Reference

Tektronix

**VX4101A
Multipaq™ Instrument**

071-0050-00

This document applies to firmware version 2.00
and above.

Refer to sections in the *VX4101A Multipaq™ Instrument User Manual* for more information.

Setup

Be sure all switches are correctly set (see the *Controls and Indicators* section). Follow Installation guidelines (see the *Installation* section).

The default condition of the VX4101A Module after the completion of the power-on self test is listed in full under the *RST command in the *IEEE 488.2 Common Commands* section.

LEDs

The VX4101A has four LEDs visible on its front panel. These LEDs are labeled as follows:

Power	This LED is On if all six fuses for the six power buses are intact. Any single fuse being blown results in the LED turning OFF.
Fail LED	This LED is normally OFF. During power-on or reset self-test, the LED will be ON for the duration of the test. If the VX4101A detects a failure during normal operation, the LED will come ON and the SYSFAIL line on the backplane will be true.
Message	This LED flickers ON when the VX4101A is being addressed on the VME backplane by its commander
ERR	This LED is normally OFF. However, it may blink on and off to indicate error conditions. The most common reason is a command syntax error has been detected. Other error conditions that will cause the LED to blink are discussed elsewhere in this manual. Sending the "SYSTem:ERRor?" query to the instrument will return the cause of the error. When all errors in the queue have been retrieved, the error LED will return to the OFF state

Command Syntax

Command protocol and syntax for the VX4101A Module are described in full in the *Command Syntax* section.

- A command is a string of ASCII characters (255 max.) terminated by a <program message terminator>.

- Commands can be strung together by delimiting the individual commands with a semi-colon (;).

- All characters in a command may be sent in either upper or lower case form.

- Multiple data parameters are separated by a comma (,).

- White space characters exclude the <LF>. White space is allowed anywhere in a command string, except: within a program mnemonic; around a (:) separator; between a mnemonic and a (?); following an (*); within a number; within a list.

- A question mark (?) following a command indicates a response is to be returned. All responses from the module are terminated with the <LF>.

Numeric Value Formats: Fixed or floating-point formats are allowed (with no embedded spaces). All numbers are rounded to the nearest value appropriate for the particular command. If a sign is not specified, the number is assumed to be positive.

VX4101A Multipaq ™ Instrument Commands

These SCPI commands control overall functions of theVX4101A.

ABORt Subsystem

Commands	ABORt[:COMPlete]
	ABORt[:IMMediate]

CALibrate Subsystem

Commands	CALibrate:ROSCillator
	:CLEar
	:MANual <frequency>

INSTrument Subsystem

Commands	INSTrument:NSELect <instrument id>
	INSTrument[:SELect] <instrument name>
Queries	INSTrument:CATalog?
	:FULL?
	:LONG?
	INSTrument:COUNt?
	INSTrument:NSELect?
	INSTrument[:SELect]?

OUTput Subsystem

Commands	OUTPut:EXTernal:SOURce <ext source>
	OUTPut:TTLTrg[<N>]:SOURce <trig source>
Queries	OUTPut:EXTernal:SOURce:CATalog?
	OUTPut:EXTernal:SOURce?
	OUTPut:TTLTrg[<N>]:SOURce?
	OUTPut:TTLTrg[<N>]:SOURce:CATalog?

SOURce Subsystem

Commands	SOURce:ROSCillator[:SOURce] <source>
	SOURce:ROSCillator:VALue <frequency>
Queries	SOURce:ROSCillator[:SOURce]?
	SOURce:ROSCillator:VALue?

SYStem Subsystem

Commands	SYSTem:LANGuage <language>
	SYSTem:RQUeue
	:QMODe <qmode>
	:SNUMber[:SET] <snumber>
	:SNUMber:STATe <state>
	SYSTem:TIMeout <timeout in seconds>
Queries	SYSTem:ERRor?
	SYSTem:LANGuage?
	SYSTem:OPTions?:DESCription <option code>
	SYSTem:READy?
	SYSTem:RQUeue?
	:QMODe
	:SNUMber[:SET]?
	:SNUMber:STATe?
	SYSTem:SNUMber?
	SYSTem:TIMeout?
	SYSTem:VERSion?

TRIGger Subsystem

Commands	TRIGger([:SEQuence1]|:STARt)[:LAYer]:FIRe <command trigger>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:TIMer <timer seconds>
Queries	TRIGger([:SEQuence1]|:STARt)[:LAYer]:TIMer?

Universal Counter Commands

ARM Subsystem

Commands   ARM([:SEQuence1]|:STARt)[:LAYer]

        :DELay <time delay>
        :ECOunt <event delay>
        :IMMediate
        :LEVel <threshold>
        :MODE <mode>
        :SLOPe <slope>
        :SOURce <source>

        ARM(:SEQuence2|:STOP)[:LAYer]

        :DELay <time delay>
        :ECOunt <event delay>
        :IMMediate
        :SOURce <source>

Queries   ARM([:SEQuence1]|:STARt)[:LAYer]

        :DELay?
        :ECOunt?
        :LEVel?
        :MODE?
        :SLOPe?
        :SOURce?

           :CATalog[:ALL]?

            :DELayable?
            :FIXed?

        ARM(:SEQuence2|:STOP)[:LAYer]

        :DELay?
        :ECOunt?
        :SOURce?

          :CATalog[:ALL]?

            :DELayable?
            :FIXed?

CALCulate Subsystem

Commands	CALCulate:LIMit

        :ENVelope[:DATA] <threshold1>,<threshold2>
        :LOWer[:DATA] <threshold>
        :UPPer[:DATA] <threshold>

    CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
    CALCulate:TRANsform:HISTogram:POINts <numeric_value>
    CALCulate:TRANsform:HISTogram:RANGe <min>,<max>
    CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>

Queries   CALCulate:AVERage?[<count>[,<offset>[,<step_size>]]]
    CALCulate:LIMit

        :ENVelope
        :FCOunt?
        :LOWer [DATA]?
        :REPort [:DATA]?
        :UPPER [DATA]

    CALCulate:MEDian?[<count>[,<offset>[,<step_size>]]]
    CALCulate:MINimum?[<count>[,<offset>[,<step_size>]]]
    CALCulate:MAXimum?[<count>[,<offset>[,<step_size>]]]
    CALCulate:SDEViation?[<count>[,<offset>[,<step_size>]]]
    CALCulate:TRANsform:HISTogram?

        :ABOVe?
        :BELow?
        :COUNt?
        :POINts?
        :RANGe?
        :RANGe:AUTO?

    CALCulate:VARiance?[<count>[,<offset>[,<step_size>]]]

CALibrate Subsystem

Commands	CALibrate:ARM[:VALue] <arm input voltage>
	CALibrate[<channel>]
	:DELay <cross channel specifications>
	:HYSTeresis
	:LFCOmp
	:LINearity
	:VALue <input voltage>
	:ZERO
	CALibrate:DTI
	CALibrate3:BIAS
	CALibrate[1|2|3]:ROSCillator <input frequency>

CONFigure Subsystem

Commands	CONFigure[1|2|3]([:SCALar]|:ARRay)

       :FREQuency[<array size>[,<expected value>[,resolution>]]]

       :FREQuency:RATio[<array size>[,<expected value>[,resolution>]]]

       :PERiod[<array size>[,<expected value>[,resolution>]]]

    CONFigure[1\|2]([:SCALar]\|:ARRay)

       :NDUTy cycle\|PDUTCycle\|DCYCle [<array size>[,<dcycle reference>[,<expected value>[,resolution>]]]

       :NWIDth\|PWIDth [<array size>[,<pwidth reference>[,<expected value>[,resolution>]]]

       :PHASe [<array size>[,<expected value>[,resolution>]]]

       :RTIMe\|FTIMe\|RISE:TIME\|FALL:TIME [<array size>[,<low reference>[,<high reference> [,<expected value>[,resolution>]]]]]

       :TINTerval [<array size>[,<expected value>[,resolution>]]]

       [:VOLTage]

           :AC [<array size>[,<expected value>[,resolution>]]]
           :DC [<array size>[,<expected value>[,resolution>]]]
           :MINimum [<array size>[,<expected value>[,resolution>]]]
           :MAXimum [<array size>[,<expected value>[,resolution>]]]
           :PTPeak [<array size>[,<expected value>[,resolution>]]]

    CONFigure[1\|2\|11\|12\|21\|22]([:SCALar]\|:ARRay)

       :TINTerval:DELay:TIME\|EVENts[<array size>[,<delay time>\|<delay events>[,<expected value>[,resolution>]]]]]

       CONFigure[1\|2\|10\|20]([:SCALar]\|:ARRay)

       :TOTalize

    NOTE: If you specify SCALar in the command, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.

Queries	CONFigure?

FETCh? Subsystem

Queries	FETCh[:...]?
	:DCYCle?
	:FALL:TIME?
	:FREQuency?
	:FREQuency:RATio?
	:FTIMe?[<count>
	:NDUTycycle?
	:PDUTycycle?
	:NWIDth?
	:PWIDth?
	:PHASe?
	:PERiod?
	:RISE:TIME?
	:RTIMe?
	:TINTerval?
	:TINTerval:DELay?
	FETCh:COUNt?
	FETCh:TOTalize?
	FETCh[:VOLTage][:...]?
	:AC?
	:DC?
	:MINimum?
	:MAXimum?
	:PTPeak?

INITiate Subsystem

Commands	INITiate[:IMMediate]
	INITiate:CONTinuous [ON]|OFF|0|1
Queries	INITiate:CONTinuous?

INPut Subsystem

Commands	INPut[1|2]

INPut[1|2]

    :ATTenuation <attenuation>|DEFault|MINimum|MAXimum
    :COMParator[1|2]:LEVel

        [:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
        :RELative <relative level>

    :HYSTeresis

        [:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
        :RELative <relative level>

    :SLOPe<slope>

INPut[1|2]:COUPling <coupling>
INPut[1|2]:FILTer[:LPASs]

    :FREQuency<frequency>|DEFault|MINimum|MAXimum
    [:STATe]<filter state>

    :GAIN<gain>|DEFault|MINimum|MAXimum
    :IMPedance<impedance>|DEFault
    :OFFSet

     [:ABSolute]<absolute offset>|DEFault|MINimum|MAXimum
     :RELative<relative offset>

    :SETup<expected ptp>[,<expected offset>]

INPut:SETup:AUTO <auto>

Queries	INPut[1|2]

INPut[1|2]

    :ATTenuation?[DEFault|MINimum|MAXimum]
    :COMParator[1|2]:LEVel

        :LEVEL
         [:ABSolute]?[DEFault|MINimum|MAXimum]
          :RELative?

    :HYSTeresis

        [:ABSolute]?[DEFault|MINimum|MAXimum]
        :RELative?

        :COUPling?[DEFault]
        :IMPedance?[DEFault]
        :FILTer[:LPASs]

        :FREQuency?[MINimum|MAXimum|DEFault][:STATe]?[DE-
        Fault]

        :GAIN?[DEFault|MINimum|MAXimum]

        :OFFSet[:ABSolute]?[MINimum|MAXimum|DEFault]:RELative?

INPut:SETup:AUTO?[DEFault]

INSTrument Subsystem

Commands	INSTrument
	:ABORt
	:RESet

MEASure? Subsystem

Queries   MEASure[1|2|3]([:SCALar]|:ARRay)

> :FREQuency? [<array size>[,<expected value>[,resolution>]]]

> :FREQuency:RATio? [<array size>[,<second channel>[,<expected value>[,resolution>]]]]

> :PERiod? [<array size>[,<expected value>[,resolution>]]]

MEASure[1|2]([:SCALar]|:ARRay)

> :NDUTycycle|PDUTycycle|DCYCle? [<array size>[,<reference>[,<expected value>[,resolution>]]]]

> :NWIDth|PWIDth? [<array size>[,<reference>[,<expected value>[,resolution>]]]]

> :PHASe? [<array size>[,<expected value>[,resolution>]]]

> :RTIMe|FTIMe|RISE:TIME|FALL:TIME? [<array size>[,<low reference>[,<high reference> [,<expected value>[,resolution>]]]]]

> :TINTerval? [<array size>[,<expected value>[,resolution>]]]

> [:VOLTage]

>> :AC? [<array size>[,<expected value>[,resolution>]]]

>> :DC? [<array size>[,<expected value>[,resolution>]]]

>> :MINimum? [<array size>[,<expected value>[,resolution>]]]

>> :MAXimum? [<array size>[,<expected value>[,resolution>]]]

>> :PTPeak? [<array size>[,<expected value>[,resolution>]]]

MEASure[1|2|11|12|21|22]([:SCALar]|:ARRay)

> :TINTerval:DELay:TIME|EVENts? [<array size>[,<delay time>|<delay events>[,<expected value>[,resolution>]]]]

MEASure[1|2|10|20]([:SCALar]|:ARRay)

> :TOTalize?

NOTE: If SCALar is specified, the <array size is not valid. If ARRay is specified, <array size> is a required parameter.

OUTPut Subsystem

Commands	OUTPut:TRIGger:SOURce <channel>,<count>
Queries	OUTPut:TRIGger:SOURce?

READ? Subsystem

Queries	READ?

SENSe Subsystem

Commands	SENSE:APERture <time>|DEFault|MINimum|MAXimum
	SENSe:COUNt <array size>|DEFault|MINimum|MAXimum
	SENSe:MODe <mode>
	SENSe:EVENts <# of events>
	SENSe[1|2|3]:FUNCtion
	
	    ″TOTALize″
	    ″FREQuency″
	
	        ″FREQuency:RATio<second channel>″
	
	        ″PERiod″
	
	SENSe[1,2]:FUNCtion
	
	    ″DCYCle″
	    ″FALL:TIME″
	    ″FTIMe″
	    ″NDUTycycle″
	    ″NWIDth″
	    ″PDUTycycle″
	    ″PHASe″
	    ″PWIDth″
	    ″RISE:TIME″
	    ″RTIMe″
	    ″TINTerval″
	    ″VOLTage:AC″
	    ″VOLTage:DC″
	    ″VOLTage:MINimum″
	    ″VOLTage:MAXimum″
	    ″VOLTage:PTPeak″
	
	SENSe[1|2|11|12|21|22]:FUNCtion
	
	    ″TINTerval:DELay:TIME″
	    ″TINTerval:DELay:EVENts″
	
	SENSe[1|2|10|20]:FUNCtion″TOTalize″
	SENSe:TINTerval:DELay:EVENts <events>|MINimum|MAXimum|DEFault
	SENSE:TINTerval:DELay:TIME <time>|DEFault|MINimum|MAXimum

Queries	SENSe:APERture?[DEFault|MINimum|MAXimum] SENSe:COUNt?[DEFault|MINimum|MAXimum] SENSe:EVENts?[DEFault|MINimum|MAXimum] SENSe:MODe?[DEFault] SENSe[1|2|3]:FUNCtion? SENSe:TINTerval:DELay:EVENts?[DEFault]|MINimum]|MAXimum]

SOURce Subsystem

Commands	SOURce:COSCillator[:SOURce] <source>
Queries	SOURce:COSCillator[:SOURce]? SOURce:COSCillator:VALue?

STATus Subsystem

Queries	STATus:OPERation:CONDition?

TEST Subsystem

Queries	TEST:ALL?

UNIT Subsystem

Commands	UNIT:ANGLeDEGree|RADian[,MINimum|CENTer|AUTo]
Queries	UNIT:ANGLe?

Digital Input Commands

CALibration Subsystem

Commands	CALibration:VALue <cal value>

CONFigure Subsystem

Commands	CONFigure[:SCALar]
	:DIGLobal
	[:NORMal] <voltage_threshold>
	:INVerted <voltage_threshold>
	:DIPort
	[:NORMal] <voltage_threshold>,<numeric_port_list>
	:INVerted <voltage_threshold>,<numeric_port_list>
	:DIBit
	[:NORMal] <voltage_threshold>,<channel_bit_list>
	:INVerted <voltage_threshold>,<channel_bit_list>
	CONFigure[:ARRay]
	:DIGLobal
	[:NORMal] <repetitions>,<voltage_threshold>
	:INVerted <repetitions>,<voltage_threshold>
	:DIPort
	[:NORMal] <repetitions>,<voltage_threshold>,numeric_port_list>
	:INVerted <repetitions>,<voltage_threshold>,numeric_port_list>
	:DIBit
	[:NORMal] <repetitions>,<voltage_threshold>,<channel_bit_list>
	:INVerted <repetitions>,<voltage_threshold>,channel_bit_list>
Queries	CONFigure?

FETch? Subsystem

Queries	FETCh?[<count>[<offset>[,<step_size>]]] FETCh:COUNt?

FORMat Subsystem

Commands	FORMat[:DATA] <format>
Queries	FORMat[:DATA]?

INITiate Subsystem

Commands	INITiate:[IMMediate] <control>

INSTrument Subsystem

Commands	INSTrument:ABORt[IMMediate] INSTrument:RESet

MEASure? Subsystem

Queries   MEASure[:SCALar]

           :DIGLobal

                [:NORMal]? <voltage_threshold>
                :INVerted? <voltage_threshold>

           :DIPort

                [:NORMal]? <voltage_threshold>,<numeric_port_list>
                :INVerted? <voltage_threshold>,<numeric_port_list>

           :DIBit

                [:NORMal]? <voltage_threshold>,<numeric_port_list>
                :INVerted? <voltage_threshold>,<numeric_port_list>

           MEASure[:ARRay]

           :DIGLobal

                [:NORMal]? <repetitions>,<voltage_threshold>
                :INVerted? <repetitions>,<voltage_threshold>

           :DIPort

                [:NORMal]? <repetitions>,<voltage_threshold>,<numeric_port_list>
                :INVerted? <repetitions>,<voltage_threshold>,<numeric_port_list>

           :DIBit

           [:NORMal]? <repetitions>,<voltage_threshold>,<channel_bit_list>
           :INVerted? <repetitions>,<voltage_threshold>,<channel_bit_list>

READ? Subsystem

Queries   READ?

SENSe Subsystem

Commands	SENSe:ARRay <array size> SENSe:MODE <mode> SENSe:PSELect
	:DIBit
	[:NORMal] <ENABle|DISable>,<channel_bit_list>|<numerical_bit_mask> :INVerted <ENABle|DISable>,<channel_bit_list>|<numerical_bit_mask>
	:DIGLobal
	[:NORMal] <ENABle|DISable> :INVerted <ENABle|DISable>
	:DIPort
	[:NORMal] <ENABle|DISable>,<port list> :INVerted <ENABle|DISable>,<port list>
	SENSe:SRATe <sample rate> SENSe:THReshold <voltage_threshold>
Queries	SENSe:ARRay? SENSe:MODE? SENSe:PSELect? SENSe:SRATe? SENSe:THReshold?

STATus Subsystem

Queries	STATus:OPERation:CONDition?

TEST Subsystem

Queries	TEST:ALL?

TRIGger Subsystem

Commands	TRIGger:MASK <mask>
	TRIGger:MATCh <pattern>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuestNORMal|INVerted <polarity>
	:STRobeNORMal|INVerted <polarity>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
Queries	TRIGger:MASK?
	TRIGger:MATCh?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest?
	:STRobe?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Digital Output Commands

FORMat Subsystem

Commands	FORMat[:DATA]<format>
Queries	FORMat [:DATA]?

INITiate Subsystem

Commands	INITiate
	:CONTinuous <control>
	[:IMMediate]
Queries	INITiate:CONTinuous?

INSTrument Subsystem

Commands	INSTrument:ABORt ([IMMediate]|:COMPlete
	INSTrument:RESet

OUTPut Subsystem

Commands	OUTPut:LEVel <voltage level>
	OUTPut:TRIGger:SOURce <source>
Queries	OUTPut:LEVel?
	OUTPut:TRIGger:SOURce?

STATus? Subsystem

Queries	STATus:OPERation:CONDition?

TEST Subsystem

Queries	TEST:ALL?

TRACe Subsystem

Commands	TRACe:CLEar
	TRACe[:DATA]<indices>{,<ascii hex_data>}
	TRACe|DATA[:DATA] <index>,{,ascii_hex_data, indefinite_binary_data}
	TRACe:POINts <number_of_points>
	TRACe:RFRequency <frequency>
	TRACe:RPERiod <repeat_period>
	TRACe:SRATe <sample_rate>
Queries	TRACe[:DATA]? <numeric_list>
	TRACe:INDice?
	TRACe:POINts?
	TRACe:RFRequency?
	TRACe:RPERiod?
	TRACe:SRATe?

TRIGger Subsystem

Commands	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest <polarity>
	:STRobe <polarity>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
Queries	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest?
	:STRobe?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Digital to Analog Converter (DAC) Commands

CALibrate Subsystem

Commands	`CALibrate[<channel>]:OUTput <channel >` CALibrate[<channel>]:VALue <channel><DVM reading> CALibrate[<channel>]:SAVE

FORMat Subsystem

Commands	FORMat[:DATA] <format>
Queries	FORMat[:DATA]?

INITiate Subsystem

Commands	INITiate
	    :CONTinuous[0|OFF|1|ON]
	    [:IMMediate]

Queries	INITiate:CONTinuous?

INSTrument Subsystem

Commands	INSTrument:ABORt ([:IMMediate]|COMPlete) INSTrument:RESet

OUTPut Subsystem

Commands	OUTPut:TRIGger:SOURce <trigger source>
Queries	OUTPut:TRIGger:SOURce?

STATus? Subsystem

Queries	STATus:OPERation:CONDition?

TEST Subsystem

Queries	TEST:ALL?

TRACe Subsystem

Commands	TRACe:Clear
	TRACe|DATA[:DATA]
	\<Channel\>,\<index\>{,ASCII Trace Voltages|BINary Indefinite Block TRACe Voltages\>}
	\<Channel\>,\<Numeric List of Indices\>{,\<ASCII Trace Voltages\>}
	TRACe:POINts\<Trace Buffer Points\>
	TRACe:RFRequency\<repeat frequency|OFF\>
	TRACe:RPERiod\<repeat period|OFF\>
	TRACe:SRATe\<sample_rate\>
Queries	TRACE|DATA[:DATA]?\<Channel Select\>|\<Numeric List of Indices\>
	TRACe:INDice?
	TRACe:POINts?
	TRACe:RFRequency?
	TRACe:RPERiod?
	TRACe:SRATe?

TRIGger Subsystem

Commands	TRIGger:COUNt \<count\>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay \<time delay\>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt \<event delay\>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest \<polarity\>
	:STRobe \<polarity\>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE \<mode\>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce \<source\>
Queries	TRIGger:COUNt?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest?
	:STRobe?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?

VXI:FDC Subsystem

Commands	VXI[:SERVant]:FDC:CLOSe[<channel number>]
	VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
	VXI[:SERVant]:FDC:SEL <channel number>
	VXI[:SERVant]:FDC:BUFFer <buffer length>,[,<channel number>]
Queries	VXI[:SERVant]:FDC?
	VXI[:SERVant]:FDC:CONFiguration?[<channel number>]
	VXI[:SERVant]:FDC:SEL?
	VXI[:SERVant]:FDC:BUFFer?[<channel number>]

Digital Multimeter (DMM) Commands

CALCulate Subsystem

Commands   CALCulate:LIMit

       :ENVelope[:DATA] <threshold1>,<threshold2>
       :LOWer[:DATA] <threshold>
       :REPort[:DATA]
       :UPPer[:DATA] <threshold>

       CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
       CALCulate:TRANsform:HISTogram:POINts <numeric_value>
       CALCulate:TRANsform:HISTogram:RANGe <min>.<max>
       CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>

Queries   CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]
       CALCulate:LIMit:ENVelope[:DATA]
       CALCulate:LIMit:FCOunt?
       CALCulate:LIMit:LOWer[:DATA]?
       CALCulate:LIMit:REPort[:DATA]?
       CALCulate:LIMit:UPPer[:DATA]?
       CALCulate:MEDian?[<count>[,<offset>[,<step_size>]]]
       CALCulate:MINimum?[<count>[,<offset>[,<step_size>]]]
       CALCulate:MAXimum?[<count>[,<offset>[,<step_size>]]]
       CALCulate:SDEViation?[<count>[,<offset>[,<step_size>]]]
       CALCulate:TRANsform:HISTogram?

          :ABOVe?
          :BELow?
          :COUNt?
          :POINts?
          :RANGe?
          :RANGe:AUTO?

       CALCulate:VARiance?[<count>[,<offset>[,<step_size>]]]

CALibrate Subsystem

Commands	CALibrate:LFRequency <line frequency>
	CALibrate:VALue <frequency>
	CALibrate:ZERO:AUTO <auto>
Queries	CALibrate:LFREquency?
	CALibrate:ZERO:AUTO?

CONFigure Subsystem

Commands	CONFigure([:SCALar]|:ARRay)[:...]
	[:VOLTage]
	:AC [<array size>[,<Expected Value>[,<Resolution>]]]
	:ACDC [<array size>[,<Expected Value>[,<Resolution>]]]
	:DC [<array size>[,<Expected Value>[,<Resolution>]]]
	:CURRent [:DC][<array size>[,<Expected Value>[,<Resolution>]]]
	:RESistance [<array size>[,<Expected Value>[,<Resolution>]]]
	:FRESistance [<array size>[,<Expected Value>[,<Resolution>]]]
Queries	CONFigure?

FETCh? Subsystem

Queries	FETCh? [<Count>[,<Offset>[,<Step Size>]]]
	FETCh:COUNt?

INITiate Subsystem

Commands	INITiate[:IMMediate]
	INITiate:CONTinuous [<Boolean>]
Queries	INITiate:CONTinuous?

INPut Subsystem

Commands	INPut:IMPedance <Impedance>
Queries	INPut:IMPedance?

INSTrument Subsystem

Commands	INSTrument:ABORt
	INSTrument:RESet

MEASure? Subsystem

Queries	MEASure([:SCALar]|:ARRay)
	[:VOLTage]
	:AC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	:ACDC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	[:DC]? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	:CURRent[:DC]? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	:RESistance? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	:FRESistance? [<Array Size>[,<Expected Value>[,<Resolution>]]]
	NOTE: If SCALar is specified, the <array size is not valid. If ARRay is specified, <array size> is a required parameter.

READ? Subsystem

Queries	READ?

SENSe Subsystem

Commands	SENSe:BANDwidth:DETector \<expected frequency>|DEFault|MINimum|MAXimum
	SENSe:FUNCtion \<Measurement Function>
	SENSe:VOLTage
	[:DC]
	:AC
	:ACDC
	:RANGe[:UPPer] \<Input Voltage Range>
	:AUTO \<auto>
	:APERture \<Aperture Value>
	:RPSecond \<Readings/Second>
	:NPLCycles \<Number of Power Line Cycles>
	:RESolution \<Expected Resolution>
	:COUNt \<Array Size>
	SENSe:CURRent[:DC]
	:RANGe[:UPPer] \<Input Current Range>
	:AUTO \<auto>
	:APERture \<Aperture Value>
	:RPSecond \<Readings/Second>
	:NPLCycles \<Number of Power Line Cycles>
	:RESolution \<Expected Resolution>
	:COUNt \<Array Size>
	SENSe:[RESistance|:FRESistance]
	:RANGe[:UPPer] \<Input Resistance Range>
	:AUTO \<Boolean|ONCE>
	:APERture \<Aperture Value>
	:RPSecond \<Readings/Second>
	:NPLCycles \<Number of Power Line Cycles>
	:RESolution \<Expected Resolution>
	:COUNt \<Array Size>
Queries	SENSe[...]?
	SENSe:BANDwidth:DETector?

STATus? Subsystem

Commands	STATUS:OPERation:CONDition?

TEST Subsystem

Commands	TEST:ALL?

STATus? Subsystem

Commands	STATUS:OPERation:CONDition?

TRIGger Subsystem

Commands	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest NORMal|INVerted <polarity>
	:STRobe NORMal|INVerted <polarity>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
Queries	TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
	:REQuest?
	:STRobe?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
	TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?

VXI:FDC Subsystem

Commands	VXI[:SERVant]:FDC:SEL <channel number>
	VXI[:SERVant]:FDC:CLOSe [<channel number>]
	VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
Queries	VXI[:SERVant]:FDC:SEL? <channel number>

Relay Driver Commands

ROUTe Subsystem

Commands	[ROUTe:]CLOSe <channel list>
	[ROUTe:]OPEN <channel list>
Queries	[ROUTe]:CLOSe:STATe?

STATus Subsystem

Queries	STATus:OPERation:CONDition?

SurePath™ Commands

INITiate Subsystem

Commands	INITiate
	:CONTinuous [0|OFF|1|ON]
	[:IMMediate]
Queries	INITiate:CONTinuous?

INSTrument Subsystem

Commands	INSTrument:ABORt
	INSTrument:RESet

ROUTe Subsystem

Commands	[ROUTe:]CLOSe <channel list>
	:DWELl
	:MODE <mode>,<module_name>,<section_list>
	[ROUTe:]CONFiqure <configuration>,<module_name>,<section_list>
	:DISJoin <module name>
	:JOIN <module_name>,<section_list>
	[ROUTe:]MODule
	:DELete[:NAME] <module_name>
	:DELete:ALL
	[:DEFine] <module_name>,<nrf>
	[ROUTe:]OPEN
	:ALL[module_name]
	:DWELl <channel_list>
	[ROUTe:]PFAil <action_at_powefail>
	[ROUTe:]SCAN <channel_list>
	:RATE <scan_rate>,<module_name>
Queries	[ROUTe:]CLOSe? <channel_list>
	[ROUTe:]ID?
	[ROUTe:]MODule[:DEFine]?<module_name
	[ROUTe:]OPEN? <channel_list>
	[ROUTe:MODule:CATalog?
	[ROUTe:MODule:CATalog:SUPPorted?

STATus Subsystem

Queries	STATus:OPERation:CONDition?

TRIGger Subsystem

Commands	TRIGger([:SEQuence]|:STARt)[:LAYer]
	:DELay \<delay in seconds\>
	:COUNt \<count\>
	:ECOunt \<triggers to count\>
	:IMMediate
	:SOURce \<source\>
Queries	TRIGger([:SEQuence1]|:STARt)[:LAYer]
	:COUNt?
	:DELay?
	:ECOunt?
	:MODE?
	:SOURce?
	:CATalog[:ALL]?
	:CATalog:DELayable?
	:CATalog:FIXed?

IEEE 488.2 Commands

These commands are described in full in the *IEEE 488.2 Common Commands* section.

*CAL?	Calibration Query; returns the state of the calibration routine.
*CLS	Clear Status; This commands clears the Event Status Register (ESR) and any pending Service Requests (SRQs).
*ESE <mask>	Event Status Enable (ESE) command; defines the mask for event status reporting.
*ESE?	Event Status Enable (ESE) query; returns the value of the Event Status Enable register.
*ESR?	Event Status Register (ESR) query; returns the value of the Event Status Register.
*IDN?	Identification Query; Returns a four field response.
*OPC	Operation Complete command; controls setting the OPC bit in the ESR.
*OPC?	Operation Complete query; puts a 1 in the output queue when all pending operations have been completed.
*RST	Reset; Resets the module to its power on state.
*SRE	Service Request Enable (SRE) register; defines the mask for generating VXI Request True interrupts.
*SRE?	Service Request Enable (SRE) query; returns the value of the Service Request Enable register.
*STB?	Status Byte (SB) query; returns the value of the status byte register.
*TST	Execute self test.
*TST?	Returns a summary of the self test results.
*WAI	Wait to Continue; suspends command processing until all pending operations have been completed.

STATus and Event Commands

The VX4101A Status commands provide status information on all instruments.

Commands	STATus:PRESet
	STATus:OPERation
	:ENABle
	:NTRansition
	:PTRansition
	STATus:QUEue
	:ENABle <numeric list>
	STATus:QUEStionable
	:ENABle
	:NTRansition
Queries	STATus:PRESet
	STATus:OPERation
	:PTRansition
	STATus:QUEue
	:ENABle?
	[:NEXT]?
	STATus:QUEStionable
	[:EVENt]?
	:CONDition?
	:ENABle?
	:NTRansition?
	:PTRansition?