

## Errata

**Title & Document Type:** 54520A/54540A Series Oscilloscope  
Programmer's Quick Reference Guide

**Manual Part Number:** 54542-97037

**Revision Date:** November 1, 1995

---

### HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

### About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

### Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent Test & Measurement website:

[www.tm.agilent.com](http://www.tm.agilent.com)

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

---

# Programmer's Quick Reference Guide

Publication number 54542-97037  
First edition, November 1995

For Safety information, Warranties, and Regulatory  
information, see the pages at the end of this book.

© Copyright Hewlett-Packard Company 1993-1995  
All Rights Reserved

---

## HP 54520A and 54540A Series Oscilloscopes

---

# Introduction

The Quick Reference Guide lists the commands and queries with their corresponding arguments and returned formats. The arguments for each command list the minimum argument required. The part of the command or query listed in uppercase letters refers to the short form of that command or query. The long form is the combination of uppercase and lowercase letters.

## Conventions

The following conventions are used in this guide:

- <> Angular brackets enclose words or characters that symbolize a program code parameter or an HP-IB command.
- ::= "is defined as." For example, <A> ::= <B> indicates that <A> can be replaced by <B> in any statement containing <A>.
- | "or." Indicates a choice of one element from a list. For example, <A> | <B> indicates <A> or <B> but not both.
- ... An ellipsis (trailing dots) indicate that the preceding element may be repeated one or more times.
- [ ] Square brackets indicate that the enclosed items are optional.
- { } When several items are enclosed by braces, one, and only one of these elements may be selected.

## Suffix Multipliers

The suffix multipliers available for arguments are:

EX ::= 1E18	M ::= 1E-3
PE ::= 1E15	U ::= 1E-6
T ::= 1E12	N ::= 1E-9
G ::= 1E9	P ::= 1E-12
MA ::= 1E6	F ::= 1E-15
K ::= 1E3	A ::= 1E-18

For more information on specific commands or queries, refer to the *HP 54520 and 54540 Series Oscilloscopes Programmer's Reference*.

---

**\*CLS**

Command \*CLS

---

**\*DMC**

Command \*DMC <ascii\_string>, <block\_data>  
 Where: <ascii\_string> ::= a quoted ascii string  
 <block\_data> ::= definite block data in IEEE 488.2 # format

---

**\*EMC**

Command \*EMC {{OFF| 0} | {ON |1}}  
 Query \*EMC?  
 Returned Format {0| 1}<NL>

---

**\*ESE**

Command \*ESE <mask\_argument>  
 Query \*ESE?  
 Returned Format <mask\_argument><NL>  
 Where: <mask\_argument> ::= integer, 0 to 255

---

**\*ESR?**

Query \*ESR?  
 Returned Format <status><NL>  
 Where: <status> ::= integer, 0 to 255

---

**\*GMC?**

Query \*GMC? <ascii\_string>  
 Returned Format <block\_data><NL>  
 Where: <ascii\_string> ::= a quoted string  
 <block\_data> ::= definite block data in # format

---

---

**\*IDN?**

Query  
Returned Format  
Where:

\*IDN?  
HEWLETT-PACKARD, 545XXA, YYYYYYYYYY, ZZ, ZZ, ZZ.ZZ, ZZ.ZZ, ZZ.ZZ<NL>  
<XXA> ::= model number is 20A (HP 54520A), 22A (HP 54522A),  
40A (HP 54540A), or 42A (HP 54542A)  
<YYYYYYYYYY> ::= the serial number of the instrument  
<ZZ.ZZ> ::= the software revision of the software modules  
(Boot ROM, Flash ROM version of Boot ROM, System, Keyboard  
ROM). 00.00 = not installed

---

**\*LMC**

Query  
Returned Format  
Where:

\*LMC?  
<ascii\_string><NL>  
<ascii\_string> ::= string list separated by commas

---

**\*LRN?**

Query  
Returned Format  
Where:

\*LRN?  
:SYSTEM:SETup <setup><NL>  
<setup> ::= #800002048<learn\_string>  
<learn\_string> ::= 2048 bytes in length

---

**\*OPC**

Command  
Query  
Returned Format

\*OPC  
\*OPC?  
1<NL>

---

**\*OPT ?**

Query  
Returned Format

\*OPT?  
0<NL>



---

**\*TST?**

Query  
Returned Format  
Where:

\*TST?  
<result><NL>  
<result> ::= 0 or non-zero value. 0 indicates the test passed. non-zero indicates the test failed

---

**\*WAI**

Command

\*WAI

---

**:ACQuire:COMPlate**

Command  
Query  
Returned Format  
Where:

:ACQuire:COMPlate <complete\_argument>  
:ACQuire:COMPlate?  
[:ACQuire:COMPlate] <complete\_argument><NL>  
<complete\_argument> ::= integer, 0 to 100 percent

---

**:ACQuire:COUNt**

Query  
Returned Format  
Where:

:ACQuire:COUNt <count\_argument>  
:ACQuire:COUNt?  
[:ACQuire:COUNt] <count\_argument><NL>  
<count\_argument> ::= integer, 1 to 2048 (depending on the acquisition type)

---

**:ACQuire:POINts**

Command  
Query  
Returned Format  
Where:

:ACQuire:POINts <points\_argument>  
:ACQuire:POINts?  
[:ACQuire:POINts] <points\_argument><NL>  
<points\_argument> ::= integer, 500 in repetitive mode, 512, 1024, 2048, 4196, 8192, 16384, or 32768 in the real-time mode (sequential mode off), or 4 through 32768 in the real-time mode (sequential mode on)

---

---

**:ACQUIRE:TYPE**

**Command** :ACQUIRE:TYPE {NORMAL | AVERAGE | ENVELOPE | PDETECT |  
 RAWData[,<length>][, <acquisitions>]][, {NORMAL | AVERAGE |  
 ENVELOPE }]

**Query** :ACQUIRE:TYPE?

**Returned Format** [:ACQUIRE:TYPE] {NORMAL | AVERAGE | ENVELOPE | PDETECT |  
 RAWData, <length>, <acquisitions><NL>

**Where:** <length> ::= integer, 4 to 32768  
 <acquisitions> ::= dependent on length of acquisitions and  
 buffer size

---

**:AUTOSCALE**

**Command** :AUTOSCALE

**Query** :AUTOSCALE?

**Returned Format** [:AUTOSCALE] {0|1}<NL>

---

**:BEEPER**

**Command** :BEEPER [{OFF | 0} | {ON | 1}]

**Query** :BEEPER?

**Returned Format** [:BEEPER] {0|1}<NL>

---

**:BLANK**

**Command** :BLANK <display>

**Where:** <display> ::= {CHANNEL<n> | FUNCTION{1|2|3|4} |  
 WMEMORY{1 | 2 | 3 | 4} | PMEMORY{1 | 2}}

<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

**:BNC**

**Command** :BNC {PROBE | TRIGGER}

**Query** :BNC?

**Returned Format** [:BNC] {PROBE | TRIGGER}<NL>

---



---

### **:CALibrate:DATA:ASCii?**

**Query** :CALibrate:DATA:ASCii?  
**Returned Format** [:CALibrate:DATA:ASCii] <data>,<data>,...<NL>  
**Where:** <data> ::= calibration data

---

### **:CALibrate:SETup?**

**Query** :CALibrate:SETup?  
**Returned Format** :CALibrate:TNULl]  
<null\_value\_n>,<null\_value\_n>,<null\_value\_n><NL>  
**Where:** <null\_value\_n> ::= exponential, channel 1 to channel<n> skew,  
where n = 2 (HP 54520A/54522A) or 2 through 4  
(HP 54540A/54542A) in format

---

### **:CALibrate:TNULl**

**Command** :CALibrate:TNULl <null\_value\_n>  
**Query** :CALibrate:TNULl?  
**Returned Format** [:CALibrate:TNULl] <null\_value\_n><NL>  
**Where:** <null\_value\_n> ::= exponential, channel 1 to channel<n> skew,  
where n = 2 (HP 54520A/54522A) or 2 through 4  
(HP 54540A/54542A)

---

### **:CHANnel<n>:COUPling**

**Command** :CHANnel<n>:COUPling {AC | DC | DCFifty}  
**Query** :CHANnel<n>:COUPling?  
**Returned Format** [:CHANnel<n>:COUPling] {AC | DC | DCFifty}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
(HP 54540A/54542A)

---

### **:CHANnel<n>:DISPLay**

**Command** :CHANnel<n>:DISPLay {{OFF | 0} | {ON | 1}}  
**Query** :CHANnel<n>:DISPLay?  
**Returned Format** [:CHANnel<n>:DISPLay {0 | 1}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
(HP 54540A/54542A)

---

---

**:CHANnel<n>:ECL**

**Command** :CHANnel<n>:ECL  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

**:CHANnel<n>:HFReject**

**Command** :CHANnel<n>:HFReject {{OFF | 0} | {ON | 1}}  
**Query** :CHANnel<n>:HFReject?  
**Returned Format** [:CHANnel<n>:HFReject] {0 | 1}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

**:CHANnel<n>:LFReject**

**Command** :CHANnel<n>:LFReject {{OFF | 0} | {ON | 1}}  
**Query** :CHANnel<n>:LFReject?  
**Returned Format** [:CHANnel<n>:LFReject] {0 | 1}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

**:CHANnel<n>:OFFSet**

**Command** :CHANnel<n>:OFFSet <offset\_argument>  
**Query** :CHANnel<n>:OFFSet?  
**Returned Format** [:CHANnel<n>:OFFSet] <offset\_argument><NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)  
 <offset\_argument> ::= exponential, offset value in volts

---

---

## :CHANnel<n>:PROBe

**Command** :CHANnel<n>:PROBe <probe\_argument>  
**Query** :CHANnel<n>:PROBe?  
**Returned Format** [:CHANnel<n>:PROBe] <probe\_argument><NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
           (HP 54540A/54542A)  
           <probe\_argument> ::= exponential, 0.9 to 1000.0

---

## :CHANnel<n>:RANGe

**Command** :CHANnel<n>:RANGe <range\_argument>  
**Query** :CHANnel<n>:RANGe?  
**Returned Format** [:CHANnel<n>:RANGe] <range\_argument><NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
           (HP 54540A/54542A)  
           <range\_argument> ::= exponential, full-scale range value

---

## :CHANnel<n>:SETup

**Query** :CHANnel<n>:SETup?  
**Returned Format** :CHAN<n>:COUP {AC|DC|DCF};  
           DISP {0 | 1};  
           HFR {0 | 1};  
           LFR {0 | 1};  
           OFFS <offset\_argument>;  
           PROB <probe\_argument>;  
           RANG<range\_argument><NL>  
**Where:** <range\_argument> ::= exponential, full-scale range value  
           <offset\_argument> ::= exponential, offset value in volts  
           <probe\_argument> ::= exponential, 0.9 to 1000.0  
           <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
           (HP 54540A/54542A)

---

## :CHANnel<n>:TTL

**Command** :CHANnel<n>:TTL  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
           (HP 54540A/54542A)

---

---

**:DIGitize****Command****Where:**

```
:DIGitize CHANnel<n>[,CHANnel<n>[,CHANnel<n>[,CHANnel<n>]]]
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

**:DISK:CDIRectory****Command****Where:**

```
:DISK:CDirectory <directory_name>
<directory_name> ::= 1 to 65 character quoted ASCII string
```

---

**:DISK:DELEte****Command****Where:**

```
:DISK:DELEte <file_name>
<file_name> ::= 1 to 8 character quoted ASCII string, if DOS,
can have a 0 to 3 character extension
```

---

**:DISK:DIRectory?****Query****Returned Format****Where:**

```
:DISK:DIRectory?
[:DISK:DIRectory] <number_of_files><cr><lf><directory>
<number_of_files> ::= integer, number of files (that follow)
in the root directory
<directory> ::= {<filename>,<ext>,<date>,<time>,<size>,<description><cr><lf>...}
<ext> ::= {.SETup | .WAVEform | .TEXT | .PIXel}
<date> ::= DDMMYY
<time> ::= HH:MM:SS
<size> ::= an integer
<description> ::= LIF format: Model + C1 + HP545XX + Source,
DOS format: Model = DOS file (no description)
```

---

**:DISK:FORMat****Command****Where:**

```
:DISK:FORMat <format_type>
<format_type> ::= {DOS | LIF}
```

---

---

## :DISK:LOAD

Command  
Where:

```
:DISK:LOAD <file_name>,<destination> [,<format>]
<filename> ::= quoted ascii string  DOS compatible filename.
  1 to 8 character ASCII string, if DOS, may have a 0 to 3
character extension
  Either .wav or .txt may be used as a suffix after the
  filename.  If no file suffix is specified, the default is
  .wav.
<destination> ::= {WMEemory {1 | 2 | 3 | 4 } |
  SETup{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9} |
  PMEemory{ 1 | 2}}
<format>::= {TEXT | INTernal}
```

---

## :DISK:MDIRectory

Command  
Where:

```
:DISK:MDirectory <directory_name>
<directory_name> ::= 1 to 65 character quoted ASCII string
```

---

## :DISK:PWD?

Query  
Returned Format:

```
:DISK:PWD?
[:DISK:PWD] <present_working_directory>
```

---

## :DISK:SIMage

Command

Where

```
:DISK:SIMage <file_name> [|,<format>| [|,<compression>|
|,<rendering>]]]
<filename>::= 1 to 8 character quoted ASCII string, if DOS,
  can have a 0 to 3 character extension
<format>::={TIFF | PCX | EPS}
<compression>::={ON | OFF}
<rendering>::={BW | GRAY [ GREen]}
```

---

**:DISK:STORe****Command**`:DISK:STORe <source>,<file_name>[,<format>]`**Where:**

```

<source> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4} |
  SETup {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9} |
  PMEMory {0 | 1 | 2 | 3 | 4}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
<filename> ::= a descriptive name of the file up to 8
      characters long
<format> ::= {INTernal | TEXT [,{<XYPairs> | <YVALues> |
      <VERBose>}]}

```

---

**:DISPlay:COLumn****Command**`:DISPlay:COLumn <column_number>`**Query**`:DISPlay:COLumn?`**Returned Format**`[ :DISPlay:COLumn ] <column_number><NL>`**Where:**`<column_number> ::= integer, 0 through 78`

---

**:DISPlay:CONNect****Command**`:DISPlay:CONNect {{OFF | 0} | {ON | 1}}`**Query**`:DISPlay:CONNect?`**Returned Format**`[ :DISPlay:CONNect ] {0 | 1}<NL>`

---

**:DISPlay:DATA****Command**`:DISPlay:DATA <binary_block>`**Query**`:DISPlay:DATA?`**Returned Format**`[ :DISPlay:DATA ] #800016576<16576 bytes of binary data><NL>`**Where:**`<binary_block> ::= definite block data in # format`

---

**:DISPlay:FORMat****Command**`:DISPlay:FORMat {1 | 2 | 4}`**Query**`:DISPlay:FORMat?`**Returned Format**`[ :DISPlay:FORMat ] {1 | 2 | 4}<NL>`

---

## **:DISPlay:GRATicule**

**Command** :DISPlay:GRATicule {AXES | GRID | FRAME | OFF}  
**Query** :DISPlay:GRATicule?  
**Returned Format** [:DISPlay:GRATicule] {AXES | GRID | FRAME | OFF}<NL>

---

## **:DISPlay:INVerse**

**Command** :DISPlay:INVerse {{OFF | 0} | {ON | 1}}  
**Query** :DISPlay:INVerse?  
**Returned Format** [:DISPlay:INVerse] {0 | 1}<NL>

---

## **:DISPlay:LINE**

**Command** :DISPlay:LINE <ascii\_string>  
**Where:** <ascii\_string> ::= any series of ascii characters enclosed in quotes

---

## **:DISPlay:MASK**

**Command** :DISPlay:MASK <mask\_argument>  
**Query** :DISPlay:MASK?  
**Returned Format** [:DISPlay:MASK] <mask\_argument><NL>  
**Where:** <mask\_argument> ::= integer, 0 through 255

---

## **:DISPlay:PERStistence**

**Command** :DISPlay:PERStistence {SINGle | INFinite | 0 | {0.5 to 10} | 11}}  
**Query** :DISPlay:PERStistence?  
**Returned Format** [:DISPlay:PERStistence] <value><NL>  
**Where:** <value> ::= exponential, {0 (minimum) | {0.5 to 10} | 11 (infinite)} in the repetitive mode, {SINGle | INFinite} in real-time mode

---

---

## **:DISPlay:ROW**

**Command** :DISPlay:ROW <row\_number>  
**Query** :DISPlay:ROW?  
**Returned Format** [:DISPlay:ROW] <row\_number><NL>  
**Where:** <row\_number> ::= integer, 0 through 24

---

## **:DISPlay:SCReen**

**Command** :DISPlay:SCReen {{OFF | 0} | {ON | 1}}  
**Query** :DISPlay:SCReen?  
**Returned Format** [:DISPlay:SCReen] {0 | 1}<NL>

---

## **:DISPlay:SETup?**

**Query** :DISPlay:SETup?  
**Returned Format** :DISP:COL <column\_number>;  
 CONN {0 | 1};  
 FORM {1 | 2 | 4};  
 GRAT {AXES | FRAM | GRID | OFF};  
 INV {0 | 1};  
 MASK <mask\_argument>;  
 PERS <pers\_argument>;  
 ROW <row\_number>;  
 SCR {0 | 1};  
 SOUR PMEM {0 | 1 | 2};  
 MARK {0 | 1}<NL>

**Where:** <column\_number> ::= integer, 0 through 78  
 <mask\_argument> ::= integer, 0 to 255  
 <pers\_argument> ::= exponential, {0 | .5 to 10 | 11} in the  
 repetitive mode, {SINGLE | INFinite} in the real-time mode  
 <row\_number> ::= integer, 0 to 24

---

## **:DISPlay:SOURce**

**Command** :DISPlay:SOURce PMEMory{0 | 1 | 2 }  
**Query** :DISPlay:SOURce?  
**Returned Format** [:DISPlay:SOURce] PMEMory{0 | 1 | 2 }<NL>

---



---

### **:DISPlay:STRing**

**Command** :DISPlay:STRing <string\_argument>  
**Where:** <string\_argument> ::= text string up to 90 characters

---

### **:DISPlay:TEXT**

**Command** :DISPlay:TEXT BLANK

---

### **:DISPlay:{MARKer|TMARker|VMARker}**

**Command** :DISPlay:{MARKer|TMARker|VMARker} {{OFF | 0} | {ON | 1}}  
**Query** :DISPlay:MARKer|TMARker|VMARker?  
**Returned Format** [:DISPlay:MARKer|TMARker|VMARker] {0 | 1}<NL>

---

### **:ERASe**

**Command** :ERASe {PMEMory0<pmemory\_num>}  
**Where:** <pmemory\_num> ::= integer, 1 or 2 @MODULE RULE =

---

### **:EXTernal:COUPling**

**Command** :EXTernal:COUPling {AC | DC | DCFifty}  
**Query** :EXTernal:COUPling?  
**Returned Format** [:EXTernal:COUPling] {AC | DC | DCFifty}<NL>

---

### **:EXTernal:HFReject**

**Command** :EXTernal:HFReject {{OFF | 0} | {ON | 1}}  
**Query** :EXTernal:HFReject?  
**Returned Format** [:EXTernal:HFReject] {0 | 1}<NL>

---

---

## **:EXTErnal:LFReject**

Command :EXTErnal:LFReject {{OFF | 0} | {ON | 1}}

Query :EXTErnal:LFReject?

Returned Format [:EXTErnalLFReject] {0 | 1}<NL>

---

## **:EXTErnal:PROBe**

Command :EXTErnal:PROBe <probe\_argument>

Query :EXTErnal:PROBe?

Returned Format [:EXTErnal:PROBe] <probe\_argument><NL>

<probe\_argument> ::= exponential, 0.9 to 1000.0

---

## **:EXTErnal:RANGe**

Command :EXTErnal:RANGe <range\_argument>

Query :EXTErnal:RANGe?

Returned Format [:EXTErnal:RANGe] <range\_argument><NL>

<range\_argument> ::= exponential, full-scale range value

---

## **:EXTErnal:SETup**

Query :EXTErnal:SETup?

Returned Format :EXT:COUP {AC|DC|DCF};  
 HFR {0 | 1};  
 LFR {0 | 1};  
 PROB <probe\_argument>;  
 RANG<range\_argument><NL>

Where: <range\_argument> ::= exponential, full-scale range value  
 <probe\_argument> ::= exponential, 0.9 to 1000.0

---

## **:FUNCTion{1 | 2 | 3 | 4}:ADD**

Command :FUNCTion{1 | 2 | 3 | 4}:ADD <operand>,<operand>

Where: <operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}

<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

---

### **:FUNction{1 | 2 | 3 | 4}:DIFF**

**Command  
Where:**

```
:FUNction{1 | 2 | 3 | 4}:DIFF <operand>
<operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

### **:FUNction{1 | 2 | 3 | 4}:DISPLay**

**Command  
Query  
Returned Format**

```
:FUNction{1 | 2 | 3 | 4}:DISPlay {{OFF | 0} | {ON | 1}}
:FUNction{1 | 2 | 3 | 4}:DISPlay?
[:FUNction{1 | 2 | 3 | 4}[:DISPlay] {0 | 1}<NL>
```

---

### **:FUNction{1 | 2 | 3 | 4}:FFT**

**Command  
Where:**

```
:FUNction{1 | 2 | 3 | 4}:FFT <operand>
<operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

### **:FUNction{1 | 2 | 3 | 4}:FREQuency**

**Command  
Query  
Returned Format  
Where:**

```
:FUNction{1 | 2 | 3 | 4}:FREQuency <frequency_argument>
:FUNction{1 | 2 | 3 | 4}:FREQuency?
[:FUNction{1 | 2 | 3 | 4}:FREQuency] <frequency_argument><NL>
<frequency_argument> ::= exponential, center frequency from
      0 Hz to 1.5X of frequency span
```

---

### **:FUNction{1 | 2 | 3 | 4}:INTegrate**

**Command  
Where:**

```
:FUNction{1 | 2 | 3 | 4}:INTegrate <operand>
<operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

## **:FUNction{1 | 2 | 3 | 4}:INVert**

**Command** :FUNction{1 | 2 | 3 | 4}:INVert <operand>  
**Where:** <operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

## **:FUNction{1 | 2 | 3 | 4}:LEVel**

**Command** :FUNction{1 | 2 | 3 | 4}:LEVel <level\_argument>  
**Query** :FUNction{1 | 2 | 3 | 4}:LEVel?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:LEVel] <level\_argument><NL>  
**Where:** <level\_argument> ::= exponential, level value 0 to + or -  
 600 dbm

---

## **:FUNction{1 | 2 | 3 | 4}:MAGNify**

**Command** :FUNction{1 | 2 | 3 | 4}:MAGNify {{OFF | 0} | {ON | 1}}  
**Query** :FUNction{1 | 2 | 3 | 4}:MAGNify?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:MAGNify] {0 | 1}<NL>

---

## **:FUNction{1|2|3|4}:MODE?**

**Query** :FUNction{1 | 2 | 3 | 4}:MODE?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:MODE]  
 <operation>,<operand>[,<operand>]  
**Where:** <operation> ::= {ADD | SUBTract | MULTiply | VERSus | ONLY |  
 INVert | INTegrate | DIFF | FFT}  
 <operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

## **:FUNction{1 | 2 | 3 | 4}:MULTiply**

**Command** :FUNction{1 | 2 | 3 | 4}:MULTiply <operand>,<operand>  
**Where:** <operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

---

## **:FUNction{1 | 2 | 3 | 4}:OFFSet**

**Command** :FUNction{1 | 2 | 3 | 4}:OFFSet <offset\_argument>  
**Query** :FUNction{1 | 2 | 3 | 4}:OFFSet?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:OFFSet] <offset\_argument><NL>  
**Where:** <offset\_argument> ::= exponential, offset value of 0 to  $\pm$ voltage full scale, or 0 to  $\pm$ 200 dbm for FFT

---

## **:FUNction{1 | 2 | 3 | 4}:ONLY**

**Command** :FUNction{1 | 2 | 3 | 4}:ONLY <operand>  
**Where:** <operand> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4 (HP 54540A/54542A)

---

## **:FUNction{1 | 2 | 3 | 4}:PEAK**

**Command** :FUNction{1 | 2 | 3 | 4}:PEAK <peak1\_number>,<peak2\_number>  
**Query** :FUNction{1 | 2 | 3 | 4}:PEAK?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:PEAK] <peak1\_number>,<peak2\_number><NL>  
**Where:** <peak1\_number> ::= integer, 1 through 99  
 <peak2\_number> ::= integer, 1 through 99

---

## **:FUNction{1 | 2 | 3 | 4}:POINTs**

**Command** :FUNction{1 | 2 | 3 | 4}:POINTs <points\_argument>  
**Query** :FUNction{1 | 2 | 3 | 4}:POINTs?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:POINTs] <points\_argument><NL>  
**Where:** <points\_argument> ::= integer, 512, 1024, 2048, 4096, 8192, 16384, or 32768

---

## **:FUNction{1 | 2 | 3 | 4}:RANGe**

**Command** :FUNction{1 | 2 | 3 | 4}:RANGe <range\_argument>  
**Query** :FUNction{1 | 2 | 3 | 4}:RANGe?  
**Returned Format** [:FUNction{1 | 2 | 3 | 4}:RANGe] <range\_argument><NL>  
**Where:** <range\_argument> ::= exponential full scale vertical range in volts, dB, or dBm

---

## :FUNction{1 | 2 | 3 | 4}:SETup?

Query

Returned Format

```
:FUNction{1 | 2 | 3 | 4}:SETup?
:FUNction{1 | 2 | 3 | 4}:DISP {0 | 1};
  {ADD | SUBT | MULT | VERS | ONLY | INV | INT | DIFF |
   FFT} {CHAN<n> | WMEM {1 | 2 | 3 | 4} [, {CHAN<n> |
   WMEM {1 | 2 | 3 | 4}}];
OFF <offset_argument>;
RANG <range_argument>;
FREQ <frequency_argument>; (FFT only)
LEV <level_argument>; (FFT only)
MAGN {0 | 1}; (FFT only)
PEAK <peak1_number>,peak2_number>; (FFT only)
POIN <points_argument>; (FFT only)
SPAN <span_argument>; (FFT only)
WIND {RECT | HANN | FLAT} (FFT only)<NL>
```

Where:

```
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
<offset_argument> ::= exponential offset value of 0 to ± 200
                    dBm for FFT function
<range_argument> ::= exponential full scale vertical range in
                    volts, dB, or dBm
<frequency_argument> ::= exponential center frequency from
                       0 Hz to 1.5X of frequency span
<level_argument> ::= exponential level value 0 to ± 600 dBm
<peak1_number> ::= integer, 1 through 99
<peak2_number> ::= integer, 1 through 99
<points_argument> ::= integer, 512, 1024, 2048, 4096, 8192,
                    16384, or 32768
<span_argument> ::= exponential number in hertz
```

## :FUNction{1 | 2 | 3 | 4}:SPAN

Command

Query

Returned Format

Where:

```
:FUNction{1 | 2 | 3 | 4}:SPAN <span_argument>
:FUNction{1 | 2 | 3 | 4}:SPAN?
[:FUNction{1 | 2 | 3 | 4}:SPAN] <span_argument><NL>
<span_argument> ::= exponential number in hertz
```

---

## **:FUNction{1 | 2 | 3 | 4}:SUBTract**

**Command**

**Where:**

```
:FUNction{1 | 2|3|4}:SUBTract <operand1>,<operand2>
<operand1> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<operand2> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

## **:FUNction{1 | 2 | 3 | 4}:VERSus**

**Command**

**Where:**

```
:FUNction{1 | 2 | 3 | 4}:VERSus <Y_operand>,<X_operand>
<Y_operand1> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<X_operand2> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
      (HP 54540A/54542A)
```

---

## **:FUNction{1 | 2 | 3 | 4}:WINDow**

**Command**

**Query**

**Returned Format**

```
:FUNction{1 | 2 | 3 | 4}:WINDow {RECTangular | HANNing |
  FLATtop}
:FUNction:WINDow?
[:FUNction{1|2|3|4}:WINDow] {RECTangular | HANNing |
  FLATtop}<NL>
```

---

## **:HARDcopy:LENGth**

**Command**

**Query**

**Returned Format**

```
:HARDcopy:LENGth {11 | 12}
:HARDcopy:LENGth?
[:HARDcopy:LENGth] {11 | 12}<NL>
```

---

**:HARDcopy:MODE**

**Command** :HARDcopy:MODE {THINKJET | DJ5XBW75DPI | DJ5XBW100DPI | DJ5XBW150DPI | DJ5XBW300DPI | DJET75DPI | DJET100DPI | DJET150DPI | DJET300DPI | LASERJET\_II | LASERJET\_IIP | PJETXL300 | PJETXL | PAINTJET | COLORPRO | HP7475A | HP7470A | HP7550A | DJET310C | DJET320C | DJET500C | DJET540C | DJET560C | DJET600C | DJET660C | DJET850C | DJET855C | P1200C | QUIETJETALT | EPSON5000}

**Query** :HARDcopy:MODE?

**Returned Format** [:HARDcopy:MODE] {THINKJET | DJ5XBW75DPI | DJ5XBW100DPI | DJ5XBW150DPI | DJ5XBW300DPI | DJET75DPI | DJET100DPI | DJET150DPI | DJET300DPI | LASERJET\_II | LASERJET\_IIP | PJETXL300 | PJETXL | PAINTJET | COLORPRO | HP7475A | HP7470A | HP7550A | DJET310C | DJET320C | DJET500C | DJET540C | DJET560C | DJET600C | DJET660C | DJET850C | DJET855C | P1200C | QUIETJETALT | EPSON5000}<NL>

---

**:HARDcopy:PAGE**

**Command** :HARDcopy:PAGE {MANual | AUTomatic}

**Query** :HARDcopy:PAGE?

**Returned Format** [:HARDcopy:PAGE] {MANual | AUTomatic}<NL>

---

**:HARDcopy:PLOT:AREA**

**Command** :HARDcopy:PLOT:AREA {ALL | DISPlay | FACTors | GRATICule | LABeled}

**Query** :HARDcopy:PLOT:AREA?

**Returned Format** [:HARDcopy:PLOT:AREA] {ALL | DISPlay | FACTors | GRATICule | LABeled}<NL>

---

**:HARDcopy:PLOT:INITialize**

**Command** :HARDcopy:PLOT:INITialize {{OFF | 0} | {ON | 1}}

**Query** :HARDcopy:PLOT:INITialize?

**Returned Format** [:HARDcopy:PLOT:INITialize] {0 | 1}<NL>

---



---

## **:HARDcopy:PLOT:{PEN|COLor}**

**Command**  
**Query**  
**Returned Format**  
**Where:**

```
:HARDcopy:PLOT:{PEN|COLor } <item>,<pen_number>
:HARDcopy:PLOT:{PEN|COLor}? <item>
[:HARDcopy:PLOT:{PEN|COLor}] <pen_number><NL>
<item> ::= {CHANnel<n> | WMEMory{1 | 2 | 3 | 4} |
  FUNction{1 | 2 | 3 | 4} | PMEMory{1 | 2} | Y{1 | 2}Marker |
  X{1 | 2}Marker | GRATicule | TRIGger | TIMEbase | MEASure |
  TITLes}
<n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4
  (HP 54540A/54542A)
<pen_number> ::= integer, 0 through 8
```

---

## **:LER?**

**Query**  
**Returned Format**

```
:LER?
[:LER] {0 | 1}<NL>
```

---

## **:LTER?**

**Query**  
**Returned Format**

```
:LTER?
[:LTER] {0 | 1}<NL>
```

---

## **:MARKer:DISPlay**

**Command**  
**Query**  
**Returned Format**

```
:MARKer:DISPlay {{OFF | 0} | {ON | 1}}
:MARKer:DISPlay?
[:MARKer:DISPlay] {0 | 1}<NL>
```

---

## **:MARKer:MODE**

**Command**  
**Query**  
**Returned Format**

```
:MARKer:MODE {MANual | WAVeform}
:MARKer:MODE?
[:MARKer:MODE] {MANual | WAVeform}<NL>
```

---

## :MARKer:SETup?

**Query** :MARKer:SETup?  
**Returned Format** :MARK:MODE {MAN | WAV};DISP {0 | 1};  
 XDEL <xdelta>;  
 X1P <xposition\_argument>;  
 X2P <xposition\_argument>;  
 X1Y1 {CHAN<n> | FUNC{1 | 2 | 3 | 4} | WMEM{1 | 2 | 3 | 4}};  
 X2Y2 {CHAN<n> | FUNC{1 | 2 | 3 | 4} | WMEM{1 | 2 | 3 | 4}};  
 Y1P <yposition\_argument>; (MAN mode only)  
 Y2P <yposition\_argument>; (MAN mode only)  
 YDEL <ydelta>; (MAN mode only)  
 Y1P <yposition\_argument>; (MAN mode only)  
 Y2P <yposition\_argument><NL> (MAN mode only)

**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or  
 ::= integer, 1 through 4 (HP 54540A/54542A)  
 <xdelta> ::= exponential, difference in seconds between x1 and x2 markers  
 <xposition\_argument> ::= exponential, xmarker position in seconds or  
 hertz  
 <ydelta> ::= exponential, difference between y1 and y2 markers  
 <yposition\_argument> ::= exponential, ymarker position in volts or power

## :MARKer:X1Position

**Command** :MARKer:X1Position <xposition\_argument>  
**Query** :MARKer:X1Position?  
**Returned Format** [:MARKer:X1Position] <xposition\_argument><NL>  
**Where:** <xposition\_argument> ::= exponential, xmarker time in seconds,  
 or frequency in hertz

## :MARKer:X2Position

**Command** :MARKer:X2Position <xposition\_argument>  
**Query** :MARKer:X2Position?  
**Returned Format** [:MARKer:X2Position] <xposition\_argument><NL>  
**Where:** <xposition\_argument> ::= exponential, xmarker time in  
 seconds, or frequency in hertz

### **:MARKer:X1Y1source**

**Command** :MARKer:X1Y1source {CHANnel<n> | {FUNction{1 | 2 | 3 | 4} | WMEMory{1 | 2 | 3 | 4}}

**Query** :MARKer:X1Y1source?

**Returned Format** [:MARKer:X1Y1source] {CHANnel<n>| {FUNction{1 | 2 | 3 | 4} | WMEMory{1 | 2 | 3 | 4}}<NL>

**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4 (HP 54540A/54542A)

### **:MARKer:X2Y2source**

**Command** :MARKer:X2Y2source {CHANnel<n> | {FUNction{1 | 2 | 3 | 4} | WMEMory{1 | 2 | 3 | 4}}

**Query** :MARKer:X2Y2source?

**Returned Format** [:MARKer:X2Y2source] {CHANnel<n>| {FUNction{1 | 2 | 3 | 4} | WMEMory{1 | 2 | 3 | 4}}<NL>

**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4 (HP 54540A/54542A)

### **:MARKer:XDELta?**

**Query** :MARKer:XDELta?

**Returned Format** [:MARKer:XDELta] <xdelta><NL>

**Where:** <xdelta> ::= exponential, difference between x1 and x2 markers

### **:MARKer:Y1Position** (Command ignored in waveform mode)

**Command** :MARKer:Y1Position <yposition\_argument>

**Query** :MARKer:Y1Position?

**Returned Format** [:MARKer:Y1Position] <yposition\_argument><NL>

**Where:** <yposition\_argument> ::= exponential, ymarker level in volts, or power in dBm

---

**:MARKer:Y2Position** (Command ignored in waveform mode)

**Command** :MARKer:Y2Position <yposition\_argument>  
**Query** :MARKer:Y2Position?  
**Returned Format** [:MARKer:Y2Position] <yposition\_argument><NL>  
**Where:** <yposition\_argument> ::= exponential, ymarker level in volts,  
or power in dBm

---

**:MARKer:YDELta?**

**Query** :MARKer:YDELta?  
**Returned Format** [:MARKer:YDELta] <ydelta><NL>  
**Where:** <ydelta> ::= exponential, difference between y1 and y2 markers

---

**:MEASure:ALL?**

**Query** :MEASure:ALL?  
**Returned Format** [:MEASure] [:DELay] <result>; [:DUTYcycle] <result>;  
[:FALLtime] <result>; [:FREQuency] <result>;[:NWIDth]  
<result> ; [:OVERshoot] <result>; [:PERiod] <result>;  
[:PREShoot] <result>;[:PWIDth] <result>; [:RISetime]  
<result>; [:VACRms] <result>; [:VAMPLitude] <result>;  
[:VAverage] <result>; [:VBASe] <result>; [:VDCRms] <result>;  
[:VMAX] <result>; [:VMIN] <result>; [:VPP] <result>; [:VTOP]  
<result>;<NL>  
**Where:** <result> ::=exponential, individual measurement results

---

**:MEASure:COMPare**

**Command** :MEASure:COMPare <measurement>,<upper\_limit>,<lower\_limit>  
**Query** :MEASure:COMPare? <measurement>  
**Returned Format** [:MEASure:COMPare]  
<measurement>,<upper\_limit>,<lower\_limit><NL>  
**Where:** <measurement> ::= {DELay | DUTYcycle | FALLtime | FREQuency |  
NWIDth | OVERshoot | PERiod | PREShoot | PWIDth | RISetime |  
VACRms | VAMPLitude | VAverage | VBASe | VDCRms | VMAX |  
VMIN | VPP | VTOP}  
<upper\_limit> ::= exponential, high limit value  
<lower\_limit> ::=exponential, low limit value

---

---

## :MEASure:CURSor?

**Query** :MEASure:CURSor? {DELTA | START | STOP}  
**Returned Format** [:MEASure:CURSor] <time>,<voltage><NL>  
**Where:** <time> ::= exponential, delta time, y1marker time, or y2marker time  
 <voltage> ::= exponential, delta voltage, x1marker voltage, or x2marker voltage

---

## :MEASure:DEFine

**Command** :MEASure:DEFine <define\_argument>  
**Query** :MEASure:DEFine? {DELay | PWIDth | NWIDth}  
**Returned Format** [:MEASure:DEFine] <define\_argument><NL>  
**Where:** <define\_argument> ::= {DELay,<polarity>,<edge\_number>,<level>,<polarity>,<edge\_number>,<level> | PWIDth,<level> | NWIDth,<level>}  
 <polarity> ::= {POSitive | NEGative}  
 <edge\_number> ::= integer, 1 to 4000 specifying a displayed edge  
 <level> ::= {MIDDLE | UPPer | LOWer}

---

## :MEASure:DELay

**Command** :MEASure:DELay  
**Query** :MEASure:DELay?  
**Returned Format** [:MEASure:DELay] <value><NL>  
**Where:** <value> ::= exponential, time value in seconds

---

**:MEASure:DESTination**

**Command** :MEASure:DESTination <source\_argument>,<destination\_argument>  
**Query** :MEASure:DESTination? {CHANnel<n> | FUNction{1 | 2 | 3 | 4} | SCReen}?

**Returned Format** [:MEASure:DESTination] {{CHANnel<n> | {FUNction{1 | 2 | 3 | 4} | SCReen}} {WMEMory{1 | 2 | 3 | 4} | PMEMory{1 | 2} | MULTiple | OFF}<NL>

**Where:** <source\_argument> ::= {{SCReen,{OFF | PMEMory{1 | 2}}}} | {{CHANnel<n> | FUNction{1 | 2 | 3 | 4}}}  
 <distination\_argument> ::= {WMEMory{1 | 2 | 3 | 4} | MULTiple | OFF}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4 (HP 54540A/54542A)

---

**:MEASure:DUTycycle**

**Command** :MEASure:DUTycycle  
**Query** :MEASure:DUTycycle?  
**Returned Format** [:MEASure:DUTycycle] <value><NL>  
**Where:** <value> ::= exponential, ratio of positive pulse width to period

---

**:MEASure:ESTArt**

**Command** :MEASure:ESTArt <slope\_and\_occurrence>  
**Query** :MEASure:ESTArt?  
**Returned Format** [:MEASure:ESTArt] <slope\_and\_occurrence><NL>  
**Where:** <slope\_and\_occurrence> ::= integer, -4000 to 4000 (excluding 0)

---

**:MEASure:ESTOp**

**Command** :MEASure:ESTOp <slope\_and\_occurrence>  
**Query** :MEASure:ESTOp?  
**Returned Format** [:MEASure:ESTOp] <slope\_and\_occurrence><NL>  
**Where:** <slope\_and\_occurrence> ::= integer, -4000 to 4000 (excluding 0)

---

---

## :MEASure:EANalysis

Command :MEASure:EANalysis {{OFF | 0} | {ON | 1}}

Query :MEASure:EANalysis?

Returned Format [:MEASure:EANalysis] {0 | 1}<NL>

---

## :MEASure:FALLtime

Command :MEASure:FALLtime

Query :MEASure:FALLtime?

Returned Format [:MEASure:FALLtime] <value><NL>

Where: <value> ::= exponential, time value in seconds between lower threshold and upper threshold voltage points

---

## :MEASure:FREQuency

Command :MEASure:FREQuency

Query :MEASure:FREQuency?

Returned Format [:MEASure:FREQuency] <value><NL>

Where: <value> ::= exponential, frequency in hertz

---

## :MEASure:LIMittest

Command :MEASure:LIMittest {MEASure | OFF}

---

## :MEASure:LOWer

Command :MEASure:LOWer [<source>,<lower\_threshold>

Query :MEASure:LOWer? [<source>]

Returned Format [:MEASure:LOWer] <source>,<lower\_threshold><NL>

Where: <source> ::= {CHANnel<n> | FUNction{1 | 2 | 3 | 4} | WMEMory{1 | 2 | 3 | 4}}

<lower\_threshold> ::= integer, user defined lower threshold in percent or volts (selected by :MEASure:UNITs)

<n> ::= integer 1 or 2 (HP 54520A/54522A), or 1 through 4 (HP 54540A/54542A)

---

---

**:MEASure:MODE**

Command :MEASure:MODE {STANdard | USER}  
 Query :MEASure:MODE?  
 Returned Format [ :MEASure:MODE ] {STANdard | USER}<NL>

---

**:MEASure:MWINDow**

Command :MEASure:MWINDow {MARKers | SCReen}  
 Query :MEASure:MWINDow?  
 Returned Format [ :MEASure:MWINDows ] {MARKers | SCReen}<NL>

---

**:MEASure:NWIDth**

Command :MEASure:NWIDth  
 Query :MEASure:NWIDth?  
 Returned Format [ :MEASure:NWIDth ] <value><NL>  
 Where: <value> ::= exponential, negative pulse width in seconds

---

**:MEASure:OVERshoot**

Command :MEASure:OVERshoot  
 Query :MEASure:OVERshoot?  
 Returned Format [ :MEASure:OVERshoot ] <value><NL>  
 Where: <value> ::= exponential, ratio of overshoot to Vamplitude

---

**:MEASure:PERiod**

Command :MEASure:PERiod  
 Query :MEASure:PERiod?  
 Returned Format [ :MEASure:PERiod ] <value><NL>  
 Where: <value> ::= exponential, waveform period in seconds

---

**:MEASure:POSTfailure**

Command :MEASure:POSTfailure {CONTInue | STOP}  
 Query :MEASure:POSTfailure?  
 Returned Format [ :MEASure:POSTfailure ] {CONTInue | STOP}<NL>

---



---

## :MEASure:PREShoot

**Command** :MEASure:PREShoot  
**Query** :MEASure:PREShoot?  
**Returned Format** [:MEASure:PREShoot] <value><NL>  
**Where:** <value> ::= exponential, ratio of preshoot to Vamplitude

---

## :MEASure:PWIDth

**Command** :MEASure:PWIDth  
**Query** :MEASure:PWIDth?  
**Returned Format** [:MEASure:PWIDth] <value><NL>  
**Where:** <value> ::= exponential, width of positive pulse in sseconds

---

## :MEASure:RESults?

**Query** :MEASure:RESults?  
**Returned Format** [:MEASure:RESults] <number\_of\_meas>[;<measurement> ]...<NL>  
**Where:** <number\_of\_meas> ::= integer, number of measurements displayed  
           on the screen, 0 through 23  
           <measurement> ::= {DELay <result> | DUTycycle <result> | FALLtime <result> | FREQuency <result> | NWIDth <result> | OVERshoot <result> | PERiod <result> | PREShoot <result> | PWIDth <result> | RISetime <result> | TMAX <result> | TMIN <result> | TVOLt <result> | VACRms <result> VAMPLitude <result> | VAVerage <result> | VBASE <result> | VDCRms <result> | VMAX <result> | VMIN <result> | VPP <result> | VTIme <result> | VTOP <result>}  
           <result> ::=exponential, individual measurement results

---

## :MEASure:RISetime

**Command** :MEASure:RISetime  
**Query** :MEASure:RISetime?  
**Returned Format** [:MEASure:RISetime] <value><NL>  
**Where:** <value> ::= exponential, rise time in seconds

---

---

**:MEASure:SCRatch**

Command :MEASure:SCRatch

---

**:MEASure:SOURce**

Command :MEASure:SOURce <source>[,<source>]  
Query :MEASure:SOURce?  
Returned Format [:MEASure:SOURce] <source>[,<source>]<NL>  
Where: <source> ::= {CHANnel<n> | FUNction{1 | 2 | 3 | 4} |  
WMEMory{1 | 2 | 3 | 4}}  
<n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
(HP 54540A/54542A)

---

**:MEASure:STATistics**

Command :MEASure:STATistics {{OFF | 0} | {ON | 1}}  
Query :MEASure:STATistics?  
Returned Format [:MEASure:STATistics] {0| 1}<NL>

---

**MEASure:STATistics:MODE**

Command :MEASure:STATistics:MODE {NORMAL | SDEVIation}}  
Query :MEASure:STATistics:MODE?  
Returned Format [:MEASure:STATistics:MODE] {NORMAL | SDEVIation}<NL>

---

**:MEASure:TDELta**

Query :MEASure:TDELta?  
Returned Format [:MEASure:TDELta] <value><NL>  
Where: <value> ::= exponential, difference between x2 and x1 markers

---

**:MEASure:TMAX?**

Query :MEASure:TMAX?  
Returned Format [:MEASure:TMAX] <time><NL>  
Where: <time> ::= exponential, time at maximum voltage

---

---

### **:MEASure:TMIN?**

Query :MEASure:TMIN?  
 Returned Format [:MEASure:TMIN] <time><NL>  
 Where: <time> ::= exponential, time at minimum voltage

---

### **:MEASure:TSTArt**

Command :MEASure:TSTArt <tstart \_argument>  
 Query :MEASure:TSTArt?  
 Returned Format [:MEASure:TSTArt] <tstart \_argument><NL>  
 Where: <tstart\_argument> ::= exponential, time at x1marker in seconds

---

### **:MEASure:TSTOp**

Command :MEASure:TSTOp <tstop\_argument>  
 Query :MEASure:TSTOp?  
 Returned Format [:MEASure:TSTOp] <tstop\_argument><NL>  
 Where: <tstop\_argument> ::= exponential, time at x2marker in seconds

---

### **:MEASure:TVOLt?**

Query :MEASure:TVOLt? <tvolt\_argument>,<slope\_and\_occurrence>  
 Returned Format [:MEASure:TVOLt] <time><NL>  
 Where: <tvolt\_argument> ::= real number representing positive or  
         negative voltage level that the waveform must cross  
 <slope\_and\_occurrence> ::= slope is the direction of the  
         waveform when <voltage> is crossed - rising (space character  
         or +) or falling (-)  
         occurrence is the number of crossings to be reported (if one  
         - the first crossing is reported, if two - the second  
         crossing is reported  
 <time> ::= exponential, time in seconds of specified voltage  
         crossing

---

---

**:MEASure:UNITs**

Command :MEASure:UNITs {PERCent | VOLTs}  
 Query :MEASure:UNITs?  
 Returned Format [ :MEASure:UNITs ] {PERCent | VOLTs}<NL>

---

**:MEASure:UPPer**

Command :MEASure:UPPer [<source,>]<upper\_threshold>  
 Query :MEASure:UPPer? [<source>]  
 Returned Format [ :MEASure:UPPer ] <source>,<upper\_threshold><NL>  
 Where: <source> ::= {CHANnel<n> | FUNction{1 | 2 | 3 | 4} |  
 WMEMory{1 | 2 | 3 | 4}}  
 <upper\_threshold> ::= integer, user defined upper threshold in  
 percent or volts (selected by :MEASure:UNITs)  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
 (HP 54540A/54542A)

---

**:MEASure:VACRms**

Command :MEASure:VACRms  
 Query :MEASure:VACRms?  
 Returned Format [ :MEASure:VACRms ] <value><NL>  
 Where: <value> ::= exponential, calculated ac rms voltage

---

**:MEASure:VAMPlitude**

Command :MEASure:VAMplitude  
 Query :MEASure:VAMplitude?  
 Returned Format [ :MEASure:VAMplitude ] <value><NL>  
 Where: <value> ::= exponential, difference between top and base  
 voltages

---

**:MEASure:VAverage**

Command :MEASure:VAverage  
 Query :MEASure:VAverage?  
 Returned Format [ :MEASure:VAverage ] <value><NL>  
 Where: <value> ::= exponential, calculated average voltage

---

---

### **:MEASure:VBASE**

Command	:MEASure:VBASE
Query	:MEASure:VBASE?
Returned Format	[ :MEASure:VBASE ] <value><NL>
Where:	<value> ::= exponential, voltage at base of selected waveform

---

### **:MEASure:VDCRms**

Command	:MEASure:VDCRms
Query	:MEASure:VDCRms?
Returned Format	[ :MEASure:VDCRms ] <value><NL>
Where:	<value> ::= exponential, calculated dc rms voltage

---

### **:MEASure:VDELta?**

Query	:MEASure:VDELta?
Returned Format	[ :MEASure:VDELta ] <value><NL>
Where:	<value> ::= exponential, delta V value in volts

---

### **:MEASure:VFIFty**

Command	:MEASure:VFIFty
---------	-----------------

---

### **:MEASure:VMAX**

Command	:MEASure:VMAX
Query	:MEASure:VMAX?
Returned Format	[ :MEASure:VMAX ] <value><NL>
Where:	<value> ::= exponential, maximum voltage of selected waveform

---

### **:MEASure:VMIN**

Command	:MEASure:VMIN
Query	:MEASure:VMIN?
Returned Format	[ :MEASure:VMIN ] <value><NL>
Where:	<value> ::= exponential, minimum voltage value of the selected waveform

---

---

**:MEASure:VPP**

Command :MEASure:VPP  
 Query :MEASure:VPP?  
 Returned Format [:MEASure:VPP] <value><NL>  
 Where: <value> ::= exponential, voltage peak to peak

---

**:MEASure:VRELative**

Command :MEASure:VRELative <percent\_argument>  
 Query :MEASure:VRELative?  
 Returned Format [:MEASure:VRELative] <percent\_argument><NL>  
 Where: <percent\_argument> ::= integer, Vmarker2 relative position in percent, from 0 through 100

---

**:MEASure:VSTArt**

Command :MEASure:VSTArt <vstart\_argument>  
 Query :MEASure:VSTArt?  
 Returned Format [:MEASure:VSTArt] <vstart\_argument><NL>  
 Where: <vstart\_argument> ::= exponential, voltage at y1marker

---

**:MEASure:VSTOp**

Command :MEASure:VSTOp <vstop\_argument>  
 Query :MEASure:VSTOp?  
 Returned Format [:MEASure:VSTOp] <vstop\_argument><NL>  
 Where: <vstop\_argument> ::= exponential, voltage at y2marker

---

**:MEASure:VTIME?**

Query :MEASure:VTIME? <vtime\_argument>  
 Returned Format [:MEASure:VTIME] <voltage><NL>  
 Where: <vtime\_argument> ::= real number representing the displayed time from the trigger in seconds  
 <voltage> ::= exponential, voltage at specified time

---

---

## :MEASure:VTOP

**Command** :MEASure:VTOP  
**Query** :MEASure:VTOP?  
**Returned Format** [:MEASure:VTOP] <value><NL>  
**Where:** <value> ::= exponential, voltage at the top of the waveform

---

## :MEASure:WCOMpare:ALLowance

**Command** :MEASure:WCOMpare:ALLowance <allow\_argument>  
**Query** :MEASure:WCOMpare:ALLowance?  
**Returned Format** [:MEASure:WCOMpare:ALLowance] <allow\_argument><NL>  
**Where:** <allow\_argument> ::= real number representing number of vertical divisions of allowance, from 0.0 to 8.0

---

## :MEASure:WCOMpare:COMParE

**Command** :MEASure:WCOMpare:COMParE {CHANnel<n> |  
 FUNCTION{1 | 2 | 3 | 4}}, WMEMory{1 | 2 | 3 | 4}}  
**Query** :MEASure:WCOMpare:COMParE?  
**Returned Format** [:MEASure:WCOMpare:COMParE] {CHANnel<n> |  
 FUNCTION{1 | 2 | 3 | 4}}, WMEMory{1 | 2 | 3 | 4}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
 (HP 54540A/54542A)

---

## :MEASure:WCOMpare:DESTination

**Command** :MEASure:WCOMpare:DESTination  
 <source\_argument>, <destination\_argument>  
**Query** :MEASure:WCOMpare:DESTination? {CHANnel<n> |  
 FUNCTION{1 | 2 | 3 | 4} | SCReen}  
**Returned Format** [:MEASure:WCOMpare:DESTination] {CHANnel<n>} |  
 {FUNCTION{1 | 2 | 3 | 4} | SCReen}} {WMEMory{1 | 2 | 3 | 4}  
 | PMEMory{1 | 2} | MULTiple | OFF}<NL>  
**Where:** <source\_argument> ::= {{SCReen, {OFF | PMEMory{1 | 2}}}} |  
 {{CHANnel<n> | FUNCTION{1 | 2 | 3 | 4}}}  
 <distination\_argument> ::= {WMEMory{1 | 2 | 3 | 4} | MULTiple  
 | PMEMory{1 | 2} | OFF}  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A) or 1 through 4  
 (HP 54540A/54542A)

---

---

**:MEASure:WCOMpare:HALLowance**

**Command** :MEASure:WCOMpare:HALLowance <allow\_argument>  
**Query** :MEASure:WCOMpare:HALLowance?  
**Returned Format** [:MEASure:WCOMpare:HALLowance] <allow\_argument><NL>  
**Where:** <allow\_argument> ::= real number representing number of horizontal divisions of allowance, from 0.0 to 8.0

---

**:MEASure:WCOMpare:POSTfailure**

**Command** :MEASure:WCOMpare:POSTfailure {CONTInue | STOP}  
**Query** :MEASure:WCOMpare:POSTfailure?  
**Returned Format** [:MEASure:WCOMpare:POSTfailure] {CONTInue | STOP}<NL>

---

**:MEASure:WCOMpare:VALLowance**

**Command** :MEASure:WCOMpare:VALLowance <allow\_argument>  
**Query** :MEASure:WCOMpare:VALLowance?  
**Returned Format** [:MEASure:WCOMpare:VALLowance] <allow\_argument><NL>  
**Where:** <allow\_argument> ::= real number representing number of vertical divisions of allowance, from 0.0 to 8.0

---

**:MEASure:WCOMpare:WTEST**

**Command** :MEASure:WCOMpare:WTEST {MEASure | OFF}  
**Query** :MEASure:WCOMpare:WTEST?  
**Returned Format** [:MEASure:WCOMpare:WTEST] {MEASure | OFF}<NL>

---

**:MENU**

**Command** :MENU {CHANnel<n>TIMEbase | TRIGger | DISK | DISPlay | DELTa | MATH | SAVE | MEASure | UTILity | SHOW}  
**Query** :MENU?  
**Returned Format** [:MENU] {CHANnel<n>TIMEbase | TRIGger | DISK | DISPlay | DELTa | MATH | SAVE | MEASure | UTILity | SHOW}  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4 (HP 54540A/54542A)

---



---

## **:MERGe**

**Command** :MERGe <pmemory\_num>  
**Where:** <pmemory\_num> ::= integer, 1 or 2

---

## **:MMEMory:DISPlay**

**Command** :MMEMory:DISPlay {{OFF | 0} | {ON | 1}}  
**Query** :MMEMory:DISPlay?  
**Returned Format** [:MMEMory:DISPlay] {0 | 1}<NL>

---

## **:MMEMory:FNUMber**

**Command** :MMEMory:FNUMber <failure\_number>  
**Query** :MMEMory:FNUMber?  
**Returned Format** [:MMEMory:FNUMber] <failure\_number><NL>  
**Where:** <failure\_number> ::= integer, 0 to 665

---

## **:MMEMory:SOURce**

**Command** :MMEMory:SOURce {CHANnel<n> | FUNction{1 | 2 | 3 | 4}}  
**Query** :MMEMory:SOURce?  
**Returned Format** [:MMEMory:SOURce] {CHANnel<n> | FUNction{1 | 2 | 3 | 4}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
 (HP 54540A/54542A)

---

## **:MMEMory:STORe**

**Command** :MMEMory:STORe {WMEMory{1 | 2 | 3 | 4}}

---

## **:PCFRrequency**

**Command** :PCFRrequency <cal\_freq>  
**Query** :PCFRrequency  
**Returned Format** [:PCFRrequency] <cal\_freq>  
**Where:** <cal\_freq> ::= exponential, probe compensation signal  
 frequency in hertz

---

---

**:PLOT**

Query :PLOT?

---

**:PMEMory{1 | 2}:CLEAr**

Command :PMEMory{1 | 2}:CLEAr

---

**:PMEMory{1 | 2}:DISPlay**

Command :PMEMory{1 | 2}:DISPlay {{OFF | 0} | {ON | 1}}

Query :PMEMory{1 | 2}:DISPlay?

Returned Format [:PMEMory{1 | 2}:DISPlay] {0 | 1}<NL>

---

**:PMEMory{1 | 2}:MERGe**

Command :PMEMory{1 | 2}:MERGe

---

**:PMEMory{1 | 2}:SETup?**

Query :PMEMory{1 | 2}SETup?

Returned Format PMEMory{1 | 2}:DISP{0 | 1}<NL>

---

**:PLOT?**

Query :PLOT?

---

**:POWERup{RUNning | STOPped}**

Command :POWERup{RUNning | STOPped}

Query :POWERup?

Returned Format [:POWERup] {STOPped | RUNning}

---

---

### **:PRINT?**

Query :PRINT?

---

### **:RUN**

Command :RUN

---

### **:SEquential:DISPlay**

Command :SEquential:DISPlay {{OFF | 0} | {ON | 1}}  
 Query :SEquential:DISPlay?  
 Returned Format [:SEquential:DISPlay] {0 | 1}<NL>

---

### **:SEquential:EXCLude**

Command :SEquential:EXCLude <from\_argument>,<to\_argument>  
 Query :SEquential:EXCLude?  
 Returned Format [:SEquential:EXCLude] <exclude\_list><NL>  
 Where: <from\_argument> ::= integer, segment number of the lower end  
 of the exclude range  
 <to\_argument> ::= integer, segment number of the upper end  
 of the exclude range  
 number greater than or equal to the value of <from\_argument>  
 <exclude\_list> ::= integer, a list of previously captured  
 segment numbers separated by commas

---

### **:SEquential:INCLude**

Command :SEquential:INCLude <from\_argument>,<to\_argument>  
 Query :SEquential:INCLude?  
 Returned Format [:SEquential:INCLude] <include\_list><NL>  
 Where: <from\_argument> ::= integer, segment number of the lower end  
 of the include range  
 <to\_argument> ::= integer, segment number of the upper end  
 of the include range  
 <include\_list> ::=integer, a list of previously captured  
 segment numbers separated by commas

---

---

**:SEquential:NPOints**

**Command** :SEquential:NPOints <points\_argument>  
**Query** :SEquential:NPOints?  
**Returned Format:** [:SEquential:NPOints] <points\_argument><NL>  
**Where:** <points\_argument> ::= integer, 4 to 32768

---

**:SEquential:NSEGments**

**Command** :SEquential:NSEGments <segments\_argument>  
**Query** :SEquential:NSEGments?  
**Returned Format** [:SEquential:NSEGments] <segments\_argument><NL>  
**Where:** <segments\_argument> ::= integer, 1 to 8888 dependent on the SEquential:NPOints selection

---

**:SEquential:SETup?**

**Query** :SEquential:SETup?  
**Returned Format** :SEQ:DISP {0 | 1};  
 EXCL <exclude\_list>;  
 INCL <include\_list>;  
 NPO <points\_argument>;  
 NSEG <segment\_argument>;  
 SNUM <segments\_argument>; (Sequential Single Shot Mode Only)  
 SOURce CHANnel<n><NL> (Sequential Single Shot Mode Only)

**Where:** <exclude\_list> ::= integer, a list of previously captured segment numbers separated by commas  
 <include\_list> ::= integer, a list of previously captured segment numbers separated by commas  
 <points\_argument> ::= integer, 4 to 32768  
 <segments\_argument> ::= integer, 1 to 8888 dependent on the points selected  
 <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4 (HP 54540A/54542A)

---

---

## **:SEquential:SNUMber**

**Command**  
**Query**  
**Returned Format**  
**Where:**

```
:SEquential:SNUMber <segments_argument>
:SEquential:SNUMber?
[:SEquential:SNUMber] <segments_argument><NL>
<segments_argument> ::= integer, 1 to 8888 dependent on the
  number of segments acquired and ACQUIRE:TYPE currently
  selected
```

---

## **:SEquential:SOURce**

**Command**  
**Query**  
**Returned Format**  
**Where:**

```
:SEquential:SOURce {CHANnel<n>}
:SEquential:SOURce?
[:SEquential:SOURce] {CHANnel<n>}<NL>
<n> ::= integer 1 or 2 (HP 54520A/54522A), or 1 through 4
  (HP 54540A/54542A)
```

---

## **:SEquential:TTAGs?**

**Query**  
**Where:**

```
:SEquential:TTAGs? <segments_argument>
<segments_argument> ::= integer, 1 to 8888 dependent on the
  SEquential:NSEGments selection
```

---

## **:SEquential:TTDifference?**

**Query**  
**Where:**

```
:SEquential:TTDifference?
<segments_argument>, <segments_argument>
<segments_argument> ::= integer, 1 to 8888 dependent on the
  SEquential:NSEGments selection
```

---

## **:SERial**

**Command**  
**Where:**

```
:SERial <serial_number>
<serial_number> ::= 10 character serial number within quotes
```

---

## :STATus?

Query  
Returned Format  
Where:

```
:STATus? <display>
[:STATus] {0 | 1}<NL>
<display> ::= {CHANnel<n> | FUNction{1 | 2 | 3 | 4} |
  WMEMory{1 | 2 | 3 | 4} | PMEMory{1 | 2}<NL>
<n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4
  (HP 54540A/54542A)
```

---

## :STOP

Command

```
:STOP
```

---

## :STORE

Command  
Where:

```
:STORE <source>,<destination>
<source> ::= {CHANnel<n> | FUNction{1 | 2 | 3 | 4} |
  WMEMory{1 | 2 | 3 | 4}}
<destination> ::= {WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4
  (HP 54540A/54542A)
```

---

## :SYSTEM:DATE

Command  
Query  
Returned Format  
Where:

```
:SYSTEM:DATE <year>,<month>,<day>
:SYSTEM:DATE?
[:SYSTEM:DATE] "DDMMYYYY"<NL>
<year>/<YYYY> ::= integer, 1990 through 2059
<month> ::= integer, 1 through 12
<MMM> ::= three digit alphabetic month
<day>/<DD> ::= integer, 1 through 31
```

---

## :SYSTEM:DSP

Command  
Query  
Returned Format  
Where:

```
:SYSTEM:DSP <ascii_string>
:SYSTEM:DSP?
[:SYSTEM:DSP] <ascii_string><NL>
<ascii_string> ::= string response data containing the last
  information written on the advisory line
```

---

## **:SYSTEM:ERRor?**

Query  
Returned Format  
Where:

```
:SYSTEM:ERRor [{NUMBER | STRING}]
[:SYSTEM:ERRor] <error >[,<ascii_string>]<NL>
<error> ::= an integer error code
<ascii_string> ::= an alpha string specifying the error
condition
```

---

## **:SYSTEM:HEADer**

Command  
Query  
Returned Format

```
:SYSTEM:HEADer {{OFF | 0} | {ON | 1}}
:SYSTEM:HEADer?
[:SYSTEM:HEADer] {0 | 1}<NL>
```

---

## **:SYSTEM:KEY**

Command  
Query  
Returned Format  
Where:

```
:SYSTEM:KEY <key_code>
:SYSTEM:KEY?
[:SYSTEM:KEY] <key_code><NL>
<key_code> ::= integer, 0 through 63
```

---

## **:SYSTEM:LONGform**

Command  
Query  
Returned Format

```
:SYSTEM:LONGform {{OFF | 0} | {ON | 1}}
:SYSTEM:LONGform?
[:SYSTEM:LONGform] {0 | 1}<NL>
```

---

## **:SYSTEM:PIMacro**

Command  
Where:

```
:SYSTEM:PIMacro <ascii_string>
<ascii_string> ::= name of the defined macro
```

---

**:SYSTEM:SETup**

**Command** :SYSTEM:SETup <block\_data>  
**Query** :SYSTEM:SETup?  
**Returned Format** [:SYSTEM:SETup] <block\_data><NL>  
**Where:** <block\_data> ::= #800002048<setup\_string>  
 <setup\_string> ::= block of binary data bytes

---

**:SYSTEM:TIME**

**Command** :SYSTEM:TIME <hour>,<minute>,<second>  
**Query** :SYSTEM:TIME?  
**Returned Format** [:SYSTEM:TIME] "HH:MM:SS"<NL>  
**Where:** <hour>/<HH> ::= integer, 0 through 23  
 <month>/<MM> ::= integer, 0 through 59  
 <second>/<SS> ::= integer, 0 through 59

---

**:SYSTEM:UTILity**

**Command** :SYSTEM:UTILity:GMARkers {ON | OFF}  
 :SYSTEM:UTILity:LABels {ON | OFF}  
 :SYSTEM:UTILity:FACTor {ON | OFF}  
**Query** :SYSTEM:UTILity:GMARkers?  
 :SYSTEM:UTILity:LABels?  
 :SYSTEM:UTILity:FACTor?  
**Returned Format** [:SYSTEM:UTILity:GMARkers] {ON | OFF}<NL>

---

**:TER?**

**Query** :TER?  
**Returned Format** [:TER] {0 | 1}<NL>

---

**:TIMEbase:DELay**

**Command** :TIMEbase:DELay <delay\_value>  
**Query** :TIMEbase:DELay?  
**Returned Format** [:TIMEbase:DELay] <delay\_value><NL>  
**Where:** <delay\_value> ::= exponential, time from trigger to display  
 refrence in seconds

---



---

## :TIMEbase:MODE

Command	:TIMEbase:MODE {AUTO   TRIGgered   SINGLE}
Query	:TIMEbase:MODE?
Returned Format	[ :TIMEbase:MODE ] {AUTO   TRIGgered   SINGLE}<NL>

---

## :TIMEbase:RANGe

Command	:TIMEbase:RANGe <range_value>
Query	:TIMEbase:RANGe?
Returned Format	[ :TIMEbase:RANGe ] <range_value><NL>
Where:	<range_value> ::= exponential, 10 ns to 50 s in a 1,2,5 sequence

---

## :TIMEbase:REFerence

Command	:TIMEbase:REFerence {LEFT   CENTER   RIGHT}
Query	:TIMEbase:REFerence?
Returned Format	[ :TIMEbase:REFerence ] {LEFT   CENTER   RIGHT}<NL>

---

## :TIMEbase:RLENGth

Command	:TIMEbase:RLENGth <length_argument>
Query	:TIMEbase:RLENGth?
Returned Format	[ :TIMEbase:RLENGth ] <length_argument><NL>
Where:	<length_argument> ::= integer, 500 in the repetitive mode, 512, 1024, 2048, 4196, 8192, 16384, or 32768 in the real-time mode (sequential off), or 4 to 32768 in the real-time mode (sequential on), depending on the current :SEQential:NSEGments value

---

## :TIMEbase:SAMPle

Command	:TIMEbase:SAMPle {REALtime   REPetitive}
Query	:TIMEbase:SAMPle?
Returned Format	[ :TIMEbase:SAMPle ] {REALtime   REPetitive}<NL>

---

---

## **:TIMebase:SAMPle:CLOCK**

<b>Command</b>	<code>:TIMebase:SAMPle:CLOCK {AUTO   &lt;clock_value&gt;}</code>
<b>Query</b>	<code>:TIMebase:SAMPle:CLOCK?</code>
<b>Returned Format</b>	<code>[:TIMebase:SAMPle:CLOCK] &lt;clock_value&gt;&lt;NL&gt;</code>
<b>Where:</b>	<code>&lt;clock_value&gt; ::= exponential, 10S/s to 2GS/s</code>

---

## **:TIMebase:SETup?**

<b>Query</b>	<code>:TIMebase:SETup?</code>
<b>Returned Format</b>	<code>:TIM:DEL &lt;delay_value&gt;; MODE {AUTO TRIG SING}; RANGe &lt;range_value&gt;; RLEN &lt;length_argument&gt;; REF {LEFT CENT RIGH}; SAMPle {REAL REP}; SAMP:CLOC {AUTO   &lt;clock_value&gt;}&lt;NL&gt;</code>
<b>Where:</b>	<code>&lt;delay_value&gt; ::= exponential, time from trigger to display reference in seconds &lt;range_value&gt; ::= exponential, 5ns to 50s &lt;length_argument&gt; ::= 500 in the repetitive mode, 512, 1024, 2048, 4196, 8192, 16384, or 32768 in the real-time mode (sequential off), or 4 to 32768 in the real-time mode (sequential on), depending on the current :SEQuential:NSEGments value &lt;clock_value&gt; ::= exponential, 10S/s to 2GS/s</code>

---

## **:TRIGger:CENTerEd**

<b>Command</b>	<code>:TRIGger:CENTerEd</code>
----------------	--------------------------------

---

## :TRIGger:CONDition

**Command** :TRIGger:CONDition {ENTER | EXIT | TRUE | FALSE |  
 GT,<gt\_argument> | LT,<lt\_argument> |  
 RANGE,<range\_gt>,<range\_lt>}

**Query** :TRIGger:CONDition?

**Returned Format** [:TRIGger:CONDition] <argument><NL>

**Where:** <argument> ::= {ENTER | EXIT | GT,<gt\_argument> |  
 LT,<lt\_argument> | RANGE,<range\_gt>,<range\_lt> } | in PATTERN  
 or DELay with QUALify:PATtern selected; {TRUE | FALSE} (in  
 STATE or DELay with QUALify:STATe selected; or RANGE,  
 <range\_gt>,<range\_lt> in TV mode  
 <gt\_argument> ::= exponential, 20 ns to 160 ms  
 <lt\_argument> ::= exponential, 20 ns to 160 ms  
 <range\_gt> ::= exponential, 20 ns to 159.999 ms (must be less  
 than <range\_lt>)  
 <range\_lt> ::= exponential, 30 ns to 160 ms (must be greater  
 than <range\_gt>)

---

## :TRIGger:COUPling

**Command** :TRIGger:COUPling {AC | DC | LFReject}

**Query** :TRIGger:COUPling?

**Returned Format** [:TRIGger:COUPling] {AC | DC | LFReject}<NL>

---

## :TRIGger:DELay

**Command** :TRIGger:DELay {TIME,<time\_argument> | EVENT,<event\_argument>}

**Query** :TRIGger:DELay?

**Returned Format** [:TRIGger:DELay] {TIME,<time\_argument> |  
 EVENT,<event\_argument>}<NL>

**Where:** <time\_argument> ::= exponential, amount of delay from 30 ns  
 to  
 160 ms  
 <event\_argument> ::= integer, number of events from 1 to  
 16000000

---

---

**:TRIGger:DELay:SLOPe**

Command           :TRIGger:DELay:SLOPe {POSitive | NEGative}  
 Query             :TRIGger:DELay:SLOPe?  
 Returned Format   [:TRIGger:DELay:SLOPe] {POSitive | NEGative}<NL>

---

**:TRIGger:DELay:SOURce**

Command           :TRIGger:DELay:SOURce {CHANnel<n> | EXTernal}  
 Query             :TRIGger:DELay:SOURce?  
 Returned Format   [:TRIGger:DELay:SOURce] {CHANnel<n> | EXTernal}  
 Where:           <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
                   (HP 54540A/54542A)

NOTE: EXTernal available only on HP 54520A/54522A

---

**:TRIGger:FIELD**

Command           :TRIGger:FIELD {1 | 2}  
 Query             :TRIGger:FIELD?  
 Returned Format   [:TRIGger:FIELD] {1 | 2}<NL>

---

**:TRIGger:GLITch:CENTERed**

Command           :TRIGger:GLITch:CENTERed

---

**:TRIGger:GLITch:HOLDoff**

Command           :TRIGger:GLITch:HOLDoff <holdoff\_time>  
 Query             :TRIGger:GLITch:HOLDoff?  
 Returned Format   [:TRIGger:GLITch:HOLDoff] <holdoff\_time><NL>  
 Where:           <holdoff\_time> ::= exponential, 40 ns to 320 ms rounded to  
                   nearest 20 ns increment

---

---

## :TRIGger:GLITch:LEVel

**Command** :TRIGger:GLITch:LEVel <level\_argument>  
**Query** :TRIGger:GLITch:LEVel?  
**Returned Format** [:TRIGger:GLITch:LEVel] <level\_argument><NL>  
**Where:** <level\_argument> ::= exponential, for internal triggers,  $\pm 1.5$   
 x  
 full-scale voltage from center screen, for external  
 triggers  
 (HP 54520A/54522A)  $\pm 2$  volts with probe attenuation at 1:1

---

## :TRIGger:GLITch:SOURce

**Command** :TRIGger:GLITch:SOURce {CHANnel<n> | EXTernal},{HIGH | LOW}  
**Query** :TRIGger:GLITch:SOURce?  
**Returned Format** [:TRIGger:GLITch:SOURce] {CHANnel<n> | EXTernal},{HIGH | LOW}  
**Where:** <n> ::= integer 1 or 2 (HP 54520A/54522A), or 1 through 4  
 (HP 54540A/54542A)  
 NOTE: EXTernal available only on HP 54520A/54522A

---

## :TRIGger:GLITch:WIDth

**Command** :TRIGger:GLITch:WIDth {GT | LT},<width\_argument>  
**Query** :TRIGger:GLITch:WIDth?  
**Returned Format** [:TRIGger:GLITch:WIDth] {GT | LT},<width\_argument><NL>  
**Where:** <width\_argument> ::= exponential, 5 ns to 160 ms

---

## :TRIGger:HOLDoff

**Command** :TRIGger:HOLDoff {{TIME,<holdoff\_time>} |  
 { EVENT,<event\_argument>}}  
**Query** :TRIGger:HOLDoff?  
**Returned Format** [:TRIGger:HOLDoff] {TIME,<holdoff\_value> |  
 EVENT,<event\_argument>}<NL>  
**Where:** <holdoff\_time> ::= exponential, 40 ns to 320 ms  
 <event\_argument> ::= integer, 1 to 16000000

---

---

**:TRIGger:LEVel**

**Command** :TRIGger:LEVel <level\_argument>  
**Query** :TRIGger:LEVel?  
**Returned Format** [:TRIGger:LEVel] <level\_argument><NL>  
**Where:** <level\_argument> ::= for internal triggers,  $\pm 1.5$  x full-scale voltage from center screen; for external triggers,  $\pm 2$  volts with probe attenuation at 1:1

---

**:TRIGger:LINE**

**Command** :TRIGger:LINE <line\_number>  
**Query** :TRIGger:LINE?  
**Returned Format** [:TRIGger:LINE] <line\_number><NL>  
**Where:** <line\_number> ::= integer, 1 to 625 (depends on STANDARD and FIELD selection)

---

**:TRIGger:LOGic**

**Command** :TRIGger:LOGic {HIGH | LOW | DONTcare}  
**Query** :TRIGger:LOGic?  
**Returned Format** [:TRIGger:LOGic] {HIGH | LOW | DONTcare}<NL>

---

**:TRIGger:MODE**

**Command** :TRIGger:MODE {EDGE | PATtern | STATE | DELay | TV | GLITCh}  
**Query** :TRIGger:MODE?  
**Returned Format** [:TRIGger:MODE] {EDGE | PATtern | STATE | DELay | TV | GILTch}<NL>

---

**:TRIGger:NREJect**

**Command** :TRIGger:NREJect {{OFF | 0} | {ON | 1}}  
**Query** :TRIGger:NREJect?  
**Returned Format** [:TRIGger:NREJect] {0 | 1}<NL>

---

## :TRIGger:OCCurrence

**Command** :TRIGger:OCCurrence <occurrence\_argument>  
**Query** :TRIGger:OCCurrence?  
**Returned Format** [:TRIGger:OCCurrence] <occurrence\_argument><NL>  
**Where:** <occurrence\_argument> ::= integer, 1 to 16000000

## :TRIGger:OCCurrence:SLOPe

**Command** :TRIGger:OCCurrence:SLOPe {POSitive | NEGative}  
**Query** :TRIGger:OCCurrence:SLOPe?  
**Returned Format** [:TRIGger:OCCurrence:SLOPe] {POSitive | NEGative}<NL>

## :TRIGger:OCCurrence:SOURce

**Command** :TRIGger:OCCurrence:SOURce {CHANnel<n> | EXTernal}  
**Query** :TRIGger:OCCurrence:SOURce?  
**Returned Format** [:TRIGger:OCCurrence:SOURce] {CHANnel<n> | EXTernal}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
           (HP 54540A/54542A)

NOTE: EXTernal available only on HP 54520A/54522A

## :TRIGger:PATH

**Command** :TRIGger:PATH {CHANnel<n> | EXTernal}  
**Query** :TRIGger:PATH?  
**Returned Format** [:TRIGger:PATH] {CHANnel<n> | EXTernal}<NL>  
**Where:** <n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
           (HP 54540A/54542A)

NOTE: EXTernal available only on HP 54520A/54522A

## :TRIGger:POLarity

**Command** :TRIGger:POLarity {POSitive | NEGative}  
**Query** :TRIGger:POLarity?  
**Returned Format** [:TRIGger:POLarity] {POSitive | NEGative}<NL>

---

## :TRIGger:QUALify

Command	:TRIGger:QUALify {{EDGE   PATtern   STATE}   {LOW   HIGH}}
Query	:TRIGger:QUALify?
Returned Format	[ :TRIGger:QUALify ] {{EDGE   PATtern   STATE}   {LOW   HIGH}}<NL>

---

## :TRIGger:SETup?

Query	:TRIGger:SETup?
Returned Format	:TRIG:MODE EDGE; COUP {AC   DC   LFR}; HOLD {{TIME,<holdoff_time>}   EVENT, <event_argument>}}; LEV <level_argument>; NREJ {0   1}; SLOP {POS   NEG}; SOUR {CHAN<n>   EXT   LINE   AUX}<NL> (EXT is only valid for the 54520/22)
	:TRIG;MODE PATT; COND {ENT   EXIT   GT,<gt_argument>   LT,<lt_argument>   RANG,<range_gt>,<range_lt>}; HOLD {{TIME,<holdff_time>}   EVEN, <event_argument>}}; LEV <level_argument>; LOG {HIGH   LOW   DONT}; NREJ {0   1}; PATH {CHAN<n>   EXT}<NL> (EXT is only valid for the 54520/22)
	:TRIG:MODE STAT; COND {TRUE   FALS}; HOLD {{TIME,<holdoff_time>}   {EVEN, <event_argument>}}; LEV <level_argument>; LOG {HIGH   LOW   DONT}; NREJ {0   1}; PATH {CHAN<n>   EXT}; (EXT is only valid for the 54520/22) SLOP {POS   NEG}; SOUR {CHAN<n>   EXT}<NL> (EXT is only valid for the 54520/22)

---



## Programmer's Quick Reference Guide

```
:TRIG:MODE DEL;
COND {ENT | EXIT | TRUE | FALS | GT,<gt_argument> |
  LT,<lt_argument> | RANG,<range_gt>,<range_lt>};
DEL {{TIME,<time_value>} | {EVEN, <event_value>}};
DEL:SLOP {POS | NEG};
DEL:SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
LEV <level_argument>;
LOG {HIGH | LOW | DONT};
NREJ {0 | 1};
OCC <occurrence_argument>;
OCC:SLOP {POS | NEG};
OCC:SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
PATH {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
QUAL {EDGE | PATT | STAT};
SLOP {POS | NEG};
SOUR {CHAN<n> | EXT}<NL> (EXT is only valid for the
                          54520/22)

:TRIG:MODE TV;
COND {RANG,<range_gt>,<range_lt>};
FIEL {1 | 2};
HOLD {{TIME,<holdoff_time>} | {EVEN, <event_argument>}};
LEV <level_argument>;
LINE <line_number>;
NREJ {0 | 1};
OCC <occurrence_argument>;
OCC:SLOP {POS | NEG};
POL {POS | NEG};
QUAL {LOW | HIGH};
SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
STAN {525 | 625 | USER}<NL>

:TRIG:MODE GLIT;
GLITch:HOLD TIME,<holdoff_time>;
GLITch:LEV <level_argument>;
GLITch:NREJ {0 | 1};
GLITch:SOUR {CHAN<n> | EXT}, {LOW | HIGH}; (EXT is only valid
                                           for the 54520/22)
GLITch:WIDTH {GT | LT}, <width_argument><NL>
```

## Where:

<n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4  
(HP 54540A/54542A)  
<gt\_argument> ::= exponential, 20 ns to 160 ms  
<lt\_argument> ::= exponential, 20 ns to 160 ms  
<range\_gt> ::= exponential, 20 ns to 159.999 ms (must be less  
than <range\_lt>)  
<range\_lt> ::= exponential, 30 ns to 160 ms (must be greater  
than <range\_gt>)  
<time\_value> ::= exponential, amount of delay from 30ns to  
160ms  
<event\_value> ::= integer, number of events from 1 to 16000000  
<holdoff\_time> ::= exponential, 40 ns to 320 ms rounded to  
nearest 20 ns increment  
<level\_argument> ::= exponential, trigger level in volts  
<width\_argument> ::= exponential, 5 ns to 160 ms  
<event\_argument> ::= integer, 1 to 16000000  
<line\_number> ::= integer, 1 to 625 (depends on STANDard and  
FIELD selection)

---

## :VIEW

Command  
Where:

```
:VIEW <display>
<display> ::= CHANNEL<n>|FUNCTION{1 | 2 | 3 | 4} |
           PMEMory{1 | 2} | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520A/54522A), or 1 through 4
      (HP 54540A/54542A)
<n> ::= integer 1 or 2 (HP 54520A/54522A)
```

---

## :WAVEform:DATA

Command  
Query  
Returned Format  
Where:

```
:WAVEform:DATA <block_data>
:WAVEform:DATA?
[:WAVEform:DATA] <block_data><NL>
<block_data> ::= definite block data in IEEE 488.2 # format
```

---

## :WAVEform:FORMat

Command  
Query  
Returned Format

```
:WAVEform:FORMat {ASCIi | WORD | BYTE | COMPressed}
:WAVEform:FORMat?
[:WAVEform:FORMat] {ASCIi | WORD | BYTE | COMPressed}<NL>
```

---

## :WAVEform:POINts

Query  
Returned Format

```
:WAVEform:POINts?
[:WAVEform:POINts] {512 | 1024 | 2048 | 4096 | 8192 | 16384 |
32768}<NL>
```

---

## :WAVeform:PREAmble

<p><b>Command</b>  <b>Query</b>  <b>Returned Format</b>  <b>Where:</b></p>	<pre>:WAVeform:PREAmble &lt;preamble_data&gt; :WAVeform:PREAmble? [:WAVeform:PREAmble] &lt;preamble block&gt;&lt;NL&gt; &lt;preamble_data&gt; ::= &lt;format NR1&gt;,&lt;type NR1&gt;, &lt;points NR1&gt;,     &lt;count NR1&gt;,&lt;xincrement NR3&gt;,&lt;xorigin NR3&gt;,&lt;xreference NR1&gt;,     &lt;yincrement NR3&gt;,&lt;yorigin NR3&gt;,&lt;yreference NR1&gt; &lt;format&gt; ::= 0 for ASCii format     1 for BYTE format     2 for WORD format     4 for COMPRESSED format &lt;type&gt; ::= 0 for INVALID type     1 for NORMAL type or REALTIME     2 for AVERAGE type     3 for ENVELOPE type     4 for RAWDATA type     5 for PDETECT type</pre>
--	--

---

## :WAVeform:SOURce

<p><b>Command</b>  <b>Query</b>  <b>Returned Format</b>  <b>Where:</b></p>	<pre>:WAVeform:SOURce {CHANnel&lt;n&gt;   WMEMory{1   2   3   4}} :WAVeform:SOURce? [:WAVeform:SOURce] {CHANnel&lt;n&gt;   WMEMory{1   2   3   4}}&lt;NL&gt; &lt;n&gt; ::= integer 1 or 2 (HP 54520A/54522A), or 1 through 4     (HP 54540A/54542A)</pre>
--	---

---

## :WAVeform:TYPE?

<p><b>Query</b>  <b>Returned Format</b></p>	<pre>:WAVeform:TYPE? [:WAVeform:TYPE] {INValid   AVERAge   ENVelope   NORMAl       PDETECT   RAWData}&lt;NL&gt;</pre>
---	---

---

## :WAVeform:XINCrement?

<p><b>Query</b>  <b>Returned Format</b>  <b>Where:</b></p>	<pre>:WAVeform:XINCrement? [:WAVeform:XINCrement] &lt;value&gt;&lt;NL&gt; &lt;value&gt; ::= exponential, x-increment in the current preamble</pre>
--	--

---

### **:WAVeform:XORigin?**

Query :WAVeform:XORigin?  
 Returned Format [ :WAVeform:XORigin ] <value>[,<value>]...<NL>  
 Where: <value> ::= exponential, x-origin value in the current preamble

---

### **:WAVeform:XREFerence?**

Query :WAVeform:XREFerence?  
 Returned Format [ :WAVeform:XREFerence ] <value><NL>  
 Where: <value> ::= integer, x-reference value in the current preamble

---

### **:WAVeform:YINCrement?**

Query Syntax: :WAVeform:YINCrement?  
 Returned Format [ :WAVeform:YINCrement ] <value><NL>  
 Where: <value> ::= exponential, y-increment value in the current preamble

---

### **:WAVeform:YORigin?**

Query :WAVeform:YORigin?  
 Returned Format [ :WAVeform:YORigin ] < value><NL>  
 Where: <value> ::= exponential, y-origin in the current preamble

---

### **:WAVeform:YREFerence?**

Query :WAVeform:YREFerence?  
 Returned Format [ :WAVeform:YREFerence ] <value><NL>  
 Where: <value> ::= integer, y-reference value in the current preamble

---

### **:WMEMory{1 | 2 | 3 | 4}:DISPlay**

Command :WMEMory{1 | 2 | 3 | 4}:DISPlay {{OFF | 0} | {ON | 1}}  
 Query :WMEMory{1 | 2 | 3 | 4}:DISPlay?  
 Returned Format [ :WMEMory{1 | 2 | 3 | 4}:DISPlay ] {0 | 1}<NL>

---

---

### **:WMEMemory{1 | 2 | 3 | 4}:GET**

**Command** :WMEMemory{1 | 2 | 3 | 4}:GET {CHANnel<n> | WMEMemory{1 | 2 | 3 | 4} | FUNCTION{1 | 2 | 3 | 4}}

**Where:** <n> ::= integer 1 or 2 (HP 54520A/54522A), or 1 through 4 (HP 54540A/54542A)

---

### **:WMEMemory{1 | 2 | 3 | 4}:PROTECT**

**Command** :WMEMemory{1 | 2 | 3 | 4}:PROTECT {{OFF | 0} | {ON | 1}}

**Query** :WMEMemory{1 | 2 | 3 | 4}:PROTECT?

**Returned Format** [:WMEMemory{1 | 2 | 3 | 4}:PROTECT] {0 | 1}<NL>

---

### **:WMEMemory{1 | 2 | 3 | 4}:SETup?**

**Query** :WMEMemory{1 | 2 | 3 | 4}:SETup?

**Returned Format** :WME{1 | 2 | 3 | 4}:DISP {0 | 1};  
 PROT {0 | 1};  
 XOFF <offset\_argument>;  
 XRANG <range\_argument>;  
 YOFF <yoffset>;  
 YRANG <yrange><NL>

**Where:** <range\_argument> ::= exponential, 5 ns to 50 sec  
 <offset\_argument> ::= exponential, time from trigger to display reference in seconds  
 <yoffset> ::= exponential, offset value in volts  
 <yrange> ::= exponential, full-scale range value

---

### **:WMEMemory{1 | 2 | 3 | 4}:XOFFset**

**Command** :WMEMemory{1 | 2 | 3 | 4}:XOFFset <xoffset\_argument>

**Query** :WMEMemory{1 | 2 | 3 | 4}:XOFFset?

**Returned Format** [:WMEMemory{1 | 2 | 3 | 4}:XOFFset] <xoffset\_argument><NL>

**Where:** <xoffset\_argument> ::= exponential, time from trigger to the on screen delay reference point. The maximum value depends on the :WMEMemory:XRANGe setting

---

---

## **:WMEMory{1 | 2 | 3 | 4}:XRANge**

**Command**  
**Query**  
**Returned Format**  
**Where:**

```
:WMEMory{1 | 2 | 3 | 4}:XRANge <range_argument>
:WMEMory{1 | 2 | 3 | 4}:XRANge?
[:WMEMory{1 | 2 | 3 | 4}:XRANge] <range_argument><NL>
<range_argument> ::= exponential, 5 ns to 50 s in a 1,2,5
sequence
```

---

## **:WMEMory{1 | 2 | 3 | 4}:YOFFset?**

**Query**  
**Returned Format**  
**Where:**

```
:WMEMory{1 | 2 | 3 | 4}:YOFFset?
[:WMEMory{1 | 2 | 3 | 4}:YOFFset] <yoffset><NL>
<yoffset> ::= exponential, offset value in volts
```

---

## **:WMEMory{1 | 2 | 3 | 4}:YRANge?**

**Query**  
**Returned Format**  
**Where:**

```
:WMEMory{1 | 2 | 3 | 4}:YRANge?
[:WMEMory{1 | 2 | 3 | 4}:YRANge] <yrange><NL>
<yrange> ::= exponential, full-scale range value
```

© Copyright Hewlett-Packard Company 1993-1995  
All Rights Reserved.

Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

#### Document Warranty

The information contained in this document is subject to change without notice.

**Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.**

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

#### Safety

This apparatus has been designed and tested in accordance with IEC Publication 348, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

#### Warning

- Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.
- Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock or fire hazard.

- Service instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

- If you energize this instrument by an auto transformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.

- Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.

- Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

- Do not install substitute parts or perform any unauthorized modification to the instrument.

- Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.

- Use caution when exposing or handling the CRT. Handling or replacing the CRT shall be done only by qualified maintenance personnel.

#### Safety Symbols



Instruction manual symbol: the product is marked with this symbol when it is necessary for you to refer to the instruction manual in order to protect against damage to the product.



Hazardous voltage symbol.



Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

#### WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

#### CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.



## **Product Warranty**

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of three years from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument.

Hewlett-Packard does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

## **Limitation of Warranty**

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

**No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.**

## **Exclusive Remedies**

The remedies provided herein are the buyer's sole and exclusive remedies.

Hewlett-Packard shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

## **Assistance**

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales Office.

## **Certification**

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

## **About this edition**

This is the first edition of the *HP 54520 and 54540 Series Oscilloscope Programmer's Quick Reference Guide*.

Publication number  
54542-97037

Printed in USA.

Edition dates are as follows:

First edition, November 1995

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by you. The dates on the title page change only when a new edition is published.

A software or firmware code may be printed before the date. This code indicates the version level of the software or firmware of this product at the time the manual or update was issued. Many product updates do not require manual changes; and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

The following list of pages gives the date of the current edition and of any changed pages to that edition.

All pages original edition