

# **Acute Virtual Instrument Software Architecture**

Version: 1.23

Publish: 2020/04/06

## Abstract

AqVISA provides an interface allow user to access the Acute DSO build-in function by using their program send text command through the AqVISA.dll. The commands introduced below is described in capital as abbreviation form, users and either input full text or abbreviation command to control the DSO.

The document applies for: DSO software (1.4.1.55), AqVISA DLL(1.0.0.9)

**Note: Other VISA command functions except AqVISA are not applicable and should not be used to control the Acute DSO.**

## Acute DSO Visa DLL Funtions

### **BOOL PASCAL viOpenRM( HWND hWnd )**

Connect to the DSO software and then enable the VISA function. This function will automatically startup the DSO software if the software is not running.

Parameter

hWnd[In]:

Type: HWND

Handle to the Parent window started the VISA DLL, the VISA funtion will be closed if this window closed. Input NULL if parent window not specified.

Return value

Return TRUE if success; otherwise, return FALSE.

### **BOOL PASCAL viWrite( LPSTR szCmd )**

Send DSO VISA command to the DSO software.

Parameter

szCmd[In]:

Type: LPSTR

DSO VISA command, for more detail, please refer to [Acute DSO Visa Command List](#).

Return value

Return TRUE if success; otherwise, return FALSE.

### **int PASCAL viRead( LPSTR szRet, int iBufLen )**

Read DSO VISA command execute result from the DSO software.

Parameter

szRet[Out]:

Type: LPSTR

Data buffer for the VISA command returning value.

iBufLen[In]:

Type: **int**

Buffer size of szRet.

Return value

Return the buffer size used in this function. Return 0 when function failed.

## **BOOL PASCAL viCloseRM()**

Disable the VISA function and disconnect from the DSO software.

Return value

Return TRUE if success; otherwise, return FALSE.

## **int PASCAL viErrCode()**

Retrieve the DSO VISA error code.

Return value

Error Code	Description
AQVISA_ERROR_SUCCESS 0x0000	Execute without error.
AQVISA_ERROR_VISA_NO_RESPONSE 0x0001	DSO software is in demo mode or cannot respond the VISA command.
AQVISA_ERROR_CALLOPENRM 0x0002	Calling OpenRM again without CloseRM.
AQVISA_ERROR_OCCUPIED 0x0004	VISA function was occupied by other program.
AQVISA_NULL_HWND 0x0005	Null HWND parameter input in OpenRM function. (Not used after AqVISA.dll 1.0.0.3 version)
AQVISA_ERROR_DIR_NOTFOUND 0x0006	DSO installation directory was not found.
AQVISA_ERROR_VERSION_NOTMATCH 0x0007	DSO software version too old.
AQVISA_ERROR_NULL_DSO_WND 0x0008	DSO software has been closed.
AQVISA_ERROR_NO_DATA 0x0009	DSO software returns no data during command execute.
AQVISA_ERROR_BUFF_TOO_SMALL 0x000A	Buffer size too small for reading the command execute result.
AQVISA_STARTUP_ERROR 0x000B	AqVISA startup failed.
AQVISA_VISA_CLOSED 0x000C	AqVISA closed, or not startup properly.

# Acute DSO Visa Command List

## System Command

Command	Description																		
<a href="#">*IDN?</a>	Retrieve the model name and ID of the DSO. Ex: *IDN? return Model:TS2212A, Serial No.:TSA22120414																		
<a href="#">FORCETRIGGER</a>	Send Force Trigger command to DSO and refresh the capture memory.																		
<a href="#">LOCK</a>	Lock DSO memory to Retrieve current waveform. (Function not available when DSO memory > 1M)																		
<a href="#">*NUMCHS?</a>	Retrieve the numbers of available channel(s) on DSO.																		
<a href="#">*RST</a>	Restore factory setting of the DSO.																		
<a href="#">RUNSTOP</a>	Send Run/Stop command to the DSO.																		
<a href="#">*STB?</a>	Retrieve DSO status from returned 1 Byte data. Bit 7: DSO data ready flag. Bit 6-4: Reserved. Bit 3-0: DSO capture status <table border="1"> <thead> <tr> <th>Value</th><th>Capture status</th></tr> </thead> <tbody> <tr> <td>0</td><td>Ready</td></tr> <tr> <td>1</td><td>Triggered</td></tr> <tr> <td>2</td><td>Armed</td></tr> <tr> <td>3</td><td>Auto</td></tr> <tr> <td>4</td><td>Stop</td></tr> <tr> <td>5</td><td>Read Data</td></tr> <tr> <td>6</td><td>Roll</td></tr> <tr> <td>Other</td><td>Reserved</td></tr> </tbody> </table>	Value	Capture status	0	Ready	1	Triggered	2	Armed	3	Auto	4	Stop	5	Read Data	6	Roll	Other	Reserved
Value	Capture status																		
0	Ready																		
1	Triggered																		
2	Armed																		
3	Auto																		
4	Stop																		
5	Read Data																		
6	Roll																		
Other	Reserved																		
<a href="#">UNLOCK</a>	Unlock DSO memory.																		
<a href="#">EXPORT &lt;Type&gt;</a>	Export DSO data Printer: <Type> = PRINT Printer preview: <Type> = PREVIEW Clipboard: <Type> = CLIPBOARD Microsoft Office Word: <Type> = WORD Microsoft Office Excel: <Type> = EXCEL HTML file: <Type> = HTML Text file: <Type> = TXT Csv file: <Type> = CSV Tsv file: <Type> = TSV Matlab (.m) file: <Type> = MATLAB																		

	Wav file: <Type> = WAV DSOW file: <Type> = DSOW
<a href="#">RECALL &lt;Ref&gt; &lt;Dir&gt;</a>	Recall specified DSOW file to specified reference waveform. Ex: Recall D:\MyWave.dsow to Ref 2: <Ref> = Ref2 <Dir> = D:\MyWave.dsow
<a href="#">*SHUTDOWN</a>	Shutdown the DSO software

### ACQUIRE Command

Command	Description
<a href="#">ACQUIRE:MODE?</a>	Retrieve current acquisition mode.
<a href="#">ACQUIRE:MODE &lt;Mode&gt;</a>	Set DSO acquisition mode. 「 Sample mode 」 : <Mode> = Sample. 「 Average mode 」 : <Mode> = Average 「 High resolution mode 」 : <Mode> = HiRes 「 Peak detect mode 」 : <Mode> = PeakDetect 「 Envelope mode 」 : <Mode> = Envelop
<a href="#">ACQUIRE:RESOLUTION?</a>	Retrieve current hardware vertical resolution.
<a href="#">ACQUIRE:RESOLUTION &lt;Resolution&gt;</a>	Set hardware vertical resolution. 「 8Bits 」 :<Resolution> = 8 「 12Bits 」 :<Resolution> = 12 「 14Bits 」 :<Resolution> = 14 「 15Bits 」 :<Resolution> = 15 「 16Bits 」 :<Resolution> = 16

### Calibration Command

Command	Description
<a href="#">CALibration:ALL</a>	Activate DSO self-calibration function. (Must switch all the probes to REF, or remove all the probes before issuing this command)

### CH<x> Command

Command	Description
<a href="#">CH&lt;x&gt;:BWLimit?</a>	Retrieve the bandwidth limit setting of CH<x>.
<a href="#">CH&lt;x&gt;:BWLimit&lt;BWLimit&gt;</a>	Set bandwidth limit to CH<x> 20MHz: <BWLimit> = 20MHZ 100MHz: <BWLimit> = 100MHZ

	Full Bandwidth: <BWLimit> = FULL
<a href="#">CH&lt;x&gt;:COUPing?</a>	Retrieve the coupling setting of CH<x>.
<a href="#">CH&lt;x&gt;:COUPling &lt;Coupling&gt;</a>	Set AC/DC coupling to CH<x>. AC Coupling: <Coupling> = AC DC Coupling: <Coupling> = DC
<a href="#">CH&lt;x&gt;:OFFset?</a>	Retrieve the voltage offset of CH<x>.
<a href="#">CH&lt;x&gt;:OFFset &lt;Offset&gt;</a>	Set voltage offset of CH<x>, unit uV 100mV: <Offset> = 100000 / PROBE 1V: <Offset> = 1000000 / PROBE
<a href="#">CH&lt;x&gt;:ON &lt;On/Off&gt;</a>	Enable/Disable the CH<x> waveform display. Enable: <On/Off> = 1 Disable: <On/Off> = 0
<a href="#">CH&lt;x&gt;:PROBE?</a>	Retrieve the probe setting of CH<x>.
<a href="#">CH&lt;x&gt;:PROBE&lt;Probe&gt;</a>	Set the probe setting of CH<x>. x1: <Probe> = 1                      x10: <Probe> = 10 x100: <Probe> = 100              x1000: <Probe> = 1000 x2000: <Probe> = 2000
<a href="#">CH&lt;x&gt;:PROBEType?</a>	Retrieve the probe type of CH<x>.
<a href="#">CH&lt;x&gt;:PROBEType &lt;Type&gt;</a>	Set the probe type of CH<x>. Voltage probe: <Type> = VOLTAGE Current probe: <Type> = CURRENT
<a href="#">CH&lt;x&gt;:PROBECurrent?</a>	Retrieve the voltage/current convert ratio of CH<x>. Unit: uV/A
<a href="#">CH&lt;x&gt;:PROBECurrent &lt;Current&gt;</a>	Set the voltage/current convert ratio of CH<x>. Unit: uV/A 1mV/A: <Current> = 1000 20mV/A: <Current> = 20000
<a href="#">CH&lt;x&gt;:POSition?</a>	Retrieve the ground position of CH<x> Unit: div/screen
<a href="#">CH&lt;x&gt;:POSition &lt;Position&gt;</a>	Set the ground position of CH<x>, Unit: div/screen Bottom: <Position> = -4 Middle : <Position> = 0 Top: <Position> = 4
<a href="#">CH&lt;x&gt;:VOLTS?</a>	Retrieve the voltage division setting of CH<x>.
<a href="#">CH&lt;x&gt;:VOLTS &lt;Volts&gt;</a>	Set the voltage division setting of CH<x>. 100mV/Div: <Volts> = 100mV 1V/Div: <Volts> = 1V

## CURSOR Command

Command	Description
<a href="#">CURSor:FUNctioN &lt;Function&gt;</a>	Set Cursor function Horizontal cursor: <Function> = HORizontal Vertical cursor: <Function> = VERTical All cursor: <Function> = ALL OFF: <Function> = OFF
<a href="#">CURSor:FUNctioN?</a>	Get Cursor function
<a href="#">CURSor:HBArs:DELTA?</a>	Get the position delta of horizontal cursors
<a href="#">CURSor:HBArs:POSITioN&lt;x&gt; &lt;Pos&gt;</a>	Set horizontal cursor <x> position By voltage: <Pos> = 2.5V By current: <Pos> = 0.5A By IRE: <Pos> = 1 IRE
<a href="#">CURSor:HBArs:POSITioN&lt;x&gt;?</a>	Get horizontal cursor position
<a href="#">CURSor:HBArs:UNIts &lt;Unit&gt;</a>	Set horizontal cursor display unit Voltage/Current: <Unit> = VOLT IRE: <Unit> = IRE
<a href="#">CURSor:HBArs:UNIts?</a>	Get horizontal cursor display unit
<a href="#">CURSor:MODE &lt;Mode&gt;</a>	Set cursor tracking mode Tracking enable: <Mode> = TRACK Tracking disable: <Mode> = INDEpendent
<a href="#">CURSor:MODE?</a>	Get cursor tracking mode
<a href="#">CURSor:VBArS:DELTA?</a>	Get the position delta of vertical cursors
<a href="#">CURSor:VBArS:POSITioN&lt;x&gt; &lt;Pos&gt;</a>	Set vertical cursor <x> position 1ms before trigger point: <Pos> = -1MS 1KHz time after trigger point: <Pos> = 1KHZ 5900 <sup>th</sup> data point: <Pos> = 5900point
<a href="#">CURSor:VBArS:POSITioN&lt;x&gt;?</a>	Get vertical cursor <x> position
<a href="#">CURSor:VBArS:UNIts &lt;Unit&gt;</a>	Set vertical cursor display unit Time: <Unit> = TIME Frequency: <Unit> = FREQuency
<a href="#">CURSor:VBArS:UNIts?</a>	Get vertical cursor display unit

## DISplay Command

Command	Description
<a href="#">DISplay:PANELSIZE &lt;Size&gt;</a>	Set display panel size, from 0 to 5 Panel size 0: <Size> = 0 Panel size 5: <Size> = 5

<a href="#">DISplay:QUICKDRAW &lt;OnOff&gt;</a>	Set quick draw function status Enable: <OnOff> = ON Disable: <OnOff> = OFF
<a href="#">DISplay:QUICKDRAW?</a>	Get quick draw function status
<a href="#">DISplay:ZOOMWINDOW &lt;OnOff&gt;</a>	Set Zoom window status Show: <OnOff> = ON Hide: <OnOff> = OFF

## HORizontal Command

Command	Description
<a href="#">HORizontal:RECORDlength?</a>	Retrieve record length. Record length = 10k : return 10000 Record length = 64M : return 64000000
<a href="#">HORizontal:RECORDlength &lt;Length&gt;</a>	Set record length. Record length = 10k : <Length> = 10k Record length = 64M : <Length> = 64M
<a href="#">HORizontal:SCALE?</a>	Retrieve current time scale per div.
<a href="#">HORizontal:SCALE &lt;Time&gt;</a>	Set current time scale per div 1ms/Div : <Time> = 1ms 1S/Div : <Time> = 1S
<a href="#">HORizontal:TRIGGER:POSITION?</a>	Retrieve current trigger position in percentage.
<a href="#">HORizontal:TRIGGER:POSITION &lt;TrigPosition&gt;</a>	Set current trigger position in percentage. 10%: <TrigPosition> = 10% 50%: <TrigPosition> = 50%
<a href="#">HORizontal:POSITION?</a>	Get the center position of display, from 0 to 100%
<a href="#">HORizontal:POSITION &lt;Position&gt;</a>	Set the center position of display 10%: <Position> = 10% 10ms before trigger point: <Position> = 10ms 50us after trigger point: <Position> = 50us 500 <sup>th</sup> point: <Position> = 500point

## MEASure Command

Command	Description
<a href="#">MEASurement:ALL? &lt;Range&gt;</a>	Retrieve all the <MeasureType> results of specified channel with given <Range>. Gate by cursor: <Range> = CURSOR



	Display area: <Range> = SCREEN All waveform: <Range> = WAVEFORM
<a href="#">MEASurement:MEAS&lt;x&gt;:TYPE &lt;MeasureType&gt;</a>	Set the auto-measurement tag type for the information zone.
<a href="#">MEASurement:MEAS&lt;x&gt;:TYPE ?</a>	Get the auto-measurement tag type for the information zone.
<a href="#">MEASurement:MEASTAG&lt;x&gt;:TYPE &lt;MeasureType&gt;</a>	Set the auto-measurement tag type for the waveform display zone.
<a href="#">MEASurement:MEASTAG&lt;x&gt;:TYPE ?</a>	Get the auto-measurement tag type for the waveform display zone.
<a href="#">MEASurement:&lt;MeasureType&gt;</a>	Retrieve DSO measurement result.
<MeasureType>	Frequency: <MeasureType> = FREQuency? Period: <MeasureType> = DUTY? Max.: <MeasureType> = VMAX? Min.: <MeasureType> = VMIN? High: <MeasureType> = VHIGH? Low: <MeasureType> = VLOW? Vpp: <MeasureType> = VPP? Amplitude: <MeasureType> = VAMPLitude? Mean: <MeasureType> = VMEAN? Vrms: <MeasureType> = VRMS? +Duty: <MeasureType> = POSDuty? -Duty: <MeasureType> = NEGDuty? +Width: <MeasureType> = POSPulsewidth? -Width: <MeasureType> = NEGPulsewidth? +Overshoot: <MeasureType> = POSOvershoot? -Overshoot: <MeasureType> = NEGOvershoot? Rise Time: <MeasureType> = RISingtime? Fall Time: <MeasureType> = FALLingtime? Mid: <MeasureType> = VMIDDLE? Cyc. Vrms: <MeasureType> = CYCLEVRMS? Cyc. Mean: <MeasureType> = CYCLEVMEAN?







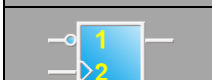

### Pass/Fail Command

Command	Description
---------	-------------

<a href="#"><u>PASSFAIL:SETMASK &lt;Ch&gt; &lt;Ref&gt; &lt;Volt&gt; &lt;Time&gt;</u></a>	Set Pass/Fail channel and tolerance range and start Pass/Fail preview. Ex: <Ch> = Ch1 <Ref> = Ref2 <Volt>/<Time>: Set tolerance voltage and time range.
<a href="#"><u>PASSFAIL:START</u></a>	Start Pass/Fail check function
<a href="#"><u>PASSFAIL:STOP</u></a>	Stop Pass/Fail check function

### TRIGger Command

Command	Description
<a href="#"><u>TRIGger:A:MODE?</u></a>	Retrieve current trigger mode.
<a href="#"><u>TRIGger:A:MODE &lt;Mode&gt;</u></a>	Set current trigger mode. Auto mode: <Mode> = Auto. Normal mode: <Mode> = Normal. Single mode: <Mode> = Single.
<a href="#"><u>TRIGger:A:SOURCE?</u></a>	Retrieve current trigger source channel.
<a href="#"><u>TRIGger:A:SOURCE &lt;Source&gt;</u></a>	Set trigger source channel. Trigger source from CH1 : <Source> = CH1. Trigger source from CH2 : <Source> = CH2. Trigger source from EXT: <Source> = Ext.
<a href="#"><u>TRIGger:A:METHod?</u></a>	Retrieve current trigger method.
<a href="#"><u>TRIGger:A:METHod EDGE &lt;Edge&gt;</u></a>	Set trigger method to Edge trigger. Rising edge : <Edge> = Rising Falling edge: <Edge> = Falling Alternate edge: <Edge> = Alternate Either edge: <Edge> = Either
<a href="#"><u>TRIGger:A:METHod WIDTH &lt;Width&gt;</u></a>	Set trigger method to Width trigger. +Width > T ms : <Width> = High G T ms -Width < T us : <Width> = Low L T us Any width = T S : <Width> = Either E T S +Width $\neq$ T ms : <Width> = High NE T ms
<a href="#"><u>TRIGger:A:METHod RUNT &lt;Runt&gt;</u></a>	Set trigger method to Runt trigger. +Runt: <Runt> = High -Runt: <Runt> = Low +Runt and +width > T ms : <Runt> = High G T ms

	-Runt and -width < T ms : <Runt> = Low L T ms
<a href="#">TRIGger:A:METhod VIDEO</a> <a href="#">&lt;Video&gt;</a>	Set trigger method to Video trigger. Any scanline: <Video> = Scanline N th scanline: <Video> = Scanline N Any field: <Video> = Field Odd field: <Video> = Odd Field Even field: <Video> = Even Field
<a href="#">TRIGger:A:METhod TIMEOUT</a> <a href="#">&lt;Timeout&gt;</a>	Set trigger method to Timeout trigger. Idle High > T ms : <Timeout> = High T ms Idle Low > T us : <Timeout> = Low T us Any Idle > T S : <Timeout> = Either T S
<a href="#">TRIGger:A:METhod PATTERN</a> <a href="#">&lt;Pattern&gt;</a>	Set trigger method to Pattern trigger.
	 <a href="#">&lt;Pattern&gt; = CH1 AND CH2</a>
	 <a href="#">&lt;Pattern&gt; = !CH1 OR CH2</a>
	 <a href="#">&lt;Pattern&gt; = CH1 NAND !CH2</a>
	 <a href="#">&lt;Pattern&gt; = !CH1 NOR CH2</a>
<a href="#">TRIGger:A:METhod STATE &lt;State&gt;</a>	Set trigger method to State trigger.
	 <a href="#">&lt;State&gt; = CH1 CH2</a>
	 <a href="#">&lt;State&gt; = CH2 CH1</a>
	 <a href="#">&lt;State&gt; = !CH1 CH2</a>
	 <a href="#">&lt;State&gt; = !CH1 !CH2</a>

## WAVEFORM Command

Command	Description
<a href="#">DATA:SOURCE:CH &lt;Ch&gt;</a>	Set data source from DSO CH<x>. CH<x> is available from CH1 to CH12.
<a href="#">WAVEform?</a>	Retrieve the raw data from selected channel. Voltage convert formula: $V = ((\text{Raw Data} - \text{YOFSA}) * \text{YMUL} + \text{YOFBSB}) * \text{PROBE} / 1000000) - \text{OFFSET}$

<a href="#"><u>WFMPre:CONVMethod?</u></a>	Retrieve the convert method of CH<x>. High Resolution: 1, SinX/X: 2
<a href="#"><u>WFMPre:OFFset?</u></a>	Retrieve the voltage offset of CH<x>, Unit: Volt
<a href="#"><u>WFMPre:ON?</u></a>	Retrieve the display state of CH<x>.
<a href="#"><u>WFMPre:POSition?</u></a>	Retrieve the ground position of CH<x>.
<a href="#"><u>WFMPre:PROBE?</u></a>	Retrieve the probe setting of CH<x>.
<a href="#"><u>WFMPre:VOLTS?</u></a>	Retrieve the voltage division of CH<x>.
<a href="#"><u>WFMPre:XINCR?</u></a>	Retrieve the sampling points interval of CH<x>.
<a href="#"><u>WFMPre:XOFS?</u></a>	Retrieve the voltage convert parameter of CH<x>.
<a href="#"><u>WFMPre:YMUL?</u></a>	Retrieve the voltage convert parameter of CH<x>.
<a href="#"><u>WFMPre:YOFSa?</u></a>	Retrieve the voltage convert parameter of CH<x>.
<a href="#"><u>WFMPre:YOFSB?</u></a>	Retrieve the voltage convert parameter of CH<x>.
<a href="#"><u>WFMREADY?</u></a>	Get waveform prepare status Return "0": Still preparing. Return "1": Ready for WAV? waveform read.

# AqVisa Command Description and Samples

## Symbol Description

< > Defined element  
= Is defined as  
| Exclusive OR  
{ } Group; one element is required  
[ ] Optional; can be omitted  
. . . Previous element(s) may be repeated  
// Comment  
<NR1> Signed integer value  
<NR2> Floating point value without an exponent  
<NR3> Floating point value with an exponent  
<bin> Digital data in binary format

The commands introduced below is described in capital as abbreviation form, users and either input full text or abbreviation command to control the DSO.

## System Command

### \*IDN?

Retrieve the model name and ID of the DSO.

Syntax: \*IDN?

Example: \*IDN? //Ex: Model:TS2212A, Serial No.:TSA22120414

### LOCK

Lock DSO memory to Retrieve current waveform.

(Function not available when DSO memory > 1M)

Syntax: LOCK

Example: LOCK

### \*NUMCHS?

Retrieve the numbers of available channel(s) on DSO.

Syntax: \*NUMCHS?

Example: \*NUMCHS? //Retrieved number is in Decimal (DEC) mode.

## \*RST

Restore factory setting of the DSO.

Syntax: \*RST

Example: \*RST

## RUNSTOP

Send Run/Stop command to the DSO.

Syntax: RUNSTOP

Example: RUNSTOP

## \*STB?

Retrieve DSO status from returned 1 Byte Decimal data.

Syntax: \*STB

Example: \*STB //DSO returns 128 when data ready.  
//DSO returns 0 when waiting for trigger signal.

//Bit 7: DSO Data Ready Flag

//Bit 6-4: Reserved

//Bit 3-0: DSO Capture Status

Value	Description
0	Ready
1	Triggered
2	Armed
3	Auto
4	Stop
5	Read Data
6	Roll
Other	Reserved

## EXPORT

Export DSO data

Syntax: EXPORT {PRINT|PREVIEW|WORD|EXCEL|HTML|

TXT|CSV|TSV|MATLAB|CLIPBOARD|DSOW}

Parameter: {PRINT....DSOW}

Command	Parameter
Hard Copy	PRINT
Preview	PREVIEW
Clipboard	CLIPBOARD
Word Format	WORD [DIR]
Excel Format	EXCEL [DIR]
HTML Format	HTML [DIR]
Text Format	TXT [DIR] [CHANNEL]
CSV Format	CSV [DIR] [CHANNEL]
TSV Format	TSV [DIR] [CHANNEL]
MATLAB (.m)	MATLAB [DIR] [CHANNEL]
WAV Format	WAV [DIR] [CHANNEL]
DSOW File	DSOW [DIR] [CHANNEL][RANGE]

- [DIR] Specify Export data directory, if this input is empty, the software will pop up a saving dialog for manual select.
- [CHANNEL] Export channel, input 0 for channel 1, input -1 for all channel.  
The software will automatically append channel text (CH1 or CH2) at the end of file name for distinguish the data source when exporting Text/CSV/TSV format file. It is also available to export all the waveform data into one file by add additional ONEFILE parameter in Text/CSV/TSV file format. However, this function is **not supported** in roll mode and logger mode.
- [RANGE] (Optional) Select data export range as SCREEN or CURSOR range, if this input is empty, the software will export the entire waveform data.

Example: EXPORT PRINT  
//Hard copy the software screen.  
EXPORT WORD D:\TestFile.doc  
//Export Word file format.  
EXPORT CSV D:\TestFile.csv 0  
//Export channel 1 waveform data in CSV format.  
EXPORT CSV D:\TestFile.csv ONEFILE  
//Export all channel data into single CSV format file.

EXPORT DSOW D:\MyDsow.dsow ch1

//Export channel 1 waveform data in DSOW format.

## UNLOCK

Unlock DSO memory.

Syntax: UNLOCK

Example: UNLOCK //Unlock DSO memory for next capture.

## ACQUIRE Command

### ACQUIRE:MODE

Access the DSO acquisition mode.

Syntax: ACQUIRE:MODE {SAMPLE|AVERAGE|HIRES|ENVELOPE|PEAKDETECT}  
ACQUIRE:MODE?

Parameter: SAMPLE Sample mode  
AVERAGE Average mode  
HIRES High Resolution mode  
ENVELOPE Envelope mode  
PEAKDETECT Peak Detect mode

Example: ACQUIRE:MODE HIRES  
//Set the DSO Acquisition mode to High Resolution

### ACQUIRE:RESOLUTION

Access the DSO acquisition hardware vertical resolution.

Syntax: ACQUIRE:RESOLUTION{8|12|14|15|16}  
ACQUIRE:RESOLUTION?

Parameter: 8 to 16 Bits

Example: ACQUIRE:RESOLUTION 8 //Set vertical resolution to 8Bits  
ACQUIRE:RESOLUTION 16 //Set vertical resolution to 16Bits

## CH<x> Command



## CH<x>:BWLimit

Access the bandwidth limit setting of CH<x>.

**Syntax:** CH<x>:BWLimit {FULL|20MHZ|100MHZ}  
CH<x>:BWLimit?

**Parameter:** {FULL|20MHZ|100MHZ} Bandwidth options

**Example:** CH1:BWL 100MHZ //Set Channel 1 Bandwidth limit to 100MHz.

## CH<x>:COUPling

Access the coupling setting of CH<x>.

**Syntax:** CH<x>:COUPling {AC|DC}  
CH<x>:COUPling?

**Parameter:** {AC|DC} Coupling options

**Example:** CH1:COUP AC //Set Channel 1 Coupling to AC coupling.

## CH<x>:OFFset

Access the voltage offset of CH<x>, unit: uV.

**Syntax:** CH<x>:OFFset <NR1>  
CH<x>:OFFset?

**Parameter:** <NR1> Input the voltage offset value, ex: 1000, 700, 50.

**Example:** CH1:OFF 1000 //Set Channel 1 voltage offset to 1 mV (Probe x1)  
CH1:OFF 100 //Set Channel 1 voltage offset to 1 mV (Probe x10)

## CH<x>:ON

Access the display setting of CH<x>.

**Syntax:** CH<x>:ON {1|0}  
CH<x>:ON?

**Parameter:** 1: On; 0: Off

**Example:** CH1:ON 0 //Channel 1 display off

## CH<x>:POSition

Access the ground position setting of CH<x>

**Syntax:** CH<x>:POSition <NR3>  
CH<x>:POSition?

Parameter: <NR3> Input the ground position value, ex: 1.5, -2, 3.

Example: CH1:POS -1.5 //Set the ground position of channel 1 to -1.5

## CH<x>:PROBE

Access the probe setting of CH<x>.

Syntax: CH<x>:PROBE {1|10|100|1000|2000}  
CH<x>:PROBE?

Parameter: {1|10|100|1000|2000} Probe setting options

Example: CH1:PROBE 100 //Set the probe setting of channel 1 to x100

## CH<x>:PROBEType

Access the probe type setting of CH<x>.

Syntax: CH<x>:PROBEType {VOLTAGE|CURRENT}  
CH<x>:PROBEType?

Parameter: VOLTAGE Probe type = Voltage probe

CURRENT Probe type = Current probe

Example: CH1:PROBET CURRENT  
//Set the probe type of channel 1 to current probe

## CH<x>:PROBECurrent

Access the voltage/current convert ratio setting of CH<x>. Unit: uV/A

Syntax: CH<x>:PROBECurrent <NR1>  
CH<x>:PROBECurrent?

Parameter: <NR1> Input voltage/current convert ratio, unit uV/A.

Example: CH1:PROBEC 100000  
//Set the probe convert ratio of channel 1 to 100mV/A.

## CH<x>:VOLTS

Access the voltage division setting of CH<x>.

Syntax: CH<x>:VOLTS <NR1>  
CH<x>:VOLTS?

Parameter: <NR1> Input voltage division value, ex: 2V, 1V, 500mV.

Example: CH1:VOLTS 100mV  
//Set the voltage division of channel 1 to 100mV/div.

## HORizontal Command

### HORizontal:RECORDlength

Access DSO record length setting.

**Syntax:**       HORizontal:RECORDlength <NR1>  
                  HORizontal:RECORDlength?

**Parameter:**   <NR1>   Input the record length value.

**Example:**       HOR:REC 100k       //Set the record length to 100k points  
                  HOR:REC?         //Return value = "100000" (100k)

List of available record length for each DSO model:

Model	Record Length
TravelScope 2212 Series	10k, 50k, 100k, 500k, 1M, 5M, 10M, 32M, 64M
DS1002 / DS1102 Series	500, 1k, 2k, 4k, 8k, 16k, 32k, 64k
DS1202 Series	500, 1k, 2k, 4k, 8k, 16k, 32k, 512k
DS1302 Series	500, 1k, 2k, 4k, 8k, 16k, 32k, 2048k

### HORizontal:SCAle

Access current time scale setting. //Time/Div

**Syntax:**       HORizontal:SCAle <NR1>  
                  HORizontal:SCAle?

**Parameter:**   <NR1>   Input the time scale value, ex: 100ns, 40us

**Example:**       HOR:SCA 40us       //Set time scale to 40 us

### HORizontal:TRIGGER:POSITION

Access current trigger position setting.

**Syntax:**       HORizontal:TRIGGER:POSITION <NR1>  
                  HORizontal:TRIGGER:POSITION?

**Parameter:**   <NR1>   Input the trigger position value, ex: 10%, 25%, 70%

**Example:**       HOR:TRIGGER:POSITION 35%   //Set trigger position to 35%

# MEASure Command

## MEASure:<MeasureType>?

Retrieve DSO measurement result.

**Syntax:** MEASure:<MeasureType>?

**Parameter:** <MeasureType> Input the type name to Retrieve to measure result.

Measure type	<MeasureType>	Unit
Frequency	FREQuency?	Hz, kHz, MHz
Period	DUTY?	ns, us, ms, s
Max.	VMAX?	uV, mV, V
Min.	VMIN?	uV, mV, V
High	VHIGH?	uV, mV, V
Low	VLOW?	uV, mV, V
Vpp	VPP?	uV, mV, V
Amplitude	VAMPlitude?	uV, mV, V
Mean	VMEAN?	uV, mV, V
Vrms	VRMS?	uV, mV, V
+Duty	POSDuty?	%
-Duty	NEGDuty?	%
+Width	POSPulsewidth?	ns, us, ms, s
-Width	NEGPulsewidth?	ns, us, ms, s
+Overshoot	POSOvershoot?	%
-Overshoot	NEGOvershoot?	%
Rise Time	RISingtime?	ns, us, ms, s
Fall Time	FALLingtime?	ns, us, ms, s
Mid	VMIDdle?	uV, mV, V
Cyc. Vrms	CYCLEVRMS?	uV, mV, V
Cyc. Mean	CYCLEVMEAN?	uV, mV, V

**Example:**

```

DATA:SOURCE:CH1 //Set data source from CH1
MEAS:FREQ? //Get CH1 frequency, ex: 100kHz, 1MHz
MEAS:POSP? //Get CH1 +Duty, ex: 55.1%
MEAS:VMEAN? //Get CH1 Mean, ex: 35.74mV

```

# TRIGger Command

## TRIGger:A:METhod?

Retrieve current trigger method.

**Syntax:** TRIGger:A:METhod?

**Example:** TRIGger:A:METhod? //Returned value example: "Edge Falling"

## TRIGger:A:METhod EDGE

Set trigger method to Edge trigger.

**Syntax:** TRIGger:A:METhod EDGE {RISING|FALLING|ALTERNATE|EITHER}

**Parameter:**

RISING	Rising Edge
FALLING	Falling Edge
ALTERNATE	Alternate edge
EITHER	Either Edge

**Example:** TRIG:A:MET EDGE ALTERNATE  
//Set trigger method to alternate edge trigger

## TRIGger:A:METhod PATTERN

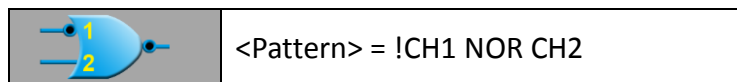
Set trigger method to Pattern trigger.

**Syntax:** TRIGger:A:METhod PATTERN {CH1|!CH1}{AND|OR|NAND|NOR}{CH2|!CH2}

**Parameter:**

CH1/!CH1	Channel1/Channel 1 Inverse
AND/OR/NAND/NOR	Logic gate
CH2/!CH2	Channel2/Channel 2 Inverse

**Example:** TRIG:A:MET PATTERN !CH1 NOR CH2



//Triggered when condition goes true.

## TRIGger:A:METhod RUNT

Set trigger method to Runt trigger.

**Syntax:** TRIGger:A:METhod RUNT {HIGH|LOW|EITHER} [L|G|E|NE] <NR1>

**Parameter:**

HIGH/LOW/EITHER	+Runt/-Runt/Any Runt	
L/G/E/NE	Less/Greater/Equal/Not Equal	//Optional
<NR1>	Input the specified width time, ex: 50us	//Optional

Example: TRIG:A:MET RUNT LOW L 150 us  
//Triggered on -Runt and -Width < 150us

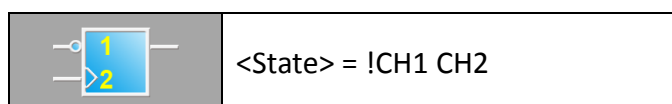
## TRIGger:A:METHod STATE

Set trigger method to State trigger.

Syntax: TRIGger:A:METHod STATE {CH1|!CH1|CH2|!CH2}  
{CH1|!CH1|CH2|!CH2}

Parameter: CH1|!CH1|CH2|!CH2 Channel Input /Channel Input Inverse  
The second input is treated as Clock.  
The two inputs must come from different channel.

Example: TRIG:A:MET STATE !CH1 CH2



//Triggered when State value goes true.

## TRIGger:A:METHod TIMEOUT

Set trigger method to Timeout trigger

Syntax: TRIGger:A:METHod TIMEOUT {HIGH|LOW|EITHER} <NR1>

Parameter: HIGH/LOW/EITHER +Width/-Width/Any Width  
<NR1> Input the specified timeout trigger time, ex: 50ms

Example: TRIG:A:MET TIMEOUT HIGH 50 us  
//Trigger when Idle High > 50us

## TRIGger:A:METHod VIDEO

Set trigger method to Video trigger.

Syntax: TRIGger:A:METHod VIDEO {SCANLINE|SCANLINE <NR1>|FIELD|ODD  
FIELD|EVEN FIELD}

Parameter: SCANLINE Trigger on any Scanline.  
SCANLINE <NR1> Trigger on specified Scanline.  
FIELD Trigger on any Field  
ODD FIELD/EVEN FIELD Trigger on Odd Field/Even Field

Example: TRIG:A:MET VIDEO SCANLINE 75 //Triggered on 75<sup>th</sup> Scanline

## TRIGger:A:METHod WIDTH

Set trigger method to Width trigger.

**Syntax:** TRIGger:A:METHOD WIDTH {HIGH|LOW|EITHER} {L|G|E|NE} <NR1>

**Parameter:** HIGH/LOW/EITHER +Width/-Width/Any Width  
L/G/E/NE Less/Greater/Equal/Not Equal  
<NR1> Input the specified width time, ex: 50us

**Example:** TRIG:A:Method Width LOW G 150 ms  
//Triggered when -Width > 150ms

## TRIGger:A:MODE

Access trigger mode setting.

**Syntax:** TRIGger:A:MODE {AUTO|NORMAL|SINGLE}  
TRIGger:A:MODE?

**Parameter:** AUTO Auto mode  
NORMAL Normal mode  
SINGLE Single mode

**Example:** TRIG:A:MODE SINGLE //Set trigger mode to single mode

## TRIGger:A:SOURCE

Access trigger source setting.

**Syntax:** TRIGger:A:SOURCE {CH1|CH2|EXT}  
TRIGger:A:SOURCE?

**Parameter:** CH1 Set trigger source to CH1  
CH2 Set trigger source to CH2  
EXT Set trigger source to EXT

**Example:** TRIG:A:SOURCE EXT //Set trigger source to EXT

# WAVeform Command

## DATA:SOURCE:CH <x>

Set data source from DSO CH<x>.

**Syntax:** DATA:SOURCE:CH<x>

**Example:** DATA:SOURCE:CH1 //Set data source from DSO CH1

## WAVeform?

Retrieve the raw data from selected channel.

Syntax: WAVeform?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WAV? //Retrieve the row data from CH1  
//Voltage Convert Formula  
//V = ((RowData – YOFS A) \* YMUL + YOFS B) \* PROBE / 1000000) – OFFSET

## WFMPre:CONVMMethod?

Retrieve the convert method of CH<x>.

Syntax: WFMPre:CONVMMethod?

Example: WFMP:CONVM? //Return 0 for Normal mode  
//Return 1 for High Resolution mode  
//Return 2 for SinX/X mode

## WFMPre:Offset?

Retrieve the voltage offset of CH<x>, Unit: Volt

Syntax: WFMPre:Offset?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:Offset?  
//Retrieve Voltage Offset value (Decimal) from CH1, unit: Volt (V)

## WFMPre:ON?

Retrieve the display state of CH<x>.

Syntax: WFMPre:ON?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:ON? //Return 0 for CH1 display OFF.  
//Return 1 for CH1 display ON.

## WFMPre:POSition?

Retrieve the ground position of CH<x>.

Syntax: WFMPre:POSition?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1



WFMP:POS?

//Retrieve the ground position of CH1, ex: 1.5 (Unit: div/screen)

## WFMPre:Probe?

Retrieve the probe setting of CH<x>.

Syntax: WFMPre:PROBE?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:PROBE? //Retrieve the probe setting of CH1  
//Return 1 means probe x1  
//Return 10 means probe x10.

## WFMPre:VOLTS?

Retrieve the voltage division of CH<x>.

Syntax: WFMPre:VOLTS?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:VOLTS? //Retrieve the voltage division of CH1

## WFMPre:XINCR?

Retrieve the sampling points interval of CH<x>.

Syntax: WFMPre:XINCR?

Example: WFMP:XINCR?  
//Retrieve the sampling points interval of CH1, ex: 1us, 10us

## WFMPre:XOFS?

Retrieve the voltage convert parameter of CH<x>.

Syntax: WFMPre:XOFS?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:XOFS? //Return a float point value, ex: 0.13579

## WFMPre:YMUL?

Retrieve the voltage convert parameter of CH<x>.

Syntax: WFMPre:YMUL?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1

WFMP:YMUL?

//Return a float point value, ex: 0.13579

## WFMPre:YOFSA?

Retrieve the voltage convert parameter of CH<x>.

Syntax: WFMPre:YOFSA?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:YOFSA? //Return a float point value, ex: 0.13579

## WFMPre:YOFSB?

Retrieve the voltage convert parameter of CH<x>.

Syntax: WFMPre:YOFSB?

Example: DATA:SOURCE:CH1 //Set data source from DSO CH1  
WFMP:YOFSB? //Return a float point value, ex: 0.13579



**Acute Technology Inc.**

**[www.acute.com.tw](http://www.acute.com.tw)**

**Address: 6F-7 #12, Ln. 609, ChongXin Rd. Sec. 5, SanChong  
Dist., New Taipei city 24159, Taiwan**

**Tel: +886-2-2999-3275**

**Fax: +886-2-2999-3276**

**E-mail: [service@acute.com.tw](mailto:service@acute.com.tw)**