



**TELEDYNE LECROY**  
Everywhereyoulook™



# WaveStation 2000/3000 SCPI Reference Manual



## **WaveStation 2000/3000SCPI Reference Manual**

© 2013 Teledyne LeCroy, Inc. All rights reserved.

Unauthorized duplication of Teledyne LeCroy documentation materials other than for internal sales and distribution purposes is strictly prohibited. However, clients are encouraged to distribute and duplicate Teledyne LeCroy documentation for their own internal educational purposes.

WaveStation and Teledyne LeCroy are registered trademarks of Teledyne LeCroy, Inc. Windows is a registered trademark of Microsoft Corporation. Other product or brand names are trademarks or requested trademarks of their respective holders. Information in this publication supersedes all earlier versions. Specifications are subject to change without notice.

922829 Rev C  
November 2013

# TABLE OF CONTENTS

<b>Introduction to the WaveStation SCPI.....</b>	<b>3</b>
How Listings Are Organized .....	3
Command Notation .....	3
Table of Commands & Queries .....	4
<b>Configuring for Remote Control.....</b>	<b>5</b>
GPIB.....	5
USB.....	5
LabView Driver.....	6
<b>Program Messages.....</b>	<b>7</b>
Command/Query Syntax.....	7
Header Path .....	7
Header.....	8
Data Parameters .....	8
<b>SCPI Command Reference .....</b>	<b>10</b>
*IDN? - Identification Query .....	10
*OPC - Operation Complete Command.....	10
*CLS – Clear Status Command .....	11
*ESE – Event Status Register Command .....	11
*ESR? – Event Status Register Query.....	11
*RST – Reset Command .....	12
*SRE – Service Request Enable Register Command .....	12
*STB? – Status Register Query.....	13
*TST? – Test Query .....	13
*WAI – Wait Command .....	13
ARWV - Arbitrary Wave Command.....	14
BSWV - Basic Wave Command.....	15
BTWV - Burst Wave Command .....	17
BUZZ - Buzzer Command.....	19

## WaveStation SCPI

---

CHDR - Command Header Command .....	19
INVT - Phase Command .....	19
MDWV - Modulate Wave Command .....	20
OUTP - Output Command .....	23
PACP - Channel Parameter Copy Command .....	23
ROSC - Clock Source Command .....	24
SCFG - Configuration Command .....	24
SCSV - Screen Save Command .....	24
STL? - Store List Query .....	25
SWWV - Sweep Wave Command.....	25
SYNC - Synchronize Command.....	27
VKEY - Virtual Key Command .....	27
WVCSS - Get .CSV File Command .....	29

# Introduction to the WaveStation SCPI

This manual describes the Standard Commands for Programmable Instruments (SCPI) that may be used to control the Teledyne LeCroy WaveStation® 2000 and WaveStation 3000 Series Function/Arbitrary Waveform Generators. All commands and queries can be executed locally or remotely. Most commands can be sent as queries to retrieve current settings.

## How Listings Are Organized

The commands are listed in alphabetical order according to their short form. Each command or query is described with syntax, valid arguments, and examples. The command name (header) is given in short form at the top of the page. Queries are recognized by the question mark (?) following the header.

A brief explanation of the function performed is given, followed by a presentation of the formal syntax with the long form header given in upper- and lower-case characters and the short form derived from it in all upper-case characters. Either form may be used in commands and queries.

**IMPORTANT:** Tables list command data parameters and valid values where there are many. Parameters are listed in the order in which they are functionally processed for convenience.

Where applicable, the query syntax also is given with the format of its response.

In some cases, examples are given at the bottom of descriptions, following the lists of values.

## Command Notation

The following notation is used in this manual to describe the command syntax:

- < > Angular brackets enclose words that are used as placeholders, of which there are two types: the header path and the data parameter of a command.
- := A colon followed by an equals sign separates a placeholder from the description of the values that may be used in a command instead of the placeholder.
- { } Braces enclose lists of values, one of which must be selected.
- [ ] Square brackets enclose optional items.
- ... An ellipsis indicates that the items both to its left and right may be repeated a number of times.

The letters used in the short command format are capitalized in the long format (e.g., BaSic\_WaVe), although they do not need to be capitalized in this fashion when writing commands.

## Table of Commands & Queries

Short	Long	Subsystem	What It Does
*IDN?	*IDN	SYSTEM	Retrieves device identification information.
*OPC	*OPC	SYSTEM	Sets the Event Status Register (ESR) OPC bit to TRUE (1).
*CLS	*CLS	SYSTEM	Clears all status data registers
*ESE	*ESE	SYSTEM	Sets the Standard Event Status Enable register (ESE)
*ESR?	*ESR?	SYSTEM	Reads and clears the contents of the Event Status Register (ESR)
*RST	*RST	SYSTEM	Initiates a device reset.
*SRE	*SRE	SYSTEM	Sets the Service Request Enable register (SRE)
*STB?	*STB?	SYSTEM	Reads the contents of the 488.1 defined status register (STB), and the Master Summary Status (MSS)
*TST	*TST	SYSTEM	Performs an internal self-test.
*WAI	*WAI	SYSTEM	Wait to continue command.
ARWV	ARBWAVE	SYSTEM	Sets the instrument to an arbitrary waveform or retrieves Arbitrary Waveform settings.
BSWV	BASIC_WAVE	SIGNAL	Sets or retrieves basic wave parameters.
BTWV	BURSTWAVE	SIGNAL	Sets instrument to a burst waveform or retrieves current Burst Wave settings.
BUZZ	BUZZER	SYSTEM	Sets or retrieves buzzer status.
CHDR	COMM_HEADER		Sets or retrieves the query return format.
INVT	INVERT	SIGNAL	Sets or retrieves the phase of the output signal.
MDWV	MODULATEWAVE	SIGNAL	Sets instrument to a modulated waveform or retrieves current Modulate Wave settings.
OUTP	OUTPUT	SIGNAL	Sets or retrieves output state.
PACP	CHANNEL_COPY	SIGNAL	Copies parameters from one channel to the other
ROSC	ROSCILLATOR	SIGNAL	Sets or retrieves the clock source.
SCFG	SYSTEM_CONFIG	SYSTEM	Sets or retrieves the state used (last or default) when powering on the WaveStation.
SCSV	SCREEN_SAVE	SYSTEM	Sets screen saver on/off or retrieves screen saver status.

Short	Long	Subsystem	What It Does
STL	STORE_LIST	SIGNAL	Retrieves all waveform names stored in WaveStation's device memory.
SWWV	SWEEP	SIGNAL	Sets instrument to sweep a waveform or retrieves Sweep Wave settings.
SYNC	SYNC	SIGNAL	Sends a Sync pulse upon occurrence of the specified function.
VKEY	VIRTUAL_KEY	SYSTEM	Sends equivalent keyboard function to device.
WVCSS	WAVE_CSV		Saves .CSV file to user-defined memory location.

## Configuring for Remote Control

Remote control is accomplished through the exchange of program messages between the WaveStation and a controller computer using the selected interface:

- The GPIB interface utilizes standard IEEE 488.1 and IEEE 488.2 messages.
- The USB interface utilizes Teledyne LeCroy's VICP protocol, which emulates IEEE 488.2 and includes standard operation bits in a header defined by the VICP protocol.

### GPIB

Install the standard WaveStation USB2-GPIB adapter. Follow the rules in Program Messages when writing your control files.

### USB

Install Teledyne LeCroy's 'VICP Passport' and NI-VISA on the controller machine. NI-VISA contains the USB drivers needed to form the interface between the WaveStation and the controller. NI-VISA can be downloaded free from [www.ni.com/visa](http://www.ni.com/visa). You may use either the NI-VISA run-time or the full download. The run-time download is significantly smaller.

The VICP Passport extends VISA to support Teledyne LeCroy's VICP protocol. The VICP passport may be downloaded from [www.teledynelecroy.com](http://www.teledynelecroy.com). Also see the application brief LAB\_WM827 Understanding VICP and the VICP Passport on Teledyne LeCroy's website for more information.

### LabView Driver

Either the USB or GPIB interface can be utilized with the free LabView Driver to remote control the WaveStation using LabView projects. To install the driver:

1. Download the driver from [teledynelecroy.com/support/softwaredownload/](http://teledynelecroy.com/support/softwaredownload/) under Waveform Generator Downloads > Software Utilities.
2. Copy the .zip archive to **C:\Program Files 86\ NI\LabView2012\instr.lib** on the PC hosting LabView.
3. Unzip the archive and open the **.lvproj** file. There are three example LabView projects in this folder that you can use to set up your own LabView workflow.
4. In the LabView software:
  - Select **USB** or **GPIB** interface, depending on your connection type.
  - In **VISA resource**, select the connected WaveStation Unit.

Waveform files saved in the WaveStation PC Software must be saved in **SendWave CSV** format to be exchanged with LabView.

# Program Messages

Program messages are composed of commands or queries separated by semicolons. **Do not use a terminator of any kind, including NULL terminators.**

```
<command/query>; . . . . ;<command/query>
```

## Command/Query Syntax

The general form of a command or a query is an optional **header path**, followed by a command **header**, optionally followed by one or several **parameters** (shown as **<data>** in the following construct):

```
[header_path:]<header>[?] [<data>, . . . , <data>]
```

**NOTE:** There is a space between the header and the first parameter.

Commas separate parameters.

The question mark **[?]** is optional and turns the command into a query.

WaveStation does not distinguish between upper- and lowercase characters.

## Header Path

Commands or queries that apply to a subsection of the WaveStation, such as a single input channel, must have their headers prefixed with a path name indicating the recipient of the command. The header path normally consists of an abbreviated path name followed by a colon (:) immediately preceding the command header.

C2:ARWV NAME,ATAN

Header Path Name	WaveStation Reference
C1, C2	Channels 1 and 2
M1 to Mxx	Memory Units 1 through 59 (WS 2000) or 67 (WS 3000)

We have included the header path in the command syntax in this manual for convenience.

## Header

The header is the mnemonic form of the operation to be performed by the oscilloscope. Most headers have a more easily recognized **long form** and a **short form** for better transfer and decoding speed. The two can be used interchangeably:

Comm\_HeaDeR is equivalent to CHDR

Some command or query mnemonics are imposed by the IEEE 488.2 standard. All these mnemonics begin with an asterisk \*. They comprise the first group of commands listed in this manual.

## Data Parameters

When a command uses additional data values, they are expressed as ASCII data that can take the form of character, numeric, or block data. Macro parameters are not implemented.

### ***Character Data***

These are simple alphanumeric words or abbreviations indicating a specific action.

In commands where you can specify many parameters, or where not all parameters are applicable at the same time, the format requires pairs of character data values. The first value names the parameter to be modified, while the second gives its value.

C1:BSWV TYPE,SINE,FRQ,1000,AMP,3,OFST,3

### ***Numeric Data***

The numeric data type is used to enter quantitative information. Numbers can be entered as integers, fractions, or exponents:

C2:BURSTWAVE STATE,ON,DLAY,2.4e-07S

Numeric values can be followed with multipliers and units modifying the value of the numeric expression. The following table of mnemonics is recognized by WaveStation:

Multiplier	Exponential Notation	Suffix
EX	1E18	Exa-
PE	1E15	Peta-
T	1E12	Tera-
G	1E9	Giga-
MA	1E6	Mega-
K	1E3	Kilo-
M	1E-3	milli-
U	1E-6	micro-
N	1E-9	nano-
P	1E-12	pico-
F	1E-15	femto-
A	1E-18	atto-

### **Block Data**

These are binary data values used to transfer waveforms between the WaveStation to the controller.

Buffer size limitations apply to block data:

- If data exceeding the WaveStation 512 byte input buffer limited is received, the excess is discarded.
- If data exceeding the output queue is transmitted, commands that are not interpreted and excess data are discarded.

## SCPI Command Reference

### \*IDN? - Identification Query

<b>DESCRIPTION</b>	The *IDN? query retrieves the instrument's identification data. Response includes manufacturer, model, serial number, software and firmware versions.
<b>SYNTAX</b>	*IDN?
<b>RESPONSE</b>	*IDN <device id>,<model>,<serial number>,<software version>,<firmware version>  <device id>:= WST is used to identify instrument.  <model>:= A model identifier (less than 14 characters).  <serial number>:= A nine- or 10-digit decimal code.  <software version>:= Software version.  <firmware version>:= Two digits giving the major release level followed by a period, then one digit giving the minor release level followed by a period, and a single-digit update level (xx.y.z).
<b>EXAMPLE</b>	Read version information.  *IDN?  Return:  *IDN WST,WaveStation 2052,LCRY0000C11111,1.01.02.28, 02-00-00-21-24

### \*OPC - Operation Complete Command

<b>DESCRIPTION</b>	The *OPC (OPeration Complete) command sets the OPC bit to TRUE in the standard Event Status Register (ESR).  The *OPC? query always responds with the ASCII character 1 because the device only responds to the query when the previous command has executed completely.
<b>SYNTAX</b>	*OPC
<b>QUERY SYNTAX</b>	*OPC?
<b>RESPONSE</b>	*OPC 1
<b>RELATED</b>	*WAI

## \*CLS – Clear Status Command

**DESCRIPTION** The \*CLS command clears all the status data registers.

**SYNTAX** \*CLS

**RELATED** ALL\_STATUS, CMR, DDR, \*ESR, EXR, \*STB, URR

## \*ESE – Event Status Register Command

**DESCRIPTION** The \*ESE command sets the Standard Event Status Enable register (ESE). This command allows one or more events in the ESR register to be reflected in the ESB summary message bit(bit 5) of the STB register.

The \*ESE? query reads the contents of the ESE register.

**SYNTAX** \*ESE <value>

<value> := 0 to 255

**QUERY SYNTAX** \*ESE?

**RESPONSE** \*ESE <value>

**EXAMPLE 1** The following instruction allows the ESB bit to be set if a user request (URQ bit 6, i.e. decimal 64) and/or a device dependent error (DDE bit 3, i.e., decimal 8) occurs. Summing these values yields the ESE register mask  $64+8=72$ .

\*ESE?

Return:

\*ESE 72

**RELATED** \*ESR

## \*ESR? – Event Status Register Query

**DESCRIPTION** The \*ESR? query reads and clears the contents of the Event Status Register (ESR). The response represents the sum of the binary values of the register bits 0 to 7.

**SYNTAX** \*ESR?

**RESPONSE** \*ESR <value>

<value> := 0 to 255

**RELATED** ALL\_STATUS, \*CLS, \*ESE

## \*RST – Reset Command

<b>DESCRIPTION</b>	The *RST command initiates a device reset. The *RST sets all eight traces to the GND line and recalls the default setup.
<b>SYNTAX</b>	*RST
<b>RELATED</b>	*CAL, *RCL

## \*SRE – Service Request Enable Register Command

<b>DESCRIPTION</b>	The *SRE command sets the Service Request Enable register (SRE). This command allows the user to specify which summary message bit(s) in the STB register will generate a service request.  A summary message bit is enabled by writing a '1' into the corresponding bit location. Conversely, writing a '0' into a given bit location prevents the associated event from generating a service request (SRQ). Clearing the SRE register disables SRQ interrupts.  The *SRE? query returns a value that, when converted to a binary number, represents the bit settings of the SRE register. Note that bit 6 (MSS) cannot be set and its returned value is always zero.
<b>SYNTAX</b>	*SRE <value>  <value> : = 0 to 255
<b>QUERY SYNTAX</b>	*SRE?
<b>RESPONSE</b>	*SRE <value>
<b>EXAMPLE 1</b>	The following instruction allows an SRQ to be generated as soon as the MAV summary bit (bit 4, i.e. decimal 16) or the INB summary bit (bit 0, i.e., decimal 1) in the STB register, or both, are set. Summing these two values yields the SRE mask $16+1 = 17$ .  *SRE?  Return:  *SRE 17

## \*STB? – Status Register Query

<b>DESCRIPTION</b>	The *STB? query reads the contents of the 488.1 defined status register (STB), and the Master Summary Status (MSS). The response represents the values of bits 0 to 5 and 7 of the Status Byte register and the MSS summary message.
	The response to a *STB? query is identical to the response of a serial poll except that the MSS summary message appears in bit 6 in place of the RQS message.
<b>SYNTAX</b>	*STB?
<b>RESPONSE</b>	*STB <value> <value> := 0 to 255
<b>RELATED</b>	ALL_STATUS, *CLS, *PRE, *SRE

## \*TST? – Test Query

<b>DESCRIPTION</b>	The *TST? query performs an internal self-test, the response indicating whether the self-test has detected any errors.
<b>SYNTAX</b>	*TST?
<b>RESPONSE</b>	*TST {0 1}
	Where:
	0 = self-test successful
	1 = self-test failed
<b>RELATED</b>	*CAL

## \*WAI – Wait Command

<b>DESCRIPTION</b>	The *WAI (WAIt to continue) command is required by the IEEE 488.2 standard but has no effect on the WaveStation, as the signal generator only starts processing a command when the previous command has been entirely executed.
<b>SYNTAX</b>	*WAI
<b>RELATED</b>	*OPC

## ARWV - Arbitrary Wave Command

**DESCRIPTION** Changes the arbitrary waveform type.

**SYNTAX** <channel>:ARbWaVe { INDEX | NAME }, <indexvalue | namevalue>

<channel>:= {C1 | C2}

<indexvalue>:= {number from tables below}

<namevalue>:= {name from the tables below}

**IMPORTANT:** Use only one parameter, either INDEX or NAME, in a single Arbitrary Wave command.

**QUERY SYNTAX** <channel>:ARbWaVe?

<channel>:= {C1 | C2}

**RESPONSE** <channel>:ARbWaVe INDEX, <value>, NAME, <value>

**EXAMPLE 1** Set StairUp arbitrary waveform output by index.

C1:ARWV INDEX, 2

**EXAMPLE 2** Set Atan arbitrary wave output by name.

C2:ARWV NAME, ATAN

**EXAMPLE 3** Read current arbitrary waveform type.

C1:ARWV?

Return:

C1:ARBWAVE INDEX, 2, NAME, StairUp

**RELATED** BSWV

### WaveStation 2000 Arbitrary Waveform Values

Index	Name	Index	Name	Index	Name	Index	Name	Index	Name
2	StairUp	11	Exp_Rise	20	Dlorentz	29	Twotone	41	Bartlett
3	StairDn	12	LogFall	21	Haversine	30	Snr	42	Tan
4	StairUD	13	LogRise	22	Lorentz	34	Hamming	43	Cot
5	PPulse	14	Sqrt	23	Gauspuls	35	Hanning	44	Sec
6	NPulse	15	Root3	24	Gmonopuls	36	Kaiser	45	Csc
7	Trapezia	16	X^2	25	Tripuls	37	Blackman	46	Asin
8	UpRamp	17	X^3	26	Cardiac	38	Gausswin	47	Acos
9	DnRamp	18	Sinc	27	Quake	39	Triang	48	Atan
10	Exp_Fall	19	Gussian	28	Chirp	40	BlackmanHarris	49	Acot

Indices 50-60 are reserved for user-defined waveforms.

**WaveStation 3000 Arbitrary Waveform Values**

Index	Name	Index	Name	Index	Name	Index	Name	Index	Name
0	StairUp	8	Sqrt	16	Gmonopuls	24	Blackman	32	Asin
1	StairDn	9	X^2	17	Cardiac	25	GaussiWin	33	Acos
2	StairUD	10	Sinc	18	Quake	26	Harris	34	Atan
3	Trapezia	11	Gaussian	19	TwoTone	27	Bartlett	35	ACot
4	ExpFall	12	Dlorentz	20	SNR	28	Tan		
5	ExpRise	13	Haversine	21	Hamming	29	Cot		
6	LogFall	14	Lorentz	22	Hanning	30	Sec		
7	LogRise	15	Gauspuls	23	Kaiser	31	Csc		

Indices 36-59 are reserved for user-defined waveforms up to 16K; indices 60-67 are for user-defined waveforms up to 512K.

**BSWV - Basic Wave Command**

**DESCRIPTION** Sets or retrieves basic waveform parameters.

**SYNTAX** <channel>:BaSicWaVe <parameter>

<channel>:= {C1|C2}

<parameter>:= {a parameter from the table below}

**IMPORTANT:** If the command doesn't specify wave type (WVTP), the remaining parameters are applied to the waveform type already selected on your WaveStation.

Parameters	Value	Description
WVTP	<wave type>	Where <wave type>:= {SINE, SQUARE, RAMP, PULSE, NOISE, ARB ,DC}
FRQ	<frequency>	Where <frequency>:= {default unit is Hz; minimum value 1e-6 Hz, maximum value depends on the model}. If WVTP is NOISE, parameter is invalid.
AMP	<amplifier>	Where <amplifier>:= {default unit is V; WS2000 Channel 1 min. 0.004V, max. 6V; Channel 2 min. 0.004V, max. 20V WS3000 Channel 1 and 2 min. 2mVpp, max. 20Vpp}. If WVTP is NOISE, parameter is invalid.
OFST	<offset>	Where <offset>:= {default unit is V; maximum value depends on the WaveStation model}. If WVTP is NOISE, parameter is invalid.
SYM	<symmetry>	Where <symmetry> :={0 to 100}. Parameter valid only when WVTP is RAMP.
DUTY	<duty cycle>	Where <duty cycle>:= {if WVTP is SQUARE, 20% to 80%; if WVTP is PULSE, 0.1% to 99.9%}.
PHSE	<phase>	Where <phase>:= {0 to 360}. If WVTP is NOISE, parameter is invalid.
STDEV	<standard deviation>	Where <standard deviation>:= {default unit is V; min. 0.5mV; WS2000 max. 1.625V, WS3000 max. 1.599V}. Parameter is valid only when WVTP is NOISE.

## WaveStation SCPI

---

Parameters	Value	Description
MEAN	<mean>	Parameter is valid only when WVTP is NOISE. Where <mean>:= {default unit is V; range depends on <standard deviation>}.
WIDTH	<width>	Value of width. Parameter is valid only when WVTP is PULSE.
RISE	<rise>	Value of rise. Parameter is valid only when WVTP is PULSE.
FALL	<fall>	Value of fall. Parameter is valid only when WVTP is PULSE.
DLY	<delay>	Where <delay>:= {default unit is S; minimum 0, maximum is Pulse Period}. Parameter is valid only when WVTP is PULSE.

**QUERY SYNTAX**    <channel>:BaSicWaVe?

                  <channel>:= {C1|C2}

**RESPONSE**    <channel>:BSWV <type>, <frequency>, <offset>, <amplifier>, <duty>, <symmetry>, <phase>

**EXAMPLE 1**    Set Channel 1 basic wave type to Ramp.

C1:BSWV WVTP ,RAMP

**EXAMPLE 2**    Set Channel 1 signal frequency to 2000Hz.

C1:BSWV FRQ,2000HZ

**EXAMPLE 3**    Retrieve Channel 1 Basic Wave settings.

C1:BSWV?

Return:

C1:BSWV TYPE,SINE,FRQ,1000,AMP,3,OFST,3

**RELATED**    ARWV, BTWV, CFG, CPL, MDWV, SWWV

## BTWV - Burst Wave Command

**DESCRIPTION** Sets waveform type to burst or retrieves Burst Wave settings.

**SYNTAX**

```
<channel>:BurstWaVe STATE,ON,<parameter>,<value>,...
```

<channel>:= {C1|C2}

<parameter>:= {a parameter from the table below}

<value>:= {a value from the table below}

**IMPORTANT:** Set STATE ON before setting other parameters. Specify carrier wave type before listing other CARR parameters (e.g., C1:BTWV STATE,ON,CARR,WVTP,<value>....). Once set, you do not need to repeat CARR; it is shown here to indicate which parameters can follow it.

Parameters	Value	Description
STATE	{ON OFF}	To set or read other Burst Wave parameters, first set STATE ON.
PRD	<period>	Where <period>:= {default unit is S; WS2000 1µs to 500S; WS3000 1µs to 1000S}. Parameter is invalid when CARR,WVTP is NOISE or GATE_NCYC is GATE. Parameter is valid only when TRSR is INT.
STPS	<start phase>	Where <start phase>:= {0 to 360}. Parameter is invalid when CARR,WVTP is NOISE or PULSE.
GATE_NCYC	{GATE NCYC}	Configures burst waveform either gated or N-cycled. Parameter is invalid when CARR,WVTP is NOISE.
TRSR	{EXT, INT, MAN}	Sets trigger source EXTERNAL, INTERNAL, or MANUAL. Parameter is valid only when GATE_NCYC is NCYC.
DLAY	<delay>	Where <delay>:= {default unit is S; min. 0; max. depends on the Burst period}. Parameter is valid only when GATE_NCYC is NCYC.
PLRT	{NEG POS}	Sets polarity NEGATIVE or POSITIVE. Parameter is valid only when GATE_NCYC is GATE.
TRMD	<trigger mode>	Where <trigger mode>:= {RISE, FALL, OFF}. Parameter is invalid when TRSR is EXT. Parameter is valid only when GATE_NCYC is NCYC.
EDGE	{RISE FALL}	Use rising or falling edge. Parameter is invalid when CARR,WVTP is NOISE. Parameter is valid only when GATE_NCYC is NCYC.
TIME	<cycle time>	Where <cycle time>:= {1 to 50000}. Parameter is valid only when GATE_NCYC is NCYC.
MTRIG		Triggers the device once. Parameter is valid only when TRSR is MAN.
CARR,WVTP	<wave type>	Where carrier <wave type>:= {SINE, SQUARE, RAMP, PULSE, NOISE, ARB}. CARR,WVTP cannot be NOISE if GATE_NCYC is NCYC.
(CARR,)FRQ	<frequency>	Where carrier wave <frequency>:= {default unit is Hz; minimum 1xe-6 Hz, maximum depends on the WaveStation model}.

## WaveStation SCPI

---

Parameters	Value	Description
(CARR,)AMP	<amplifier>	Where carrier wave <amplifier>:= {default unit is V; WS2000 Channel 1 min. 0.004V max. 6V, Channel 2 min. 0.004V max. 20V WS3000 Channel 1 and 2 min. 2mVpp max. 20Vpp}.
(CARR,)OFST	<offset>	Where carrier wave <offset>:= {default unit is V; maximum value depends on the WaveStation model}.
(CARR,)SYM	<symmetry>	Where carrier wave <symmetry>:= {0% to 100%}.
(CARR,)DUTY	<duty cycle>	Where carrier wave <duty cycle>:= {if CARR,WVTP is SQUARE, 20% to 80%; if CARR,WVTP is PULSE, 0.1% to 99.9%}.
(CARR,)PHSE	<phase>	Where carrier wave <phase>:= {0 to 360}.
(CARR,)RISE	<rise>	Value of carrier wave rising edge. Parameter valid only when CARR,WVTP is PULSE.
(CARR,)FALL	<fall>	Value of carrier wave falling edge. Parameter valid only when CARR,WVTP is PULSE.
(CARR,)DLY	<delay>	Where carrier wave <delay>:= {default unit is S; minimum 0, maximum is value of Pulse Period}. Parameter is valid only when CARR,WVTP is PULSE.
(CARR,)STDEV	<standard deviation>	Parameter is valid only when CARR,WVTP is NOISE.
(CARR,)MEAN	<mean>	Parameter is valid only when CARR,WVTP is NOISE.

**QUERY SYNTAX**      <channel>:BurstWaVe?

**RESPONSE**      <channel>:BurstWaVe <type>,<state>,<period>

**EXAMPLE 1**      Set Channel 1 burst waveform period to one second.

C1:BTWV PRD,1S

**EXAMPLE 2**      Read Channel 2 burst waveform parameters (and STATE is ON).

C2:BTWV?

Return:

```
C2:BURSTWAVE STATE,ON,PRD,0.01S,STPS,0,TRIG,IN,TRMD,OFF,
TIME,1,DLAY,2.4e-07S,GATE_NCYC,NCYC,PLRT,NEG,CARR,WVTP,SINE,
FRQ,1000HZ,AMP,4V,OFST,0V,PHSE,0
```

**EXAMPLE 3**      Read Channel 2 burst wave parameters (and STATE is OFF).

C2:BTWV?

Return:

```
C2:BURSTWAVE STATE,OFF
```

## BUZZ - Buzzer Command

**DESCRIPTION** Turns on/off the instrument's buzzer.

**SYNTAX** BUZZer {ON|OFF}

**QUERY SYNTAX** BUZZer?

**RESPONSE** BUZZer {ON|OFF}

## CHDR - Command Header Command

**DESCRIPTION** Changes the command/query return format. SHORT returns the short format; LONG returns the long format; OFF retrieves no query results.

**SYNTAX** Comm\_HeaDeR {SHORT, LONG, OFF}

**QUERY SYNTAX** Comm\_HeaDeR?

**RESPONSE** CHDR {SHORT, LONG, OFF}

**EXAMPLE 1** Use the long format when returning data.

CHDR LONG

**EXAMPLE 2** Read current command format.

CHDR?

Return:

COMM\_HEADER LONG

## INVT - Phase Command

**DESCRIPTION** Sets phase or retrieves current Phase settings.

**SYNTAX** INVerT {ON|OFF}

**QUERY SYNTAX** INVerT?

**RESPONSE** INVERT {ON|OFF}

## MDWV - Modulate Wave Command

**DESCRIPTION** Sets a modulated waveform or retrieves current Modulate Wave settings.

**SYNTAX**

```
<channel>:MoDulateWaVe <parameter>,<value>,...
```

<channel>:= {C1|C2}

<parameter>:= {a parameter from the table below }

<value>:- {a value from the table below)

**IMPORTANT:** This command cannot be used if Basic wave type is NOISE. To configure wave modulation, first set STATE ON and specify AM, DSBAM, FM, PM, PWM, ASK, FSK or CARR before setting related parameters (e.g., C1:MDWV STATE,ON,FM,<parameter>,<value>). Once set, you do not need to repeat AM, DSBAM, FM, PM, PWM, ASK, FSK or CARR; it is shown here to indicate which parameters can follow it.

Parameters	Value	Description
STATE	{ON OFF}	Turns on/off wave modulation. First set STATE ON before setting other Modulate Wave parameters.
(AM,)SRC	{INT EXT}	Specifies INTernal or EXTernal AM signal source.
(AM,)MDSP	<wave shape>	Where AM <wave shape>:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}. Parameter is valid only when AM,SRC is INT.
(AM,)FRQ	<frequency>	Where AM <frequency>:= {WS2000 0.002Hz to 20000Hz; WS3000 0.001Hz to 50000Hz}. Parameter is valid only when AM,SRC is INT.
(AM,)DEPTH	<depth>	Where AM <depth>:= {0 to 120}. Parameter is valid only when AM,SRC is INT.
(DSBAM,)SRC	{INT EXT}	Specifies INTernal or EXTernal DSBAM signal source.
(DSBAM,)MDSP	<wave shape>	Where DSBAM <wave shape>:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}. Parameter is valid only when DSBAM,SRC is INT.
(DSBAM,)FRQ	<frequency>	Where DSBAM wave <frequency>:= {default unit is Hz; minimum is 1xe-6 Hz, maximum depends on the WaveStation model}. Parameter is valid only when DSBAM,SRC is INT.
(FM,)SRC	{INT EXT}	Specifies internal or external FM signal source.
(FM,)MDSP	<wave shape>	Where FM <wave shape>:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}. Parameter is valid only when FM,SRC is INT.
(FM,)FRQ	<frequency>	Where FM <frequency>:= {WS2000 0.002Hz to 20000Hz; WS3000 0.001Hz to 50000Hz} Parameter is valid only when FM,SRC is INT.
(FM,)DEVI	<frequency offset>	Where FM <frequency offset>:= {0 to basic wave frequency/2}. Parameter is valid only when FM,SRC is INT.
(PM,)SRC	{INT EXT}	Specifies internal or external PM signal source.

Parameters	Value	Description
(PM,)MDSP	<wave shape>	Where PM <wave shape>:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}. Parameter is valid only when PM,SRC is INT.
(PM,)FRQ	<frequency>	Where PM <frequency>:= {WS2000 0.002Hz to 20000Hz; WS3000 0.001Hz to 50000Hz}. Parameter is valid only when PM,SRC is INT.
(PM,)DEVI	<phase offset>	Where PM <phase offset>:= {0 to 360}. Parameter is valid only when PM,SRC is INT.
(PWM,)FRQ	<frequency>	Where pulse wave < frequency>:= {0Hz to 4kHz}. Parameter is valid only when CARR,WVTP is PULSE.
(PWM,)DEVI	<duty cycle deviation>	Where pulse wave <duty cycle deviation>:= {depends on carrier waveform duty cycle}. Parameter is valid only when CARR,WVTP is PULSE.
(PWM,)MDSP	<wave shape>	Where pulse <wave shape>:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}. Parameter is valid only when CARR,WVTP is PULSE.
(PWM,)SRC	{INT EXT}	Sets pulse wave signal source to INTernal or EXTernal.
(ASK,)SRC	{INT EXT}	Sets ASK signal source to INTernal or EXTernal.
(ASK,)KFRQ	<ASK key frequency>	Where <ASK key frequency>:= {WS2000 0.002Hz to 20000Hz; WS3000 0.001Hz to 1MHz}. Parameter is valid only when ASK,SRC is INT.
(FSK,)KFRQ	<key frequency>	Where FSK <key frequency>:= {WS2000 0.002Hz to 50000Hz; WS3000 0.001Hz to 1MHz}. Parameter is valid only when FSK,SRC is INT.
(FSK,)HFRQ	<hop frequency>	Where FSK <hop frequency>:= {same as basic waveform frequency}.
(FSK,)SRC	{INT EXT}	Sets FSK signal source to INTernal or EXTernal.
(CARR,)WVTP	<wave type>	Where carrier <wave type>:= {SINE, SQUARE, RAMP, ARB, PULSE}.
(CARR,)FRQ	<frequency>	Where carrier wave <frequency>:= {default unit is Hz; minimum value is 1e-6 Hz, maximum value depends on the WaveStation model}.
(CARR,)AMP	<amplifier>	Where carrier wave <amplifier>:= {default unit is V; Channel 1 minimum 0.004V, maximum 6V; Channel 2 minimum 0.004V, maximum is 20V}.
(CARR,)OFST	<offset>	Where carrier wave <offset>:= {default unit is V}.
(CARR,)SYM	<symmetry>	Where carrier wave <symmetry>:= {0% to 100%}.
(CARR,)DUTY	<duty cycle>	Where carrier wave <duty cycle>:= {if CARR,WVTP is SQUARE, 20% to 80%; if CARR,WVTP is PULSE, 0.1% to 99.9%}.
(CARR,)PHSE	<phase>	Where carrier wave <phase>:= {0 to 360}.
(CARR,)RISE	<rise>	Value of rise.
(CARR,)FALL	<fall>	Value of fall.
(CARR,)DLY	<delay>	Value of delay.

## WaveStation SCPI

---

<b>QUERY SYNTAX</b>	<channel>:MoDulateWaVe? <channel>:= {C1 C2}
<b>RESPONSE</b>	<channel>:MoDulateWaVe <parameter>,<value>,... <parameter>:= {any parameters set for the modulated waveform}
<b>EXAMPLE 1</b>	Set Channel 1 modulation to AM, and AM wave shape to SINE wave. C1:MDWV STATE,ON,AM,MDSP,SINE
<b>EXAMPLE 2</b>	Set Channel 1 FM frequency to 1000Hz. C1:MDWV STATE,ON,FM,FRQ,1000HZ
<b>EXAMPLE 3</b>	Read Channel 1 Modulate Waveform parameters (and STATE is ON). C1:MDWV?  Return: C1:MDWV STATE,ON,AM,MDSP,SINE,SRC,INT,FRQ,100HZ, DEPTH,100,CARR,WVTP,RAMP,FRQ,1000HZ,AMP,4V,OFST,0V,SYM,50
<b>RELATED</b>	ARWV, BTWV, CFG, CPL, SWWV, BSWV

## OUTP - Output Command

**DESCRIPTION** Enables or disables the output connector on the front panel for a specified channel. The query returns a value of ON or OFF.

**SYNTAX**

```
<channel>:OUTPut <parameter>,<value>,...  
<channel>:= {C1|C2}  
<parameter>:= {a parameter from table below}
```

Parameters	Value	Description
ON	---	Turn on channel
OFF	---	Turn off channel
LOAD	{50 HZ}	50 = low-impedance (50 Ohm); HZ = high-impedance(Hi-Z)
PLRT	{NOR INVT}	NOR = normal polarity; INVT = inverted

**QUERY SYNTAX** <channel>:OUTPut?

**RESPONSE** <channel>:OUTPut {ON|OFF},LOAD,<load>

**EXAMPLE 1** Turn on Channel 1 output

```
C1:OUTP ON
```

**EXAMPLE 2** Read Channel 1 output state

```
C1:OUTP?
```

Return:

```
C1:OUTP ON,LOAD,HZ
```

**EXAMPLE 3** Set Channel 1 load to 50 Ohm

```
C1:OUTP LOAD,50
```

## PACP - Channel Parameter Copy Command

**DESCRIPTION** Copies all parameter data from one channel to the other.

**SYNTAX**

```
PAraCoPy <destination channel>,<source channel>  
<destination channel>:= {C1|C2}  
<source channel>:= {C1|C2}
```

**IMPORTANT:** Destination and source must both be set and cannot be the same. If C1 is the destination channel, C2 is the source channel and vice versa.

**RELATED** ARWV, BTWV, CFG, CPL, MDWV, SWWV, CHCT, BSWV

## **ROSC - Clock Source Command**

<b>DESCRIPTION</b>	Sets clock source to INTernal system clock or EXTernal clock, or retrieves the current source.
<b>SYNTAX</b>	ROSCillator {INT EXT}
<b>QUERY SYNTAX</b>	ROSCillator?
<b>RESPONSE</b>	ROSCillator {INT EXT}

## **SCFG - Configuration Command**

<b>DESCRIPTION</b>	Determines whether to use default settings or last saved settings when powering on the WaveStation.
<b>SYNTAX</b>	Sys_CFG {DEFAULT LAST}
<b>QUERY SYNTAX</b>	Sys_CFG?
<b>RESPONSE</b>	Sys_CFG {DEFAULT LAST}

## **SCSV - Screen Save Command**

<b>DESCRIPTION</b>	Sets idle time before turning on screen saver, or disables screen saver. Unit is minutes.
<b>SYNTAX</b>	SCreen_SaVe <value> <value>:= {OFF,1,5,15,30,60,120,300}
<b>QUERY SYNTAX</b>	SCreen_SaVe?
<b>RESPONSE</b>	SCreen_SaVe <value>
<b>EXAMPLE</b>	Turn on screen saver after five minutes of idle time. SCSV 5

## STL? - Store List Query

<b>DESCRIPTION</b>	Retrieves the names of waveforms saved to WaveStation memory units. If a user-defined memory unit (M50-M59 on WaveStation 2000 models and M36-M67 on WaveStation 3000 models) does not have a saved waveform, the query returns the string "EMPTY".
<b>SYNTAX</b>	SToreList?
<b>RESPONSE</b>	(example from WaveStation 2000)

```
STL M0,SINE,M1,NOISE,M2,STAIRUP,M3,STAIRDN,M4,STAIRUD,
M5,PPULSE,M6,NPULSE,M7,TRAPEZIA,M8,UPRAMP,M9,DNRAMP,
M10,EXP_FALL,M11,EXP_RISE,M12,LOGFALL,M13,LOGRISE,M14,SQRT,
M15,ROOT3,M16,x^2,M17,x^3,M18,SINC,M19,GUSSIAN,M20,DLORENTZ,
M21,HAVERSINE,M22,LORENTZ,M23,GAUSPULS,M24,GMONOPULS,
M25,TRIPULS,M26,CARDIAC,M27,QUAKE,M28,CHIRP,M29,TWOTONE,
M30,SNR,M31,EMPTY,M32,EMPTY,M33,EMPTY,M34,HAMMING,
M35,HANNING,M36,KAISER,M37,BLACKMAN,M38,GAUSSWIN,M39,TRIANG,
M40,BLACKMANHARRIS,M41,BARTLETT,M42,TAN,M43,COT,M44,SEC,
M45,CSC,M46,ASIN,M47,ACOS,M48,ATAN,M49,ACOT,M50,EMPTY,
M51,EMPTY,M52,EMPTY,M53,EMPTY,M54,EMPTY,M55,EMPTY,M56,EMPTY,
M57,EMPTY,M58,EMPTY,M59,EMPTY
```

## SWWV - Sweep Wave Command

<b>DESCRIPTION</b>	Sets sweep parameters or retrieves current Sweep Wave settings.
<b>SYNTAX</b>	<pre>&lt;channel&gt;: SWeePWaVe &lt;parameter&gt;,&lt;value&gt;... &lt;channel&gt;:= {C1 C2} &lt;parameter&gt;:= {a parameter from the table below} &lt;value&gt;:= {a value from the table below}</pre>

**IMPORTANT:** Carrier wave type cannot be PULSE or NOISE. Set STATE ON and specify carrier wave type before other CARR parameters (e.g., C1:SWWV STATE,ON,CARR,WVTP,SINE,...). Once set, you do not need to repeat CARR before other parameters; it is shown here to indicate which parameters can follow it.

Parameters	Value	Description
STATE	{ON OFF}	Turns on/off sweep. To set other Sweep Wave parameters, first set STATE ON.
TIME	<time>	Sweep time where <time>:= {0.001S to 500S}.
STOP	<stop frequency>	Where <stop frequency> may be the same as the basic wave frequency.

## WaveStation SCPI

---

Parameters	Value	Description
START	<start frequency>	Where <start frequency> may be the same as the basic wave frequency.
TRSR	{EXT, INT, MAN}	Sets trigger source to EXternal, INTernal, or MANual.
TRMD	{ON OFF}	Turns on/off trigger output. Parameter is invalid if TRSR is EXT.
SWMD	{LINE LOG}	Sets sweep mode to LINEar or LOGarithmic
DIR	{UP DOWN}	Sweep direction.
EDGE	{RISE FALL}	Use rising or falling edge. Parameter is valid only when TRSR is EXT.
MTRIG		Triggers the device once. Parameter is valid only when TRSR is MAN.
CARR,WVTP	<wave type>	Where carrier <wave type>:= {SINE, SQUARE, RAMP, ARB}.
(CARR,)FRQ	<frequency>	Where carrier wave <frequency>:= {default unit is Hz; minimum 1xe-6 Hz, maximum depends on the WaveStation model}.
(CARR,)AMP	<amplifier>	Where carrier wave <amplifier>:= {default unit is V; WS2000 Channel 1 min. 0.004V, max. 6V; Channel 2 min. 0.004V, max. 20V WS3000 Channel 1 and 2 min. 2mVpp, max. 20Vpp}.
(CARR,)OFST	<offset>	Where carrier wave <offset>:= {default unit is V; maximum depends on the WaveStation model}.
(CARR,)SYM	<symmetry>	Where carrier wave <symmetry>:= {0% to 100%}.
(CARR,)DUTY	<duty cycle>	Where carrier wave <duty cycle>:= {20 to 80}. Parameter is valid only when CARR,WVTP is SQUARE.
(CARR,)PHSE	<phase>	Where carrier wave <phase>:= {0 to 360}.

**QUERY SYNTAX**      <channel>:SWEEPWAVE?

                  <channel>:= {C1|C2}

**RESPONSE**      <parameter>:= {all current Sweep Wave parameters}

**EXAMPLE 1**      Set Channel 1 sweep time to one second.

C1:SWWV STATE,ON,TIME,1S

**EXAMPLE 2**      Read Channel 2 Sweep Wave parameters (when STATE is ON).

C2:SWWV?

Return:

C2:SWWV STATE,ON,TIME,1S,STOP,100HZ,START,100HZ,  
TRSR,MAN,TRMD,OFF,SWMD,LINE,DIR,UP,CARR,WVTP,SQUARE,  
FRQ,1000HZ,AMP,4V,OFST,0V,DUTY,50

**EXAMPLE 3**      Read Channel 2 Sweep Wave parameters (when STATE is OFF).

C2:SWWV?

Return:

C2:SWWV STATE,OFF

## SYNC - Synchronize Command

<b>DESCRIPTION</b>	Synchronizes the channel output on the back panel of the instrument to the channel output on the front panel of the instrument.
<b>SYNTAX</b>	<channel>:SYNC {ON OFF} <channel>:= {C1 C2}
<b>QUERY SYNTAX</b>	<channel>:SYNC?
<b>RESPONSE</b>	<channel>:SYNC {ON OFF}

## VKEY - Virtual Key Command

<b>DESCRIPTION</b>	Sends same function to the device as would the corresponding keyboard control. You can send the control name or number in a single command, but not both.
<b>SYNTAX</b>	VirtualKEY VALUE,<value>,STATE,1 <value>:= {a name or number from the table below.}

Name	Number	Name	Number
<b>WaveStation 2000 and WaveStation 3000</b>			
KB_CHANNEL	33	KB_OUTPUT1	153
KB_FUNC1	28	KB_OUTPUT2	152
KB_FUNC2	23	KB_NUMBER_0	48
KB_FUNC3	18	KB_NUMBER_1	49
KB_FUNC4	13	KB_NUMBER_2	50
KB_FUNC5	8	KB_NUMBER_3	51
KB.Utility	11	KB_NUMBER_4	52
KB_HELP	12	KB_NUMBER_5	53
KB_MOD	15	KB_NUMBER_6	54
KB_SWEEP	16	KB_NUMBER_7	55
KB_BURST	17	KB_NUMBER_8	56
KB_KNOB_RIGHT	175	KB_NUMBER_9	57
KB_KNOB_LEFT	177	KB_POINT	46
KB_LEFT	44	KB_NEGATIVE	43
KB_RIGHT	40		

## WaveStation SCPI

---

Name	Number	Name	Number
<b><i>WaveStation 2000 Only</i></b>			
KB_SINE	34	KB_ARB	9
KB_SQUARE	29	KB_STORE_RECALL	10
KB_RAMP	24	KB_UP	45
KB_PULSE	19	KB_DOWN	39
KB_NOISE	14		
<b><i>WaveStation 3000 Only</i></b>			
KB_FUNC6	3	KB_PARAMETER	5
KB_WAVES	4	KB_KNOB_DOWN	176

## WVCSV - Get .CSV File Command

<b>DESCRIPTION</b>	Saves a .CSV format waveform file from an outside source to a user-defined memory location on the WaveStation. The number of data points in the file is limited by your WaveStation model (16k for WaveStation 2000 and 512k for WaveStation 3000).
<b>SYNTAX</b>	<pre>WaVeCSV &lt;address&gt;, WAVENM, &lt;wavename&gt;, CSVLENG, &lt;csvlength&gt;, CSVDATA, &lt;csvdata&gt;</pre> <p>&lt;address&gt;:= {M50 to M59} (WaveStation 2000)  {M36 to M59} (16k waveforms on WaveStation 3000)  {M60 to M67} (512k waveforms on WaveStation 3000)</p> <p>&lt;wavename&gt;:= Name of file on WaveStation</p> <p>&lt;csvlength&gt;:= Number of data points in waveform</p> <p>&lt;csvdata&gt;:= List of Amplitude values (Volts) expressed in decimal format from the .CSV file, using Amplitude as header. Values must be less than the Voltage limit for your WaveStation model.</p>

**TIP:** To avoid manually entering all the values in the .CSV file in the command, use a high-level language such as C or LabView to generate the command automatically from a .CSV file saved on a specific path (e.g., C:\Documents\WaveStation\3000\Wave1.csv). Go to [teledynelecroy.com](http://teledynelecroy.com) for example LabView code.

<b>EXAMPLE</b>	If a .CSV file has the content:  Amplitude,0,0.0012,0.0023,0.0035,...  Then the final SCPI command transferred over USBTMC is:  WAVECSV M36, WAVENM, WAVE1, CSVLENG, 137166, CSVDATA, Amplitude, 0, 0.0012, 0.0023, 0.0035,...
----------------	---



700 Chestnut Ridge Road  
Chestnut Ridge, NY 10977  
USA

[teledynelecroy.com](http://teledynelecroy.com)