|OFF|NEVer}

# H – I

H1Bar, H2Bar <link>:<arg>
   YCOord: <NRx>
   YDIv: <NRx>

ID?

IDProbe?

INIt                              *(Set-only)*

INPut STO<ui>

# L – M

LCAlconstants <ui>:<NRx>

LMZone <NRx>

LONgform {ON|OFF}

MAINPos <NRx>

MCAlconstants <ui>:<NRx>

MEAS?

<meas> ::= {CROss|DELAy|FALltime|FREq|
   MAX|MEAN|MID|MIN|PDElay|PERiod|PP|
   RISetime|RMS|TTRig|WIDth|YTEnergy|
   YTMns_area|YTPls_area}

<meas>?

MESial <NRx>

MSLIst { <meas> |EMPty}

MSLOpe {PLUs|MINUs}

MSNum?

MSYs {ON|OFF}

MTRack {ON|OFF}

## N – O – P

NAVg <NRx>

NENV <NRx>

NVRam?

OUTput {STO<ui>|TRAce<ui>}

PATh {ON|OFF}

PIVersion?

POWeron?

PROBe {NT|NTAuto|SETSeq}

PROXimal <NRx>

## R

RCAlconstants <ui>:<NRx>

RECall {FPS<ui>|FPSNext}                    (Set-only)

REFLevel <NRx>

REFSet <link>:<arg>
  CURRent: <meas>                  (Set-only)
  <meas>: <NRx>

REMove {TRAce<ui>|ALL}                      (Set-only)

RMZone <NRx>

RQS {ON|OFF}

RS232 <link>:<arg>
  BAUd: <NRx>
  DELAy: <NRx>
  ECHo: {ON|OFF}
  EOL: {CR|CRLf|LF|LFCr}
  FLAgging: {SOFt|HARd|OFF}
  PARity: {ODD|EVEN|NONe}
  STOPBits: <NRx>
  VERBose: {ON|OFF}

## S

SELect TRAce<ui>

SELFcal {[<link>:]<arg>}
  FORce                            (Set-only)
  MODe: {AUTO|MANual}

SET <bblock>                                (Set-only)

SET?

SETSeq {ON|OFF}

SNRatio <NRx>

SPEaker {ON|OFF}

SRQMask <link>:<arg>
  ABStouch:{ON|OFF}
  CALDue: {ON|OFF}
  CMDerr: {ON|OFF}
  EXErr: {ON|OFF}
  EXWarn: {ON|OFF}
  IDProbe: {ON|OFF}
  INErr: {ON|OFF}
  INWarn: {ON|OFF}
  OPCmpl: {ON|OFF}
  USEr: {ON|OFF}

STByte?

STORe FPS<ui>                               (Set-only)
  TRAce<ui>: STO<ui>               (Set-only)

STOList?

STONum?

## T

TBMain <link>:<arg>
  LENgth: <NRx>
  TIMe: <NRx>
  ? XINcr: <NR3>

TBWin <link>:<arg>
  LENgth: <NRx>
  TIMe: <NRx>
  ? XINcr: <NR3>

TESt [XTNd]                                 (Set-only)

TEXt {[<link>:]<arg>}                       (Set-only)
  CLEar                            (Set-only)
  STRing: <qstring>                (Set-only)
  X: <NRx>                         (Set-only)
  Y: <NRx>                         (Set-only)

TIMe <qstring> = "hh:mm:ss"

TOPline <NRx>

TR?

TRAce<ui> <link>:<arg>
  ACCumulate: {ON|OFF}
  ? ACState: {ENHanced|NENHanced}
  DEScription: <qstring>
  GRLocation: {UPPer|LOWer}
  GRType: LINear
  ? WFMCalc: {FASt|HIPrec}
  ? XUNit: {AMPS|DIVS|OHMs|SEConds|
            VOLts|WATs}
  ? YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}

TRAce[<ui>]?

TRANUm?

TRMain <link>:<arg>
  ALEvel: <NRx>
  ANLevel: <NRx>,{VOLts|DIVS}
  COUpling: {AC|ACLf|ACHf|DCHf|ACNoise|
          DCNoise|DC}
  MODe: {AUTO|AUTOLevel|NORmal}
  SLOpe: {PLUs|MINUs}
  SOUrce: <qstring>
  ? STAtus: {TRG|NOTrg}
  TIHoldoff: <NRx>

TRWin <link>:<arg>
  ALEvel: <NRx>
  COUpling: {AC|ACLf|ACHf|DCHf|ACNoise|
          DCNoise|DC}
  EVHoldoff: <NRx>
  MODe: {AUTOLevel|NORmal}
  NLEvel: <NRx>,{VOLts|DIVS}
  SLOpe: {PLUs|MINUs}
  SOUrce: <qstring>
  ? STAtus: {TRG|NOTrg}
  TIHoldoff: <NRx>

TTAverage <NRx>

## U – V – W

UID <link>:<arg>
  CENter: <qstring>
  LEFt: <qstring>
  MAIn: <qstring>
  RIGht: <qstring>

UNDEF {<qstring>|ALL}     *(Set-only)*

UPTime?

V1Bar, V2Bar <link>:<arg>
  XCOord: <NRx>
  XDIv: <NRx>

WAVfrm?

WFMpre <link>:<arg>
  ACState: {ENHanced|NENhanced}
  ? BIT/nr: 16
  ? BN.fmt: RI
  ? BYT/nr: 2
  ? BYT.or: {LSB|MSB}
  ? CRVchk: {CHKsm0|NONe|NULl}
  ? ENCdg: {ASCii|BINary}
  NR.pt: <NRx>
  PT.fmt: {ENV|Y|XY}
  ? WFId: {STO<ui>|TRAce<ui>}
  XINcr: <NRx>
  ? XMUlt: <NR3>
  ? XUNit: {AMPS|DIVS|OHMs|SEConds|
          VOLts|WATs}
  XZEro: <NRx>
  YMUlt: <NRx>
  YUNit: {AMPS|DIVS|OHMs|VOLts|WATs}
  YZEro: <NRx>

WFMScaling {FORce|OPTional}

WIN1Pos <NRx>

WIN2Pos <NRx>

WTMode {MAIn|EVHoldoff|TIHoldoff}

# Tektronix®

COMMITTED TO EXCELLENCE

## ASCII & GPIB CODE CHART

| B7 B6 B5 / BITS / B4 B3 B2 B1 | CONTROL | | NUMBERS SYMBOLS | | UPPER CASE | | LOWER CASE | |
|---|---|---|---|---|---|---|---|---|
| 0 0 0 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 0 0 0 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0 0 1 0 | STX | DC2 | " | 2 | B | R | b | r |
| 0 0 1 1 | ETX | DC3 | # | 3 | C | S | c | s |
| 0 1 0 0 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0 1 0 1 | ENQ | NAK | % | 5 | E | U | e | u |
| 0 1 1 0 | ACK | SYN | & | 6 | F | V | f | v |
| 0 1 1 1 | BEL | ETB | ' | 7 | G | W | g | w |
| 1 0 0 0 | BS | CAN | ( | 8 | H | X | h | x |
| 1 0 0 1 | HT | EM | ) | 9 | I | Y | i | y |
| 1 0 1 0 | LF | SUB | * | : | J | Z | j | z |
| 1 0 1 1 | VT | ESC | + | ; | K | [ | k | { |
| 1 1 0 0 | FF | FS | , | < | L | \ | l | \| |
| 1 1 0 1 | CR | GS | - | = | M | ] | m | } |
| 1 1 1 0 | SO | RS | . | > | N | ^ | n | ~ |
| 1 1 1 1 | SI | US | / | ? | O | — | o | DEL (RUBOUT) |
| | ADDRESSED COMMANDS | UNIVERSAL COMMANDS | LISTEN ADDRESSES | | TALK ADDRESSES | | SECONDARY ADDRESSES OR COMMANDS (PPE) (PPD) | |

† on some keyboards or systems

## KEY

```
octal  25      PPU     GPIB code
       NAK             ASCII character
hex    15      21      decimal
```

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077