**Agilent VEE SCPI Tool Readme**

Current Version: 1.01

Last Updated: 3/16/2009

**Software Disclaimer**

The software tools and utilities that have been written by either Agilent or third parties are made available for Agilent's general customers solely as a convenience. It is provided only on an "AS IS" basis and on the express condition that it is neither warranted nor supported by Agilent Technologies, Inc. AGILENT DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING THE IMPLIED WARRANTY OF MERCHANTABILITY.

You have a royalty-free right to use, modify, reproduce and distribute this SCPI Tool (and/or any modified version) in any way you find useful, provided that you agree that Agilent Technologies has no warranty, obligations or liability for any SCPI XML files created.

**Table of Contents**

[**How to use the SCPI tool**](file:///\\agmbidae\bidae\Resources\XMLfiles\Agilent%20VEE%20SCPI%20Tool%20Readme.htm#how_to_use)

[**Introduction to the SCPI language**](file:///\\agmbidae\bidae\Resources\XMLfiles\Agilent%20VEE%20SCPI%20Tool%20Readme.htm#introduction)

Welcome to use this Agilent VEE SCPI tool. This tool is designed to help you to create SCPI command files (in XML format) that can be used by Agilent VEE to provide SCPI completion in Agilent VEE.

**How to use the SCPI tool:**

**1        Create a text file that contents all SCPI commands needed for an instrument**

* Create an empty SCPI command file in a text editor, for example the Microsoft Notepad.
* Add SCPI commands and descriptions as desired into the text file.  You can use the following example SCPI command file as a reference.
* Save the above text file as a .txt file, for example ScpiExample.txt.

**2    Use the SCPI tool to convert the SCPI command file from text format to XML format**

* Launch **ScpiTool.exe**.
* Specify information of the instrument, such as the Name and Manufacture of the instrument. The information will be used by the VEE SCPI Completion feature.
* Uncheck the “Include Common Commands” option if you don’t need to include common SCPI commands into the XML SCPI file.
* Click **Browse** and select the text file (ScpiExample.txt) to be converted. The tool will convert the selected file to XML format and display it in the left pane. Error information will be displayed in the right pane if errors occur during the conversion.
* Click **Save** to save the SCPI command file in XML format.
* To use this SCPI command file in VEE SCPI Completion, you need to save it in "**~installdir\ScpiXmlFiles**" folder.  For example, *C:\Program Files\Agilent\VEE Pro 9.0\SCPIXmlFiles.*

**The example SCPI command file below can be used as a reference or start point when you create your own SCPI command files.**

# ScpiExample.txt

# SCPI command must has "Syntax:" key word. "Description:" and "Example:" are optional.

# An example of SCPI command with description and examples.

Syntax: READ?

Description: This command changes the instrument's triggering system from the "idle" state to the "wait-for-trigger" state. Measurements will begin when the specified trigger conditions are satisfied following the receipt of the READ? command. Readings are then sent immediately to volatile memory and the instrument's output buffer.

Example:

CONF:VOLT:DC 10,0.003

TRIG:SOUR EXT

READ?

# An example of SCPI command with optional keyword.

Syntax: CALibration[:ALL]?

Description: This command performs a calibration of the multimeter using the specified calibration value (CALibration:VALue command). Before you can calibrate the instrument, you must unsecure it by entering the correct security code.

Example: CAL?

# An example of SCPI command with suffix, suffix is [1|2|3|4].

Syntax: OUTPut:ALARm[1|2|3|4]:CLEar

Description: Clear the specified alarm output lines.

# An example of SCPI command with suffix, suffix is <1-7>.

Syntax:

[SOURce:]DIGital:PIN<1-7>:FUNCtion {<function>|DIO | DINPut | TOUTput | TINPut | FAULt | INHibit | ONCouple | OFFCouple}

[SOURce:]DIGital:PIN<1-7>:FUNCtion?

Description:

This command sets the functions of the digital port pins.

Example:

DIG:PIN1:FUNC FAUL

DIG:PIN7:FUNC?

# An example of SCPI command with suffix, suffix is [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19].

Syntax:

[:SOURce]:MARKer[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19][:STATe] {ON|OFF|1|0}

[:SOURce]:MARKer[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19][:STATe]?

Description:

This command turns a marker on or off. Marker 0 is the default if the marker designator [n] is not specified.

MARKer has suffix [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19]

Example:

SOURce:MARKer0:STATe ON

SOURce:MARKer0:STATe?

# An example of SCPI command with Numeric Parameters.

Syntax: CONFigure[:VOLTage]:AC [{<range>|AUTO|MIN|MAX|DEF} [,{<resolution>|MIN|MAX|DEF}]]

Description: First, this command resets all ac voltage measurement parameters and trigger parameters to their default values. Then, it configures the meter for ac voltage measurements but does not initiate the measurement.

Example:

CONF:VOLT:AC

READ?

# An example of SCPI command with Discrete Parameters

Syntax: CALCulate:FUNCtion {NULL | DB | DBM | AVERage | LIMit}

CALCulate:FUNCtion?

Description: This command selects the calculation function to be used. The default function is NULL.

Example:

CALC:FUNC DBM

CALC:FUNC?

# An example of SCPI command with Boolean Parameters

Syntax: TRIGger:DELay:AUTO {OFF|0|ON|1}

TRIGger:DELay:AUTO?

Description: This command disables or enables an automatic trigger delay. If enabled, the instrument determines the delay based on function, range, and integration time or bandwidth.

Example:

TRIG:DEL:AUTO ON

TRIG:DEL:AUTO?

# An example of SCPI command with ASCII String Parameters

Syntax:

DISPlay[:WINDow[{1|2}]]:TEXT[:DATA] <String>

DISPlay[:WINDow[{1|2}]]:TEXT[:DATA]?

Description:

This command displays a text message on the top line ("WINDow1") or bottom line ("WINDow2") of the instrument's front-panel display. The top line ("WINDow1") is the default. The instrument can display up to 12 (top line) or 18 (bottom line) characters on the front panel; any additional characters are truncated (no error is generated).

Example:

DISP:WIND2:TEXT "WAITING FOR DATA..."

DISP:WIND2:TEXT?

# An example of SCPI command with fixed String Parameters

Syntax: DATA:COPY NVMEM, RDG\_STORE

Description: This command copies the readings in reading memory (RDG\_STORE) into non-volatile memory (NVMEM). Non-volatile memory is not erased by a power-down cycle.

Example:

DATA:COPY NVMEM, RDG\_STORE

# An example of SCPI command with Integer Parameters

Syntax: DATA:REMove? <Integer>

Description: This command reads and erases the specified number of readings from memory. The readings are erased from memory starting with the oldest reading first. The purpose of this command is to allow you to periodically remove readings from memory during a series of measurements to avoid a reading memory overflow.

Example:

DATA:REM? 3

# An example of SCPI command with channel list. chanlist, channel, ch\_list and scan\_list are identic

Syntax:

[SENSe:]COUNter:TOTalize:IVALue? (@<ch\_list>)

Description:

The query command returns a numeric value representing the initial counter value of the general purpose digital counter. Multiple responses are separated by comma.

# An example of SCPI command with optional parameters

Syntax:

APPLy:RAMP [<frequency> [,<amplitude> [,<offset>] ]]

Description:

Output a ramp wave with the specified frequency, amplitude, and dc offset. This command overrides the current symmetry setting and automatically selects 100%. The waveform is output as soon as the command is executed.

**The end of the example SCPI command file.**

[**[Top]**](file:///\\agmbidae\bidae\Resources\XMLfiles\Agilent%20VEE%20SCPI%20Tool%20Readme.htm#top1)

**Introduction to the SCPI language**

SCPI (Standard Commands for Programmable Instruments) is an ASCII-based instrument command language designed for test and measurement instruments. SCPI commands are based on a hierarchical structure, also known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems. A portion of the SENSe subsystem is shown below to illustrate the tree system.

SENSe:  
         VOLTage:   
                  DC:RANGe {<range>|MIN|MAX|DEF}  
                  DC:RANGe? [MIN|MAX]   
  
         RESistance:   
                  OCOMpensated {OFF|0|ON|1}  
                  OCOMpensated?

**SENSe** is the root keyword of the command, **VOLTage** and **RESistance** are second-level keywords, and **DC** and **OCOMpensated** are third-level keywords. A colon ( **:** ) separates a command keyword from a lower-level keyword.

**Syntax Conventions**

The format used to show commands is illustrated below:

VOLTage:DC:RANGe {<range>|MIN|MAX|DEF}

The command syntax shows most commands (and some parameters) as a mixture of upper- and lower-case letters. The upper-case letters indicate the abbreviated spelling for the command. For shorter program lines, you can send the abbreviated form. For better program readability, you can send the long form.

For example, in the above syntax statement, VOLT and VOLTAGE are both acceptable forms. You can use upper- or lower-case letters. Therefore, VOLTAGE, volt, and Volt are all acceptable. Other forms, such as VOL and VOLTAG, are not valid and will generate an error.

·         Braces ( **{ }** ) enclose the parameter choices for a given command string. The braces are not sent with the command string.

·         A vertical bar ( **|** ) separates multiple parameter choices for a given command string. For example, {*<range>*|MIN|MAX|DEF} in the above command indicates that you can specify a numeric range parameter, or "MIN", "MAX", or "DEF". The bar is not sent with the command string.

·         Triangle brackets ( **< >** ) indicate that you must specify a value for the enclosed parameter. For example, the above syntax statement shows the *<range>* parameter enclosed in triangle brackets. The brackets are not sent with the command string. You must specify a value for the parameter (for  example "VOLT:DC:RANG 10") unless you select one of the other options shown in the syntax (for example "VOLT:DC:RANG MIN").

·         Some parameters are enclosed in square brackets ( **[ ]** ). This indicates that the parameter is optional and can be omitted. The brackets are not sent with the command string. If you do not specify a value for an optional parameter, the instrument chooses a default value.

**Command Separators**

A colon ( **:** ) is used to separate a command keyword from a lower-level keyword. You must insert a blank space to separate a parameter from a command keyword. If a command requires more than one parameter, you must separate adjacent parameters using a comma as shown below:

CONF:VOLT:DC 10,0.003

A semicolon ( **;** ) is used to separate commands within the same subsystem, and can also minimize typing. For example, sending the following command string:

TRIG:SOUR EXT**;** COUNT 10

... is the same as sending the following two commands:

TRIG:SOUR EXT   
TRIG:COUNT 10

Use a colon and a semicolon to link commands from different subsystems. For example, in the following command string, an error is generated if you do not use both the colon and semicolon:

TRIG:COUN MIN;:SAMP:COUN MIN

**Using the MIN, MAX, and DEF Parameters**

For many commands, you can substitute "MIN" or "MAX" in place of a parameter. In some cases you may also substitute "DEF". For example, consider the following command:

VOLTage:DC:RANGe {<range>|MIN|MAX|DEF}

Instead of selecting a specific value for the *<range>* parameter, you can substitute MIN to set the range to its minimum value, MAX to set the range to its maximum value, or DEF to set the range to its default value.

**Querying Parameter Settings**

You can query the current value of most parameters by adding a question mark ( **?**) to the command. For example, the following command sets the trigger count to 10 readings:

TRIG:COUN 10

You can then query the count value by sending:

TRIG:COUN?

You can also query the minimum or maximum count allowed as follows:

TRIG:COUN? MIN  
TRIG:COUN? MAX

**SCPI Command Terminators**

A command string sent to the instrument must terminate with a <new line> (<[NL](javascript:BSSCPopup('Misc_Definitions/New_Line.htm');) >) character. The IEEE-488 EOI (End-Or-Identify) message is interpreted as a <NL> character and can be used to terminate a command string in place of a <NL> character. A <carriage return> followed by a <NL> is also accepted. Command string termination will always reset the current SCPI command path to the root level.

**IEEE-488.2 Common Commands**

The IEEE-488.2 standard defines a set of common commands that perform functions such as reset, self-test, and status operations. Common commands always begin with an asterisk ( **\*** ), are three characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a blank space. Use a semicolon ( **;** ) to separate multiple commands as shown below:

\*RST**;** \*CLS**;** \*ESE 32**;** \*OPC?

**SCPI Parameter Types**

The SCPI language defines several data formats to be used in program messages and response messages.

**Numeric Parameters**

Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters such as MIN, MAX, and DEF are also accepted. You can also send engineering unit suffixes with numeric parameters (e.g.,  M, k, m, or u). If a command accepts only certain specific values, the instrument will automatically round the input numeric parameters to the accepted values. The following command requires a numeric parameter for the range value:

VOLTage:DC:RANGe {<range>|MIN|MAX|DEF}

|  |  |
| --- | --- |
| Note | Because the SCPI parser is case-insensitive, there is some confusion over the letter "M" (or "m"). For your convenience, the instrument interprets "mV" (or "MV") as millivolts, but "MHZ" (or "mhz") as megahertz. Likewise "MΩ" (or "mΩ") is interpreted as megohms. You can use the prefix "MA" for mega. For example, "MAV" is interpreted as megavolts. |

**Discrete Parameters**

Discrete parameters are used to program settings that have a limited number of values (like IMMediate, EXTernal, or BUS). They have a short form and a long form just like command keywords. You can mix upper- and lower-case letters. Query responses will always return the short form in all upper-case letters. The following command requires a discrete parameters for the temperature units:

UNIT:TEMPerature {C|F|K}

**Boolean Parameters**

Boolean parameters represent a single binary condition that is either true or false. For a false condition, the instrument will accept "OFF" or "0". For a true condition, the instrument will accept "ON" or "1". When you query a boolean setting, the instrument will always return "0" or "1". The following command requires a boolean parameter:

VOLTage:DC:IMPedance:AUTO {OFF|0|ON|1}

**ASCII String Parameters**

String parameters can contain virtually any set of ASCII characters. A string must begin and end with matching quotes; either with a single quote or a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between. The following command uses a string parameter:

DISPlay:TEXT <quoted string>

For example, the following command displays the message "WAITING..." on the instrument's front panel (the quotes are not displayed).

DISP:TEXT "WAITING..."

You can also display the same message using the following command with single quotes.

DISP:TEXT 'WAITING...'