

Visacmd

From TestWiki

a command-line utility that will read/write instrument commands to/from stdout and then exit. the idea is to let command-line scripting (batch files) handle things like event processing, looping, waits, and file IO.

Contents

- 1 exit codes
- 2 arguments
 - 2.1 /a=
 - 2.2 /c=
 - 2.3 /l
 - 2.4 /b=
 - 2.5 /t=
 - 2.6 /o=
 - 2.7 /f
 - 2.8 /w
- 3 examples
 - 3.1 list of instruments
 - 3.2 query an instrument
 - 3.3 send a command
 - 3.4 save stdout to a file
 - 3.5 send command from file
 - 3.6 log 50 measurements every 5 seconds
 - 3.7 save a screen capture
 - 3.8 save sample data
 - 3.9 send a waveform
 - 3.10 save event table
- 4 other neat tricks
- 5 what visacmd cannot do (and never will)

exit codes

visacmd returns 0 when it completes without an error

visacmd returns -1 when passed an invalid argument

visacmd returns -2 when there is a visa error. the visa status value and description are printed to stderr

arguments

/a=

which visa address to send/receive commands

/c=

the command/query string. if file exists, will read contents of file. if a query is detected, a read is performed and output to stdout.

/I

suppress formatted output. allows for dumping binary data to file.

/b=

overrides default buffer (20KB). while multi-reads/writes will be handled by the software, increasing the buffer may improve performance while decreasing the buffer may improve latency, reliability

/t=

overrides default timeout (2s). during long operations, this may be required.

/o=

outputs to file specified without formatting

/f

forces a viRead. command argument becomes optional

/w

disable auto-query

examples

visacmd.exe does not install and does not require any additional files (aside from a VISA). visacmd.exe simplicity allows Windows command-line scripting to handle the heavy lifting.

Note: Windows XP does not have a delay command out of the box. Windows Server 2003 Resource Kit Tools (<http://www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cff&DisplayLang=en>) contains the **SLEEP** command which will allow you to add delays.

My examples will not go into detail about the Windows command-line. For more information refer to http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds_shelloverview.mspx?mfr=true

list of instruments

I: is my usb drive, and visacmd.exe is the only file within myfolder. Simply execute visacmd without any

arguments to get the version, available arguments, and a list of instruments.

```
I:\myfolder>visacmd
VISACMD.EXE - A simple instrument communication utility.
Version: Apr 20 2010 - 13:21:40
Options:
  /a= - instrument address (required)
  /c= - string or file containing instrument command (required, unless /r)
  /t= - specify command timeout (optional)
  /b= - specify buffer (optional)
  /o= - specify output file (without formatting) (optional)
  /l - without echo or formating (optional)
  /w - disable auto query (optional)
  /r - force read from instrument (optional)

VISA Resource Manager: Available Instruments:
GPIB0::15::INSTR
TCPIP::134.62.36.55::INSTR

*** Remember to update your VISA Resource Manager ***
```

query an instrument

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=*idn?
USB::0x0699::0x0401::C000331::INSTR <- *idn?
USB::0x0699::0x0401::C000331::INSTR -> TEKTRONIX,MSO4104,C000331,CF:91.1CT FV:v2
| 48 DPO4EMBD:V1.00
```

send a command

when you have a space in the command, you have to put double-quotes around the command. this is a Windows command-line requirement.

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="ch1:ter 50;sca 20
0e-3;:hor:sca 400e-9"
USB::0x0699::0x0401::C000331::INSTR <- ch1:ter 50;sca 200e-3;:hor:sca 400e-9
```

if you do not receive the results you expect, check for errors

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="ch1:ter 50;sca 20
0e-3;:hor:sca 400ef-9"
USB::0x0699::0x0401::C000331::INSTR <- ch1:ter 50;sca 200e-3;:hor:sca 400ef-9
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=*esr?;allev?
USB::0x0699::0x0401::C000331::INSTR <- *esr?;allev?
USB::0x0699::0x0401::C000331::INSTR -> 32;121,"Invalid character in numeric; Une
xpected input following numeric data; ch1:ter 50;sca 200e-3;:hor:sca 400e"
```

save stdout to a file

stdout (standard out) is what you see in the console. most communication is ASCII so using stdout is convenient to save to a file.

the ***lrn?** command will create a huge concatenated command of the target instrument's state. this includes things like **data** and **wfmpr** which are important for remote communication. the 'scope setup files do not. visacmd does echo the address and command and formats the reply to the console. we must turn this off with

the /I for a naked reply or else the next example will not succeed.

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=*lrn? > mso4104.lrn
n /1
```

if you turn on headers, you can do this same thing for a small subset of commands

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="head 1;verb 0;data
a?"
USB::0x0699::0x0401::C000331::INSTR <- head 1;verb 0;data?
USB::0x0699::0x0401::C000331::INSTR -> :DAT:DEST REF1;ENC RIB;SOU CH1;STAR 1;STO
P 1000000;WID 1
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=data? /1 > mso4104
\data.txt
```

send command from file

if the command is a file, the file is opened and the contents of the file are sent to the instrument. this works great with the file created in the above example. since this is a very large command, i suppress echo and formatting with /I.

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=mso4104.lrn /I
```

and just the **data** attributes

```
I:\myfolder>visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c=mso4104_data.txt /I
```

log 50 measurements every 5 seconds

min, max, and mean amplitude measurements with single shot acquisitions. since this is a bit more complex, you'll want to put this into a batch file which is simply a text file with the ***.bat** extension. you can execute batch files from the command-line the same way you execute visacmd.exe.

test.bat

```
:: measurement script
:: takes an acquisition about every five seconds and records the min, max, and
:: mean measurements to a text file

@echo off
echo running test
visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="*cls,:acq:stopa seq,:measu:imm:soul ch1;" /I
echo %DATE% %TIME% > mymeasurements.txt
echo mean:max:min >> mymeasurements.txt
FOR /L %%i IN (1,1,50) DO (
sleep 5
echo acquisition %%i/50
visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="acq:state 1" /I
visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="*opc?" /t=100000 /I > NUL
visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="measu:imm:typ mean;val?;typ max;val?;typ mini;val?" /I >>
)
visacmd /a=USB::0x0699::0x0401::C000331::INSTR /c="*esr?;allev?" /I >> mymeasurements.txt
echo done test
```

save a screen capture

stdout was never meant for binary data. by specifying a file with /o=, visacmd will place the reply from the instrument into that file unaltered. also note the string "hardc start" doesn't contain a question mark, so /r is required to force a read to capture the data.

```
visacmd /a=tcpip::134.62.36.55::instr /c="sav:imag:filef png;inks off"  
visacmd /a=tcpip::134.62.36.55::instr /c="hardc start" /r /o=my_scope.png
```

save sample data

as per page 2-64 of MSO4000 programmer manual (077-0248-00)

Depending on your oscilloscope, ASCII encoding may return scaled voltage data or digitizing levels. Because the MSO4000 series returns digitizing levels, this output will require post-processing to convert it into voltages.

Alternatively, you could set the encoding to a binary format and save the curve query to a file (using **/o=**) and process that file with a third-party program.

Alternatively alternatively, you could save the waveform file to local storage and then read the file back (as in the event table demonstration).

send a waveform

8-bit 12 sample sine wave to AWG7000 series

binary contents of file **mywfm**

```
-----[Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F]-----
00000000 3A 77 6C 69 73 3A 77 61 76 3A 6E 65 77 20 22 74 :wlis:wav:new "t
00000010 65 73 74 22 2C 31 32 2C 69 6E 74 3B 64 61 74 61 est",12,int;data
00000020 20 22 74 65 73 74 22 2C 23 32 32 34 C0 1F C0 2F "test",#224À.À/
00000030 40 3B 80 3F 40 3B C0 2F C0 1F 00 10 40 04 00 00 @;€?@;À/À...@...
00000040 40 04 00 10 @...
```

```
visacmd /a=gpib8::1 /c=*rst,*cls /
visacmd /a=gpib8::1 /c=mywfm /w /
visacmd /a=gpib8::1 /c=*esr?;syst:err?
```

save event table

readet.bat

```
:: to read the event table remotely requires the use of local storage
:: this could be either USB or compact flash. each physical port has it's own letter.
@echo off

set MYSCOPE=tcpip::134.62.36.55::instr
set TEMP_FILE=e:/my_event_table.csv

visacmd /a=%MYSCOPE% /c="sav:event:bus1 \"%TEMP_FILE%\""
visacmd /a=%MYSCOPE% /c="files:readf \"%TEMP_FILE%\" /r /l"
```

output from readet.bat

```
I:\myfolder>readet.bat
tcpip::134.62.36.55::instr <- sav:event:bus1 "e:/my_event_table.csv"
"Tektronix MSO4104, version v2.48, serial number C000331"
"Bus Definition: PARALLEL"
 Time, Data
7.700000e-07, 0
1.570000e-06, 1
2.370000e-06, 2
3.170000e-06, 3
3.970000e-06, 4
4.770000e-06, 5
5.570000e-06, 6
6.370000e-06, 7
7.170000e-06, 8
7.970000e-06, 9
8.770000e-06, A
9.570000e-06, B
```

other neat tricks

- Execute the OpenChoice Call Monitor before using **visacmd** and watch the VISA traffic!

what visacmd cannot do (and never will)

visacmd will never process any data, that is visacmd will never be able to "understand" the data to or from the instrument and act upon it.

Retrieved from "<http://localhost/mediawiki/index.php/Visacmd>"

Category: Aeware

- This page was last modified on 6 May 2010, at 21:27.