

DC Power Supply

IVI-C Programming Guide

EN01A

Nov, 2023

Revision History

This chapter declares the modifications of IVI driver in the most recent release of the programming guide version.

Version E01A at Introduction

This version, as the first version, will be compared with later versions. When the next version is released, the differences between the two versions will be marked.

Models Supported

The series of SIGLENT Programmable DC Power Supply this IVI-C driver is shown below.

Series	Release Version Supporting IVI-CDriver
SPS5000X	3.1.1.8R3 and higher
SPS6225X	1.1.1.7R6 and higher
SPD4000X	4.1.2.6R2 and higher

Software Requirement

This chapter describes how to configure the IVI driver to control the instrument. If you want to use the IVI Driver, you must install NI-VISA, the IVI Compliance Package, and a C language development system that supports the IVI driver library.

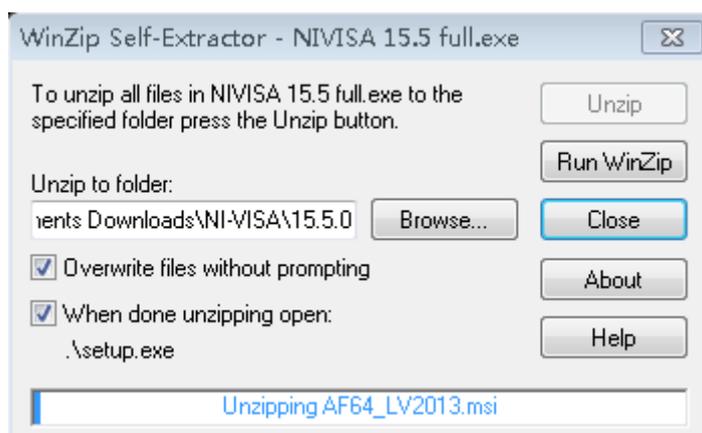
Install NI-MAX

Currently, NI-VISA is packaged in two versions: Full version and Run-Time Engine version. The full version includes the NI device drivers and a tool named NI-MAX which is a user interface to control and test remotely connected devices. You need to install the full version of NI-VISA.

You can get the NI-VISA 15.5 full version or higher version from :

<https://www.ni.com/en-us/support/downloads/drivers/download.ni-visa.html#306031>.

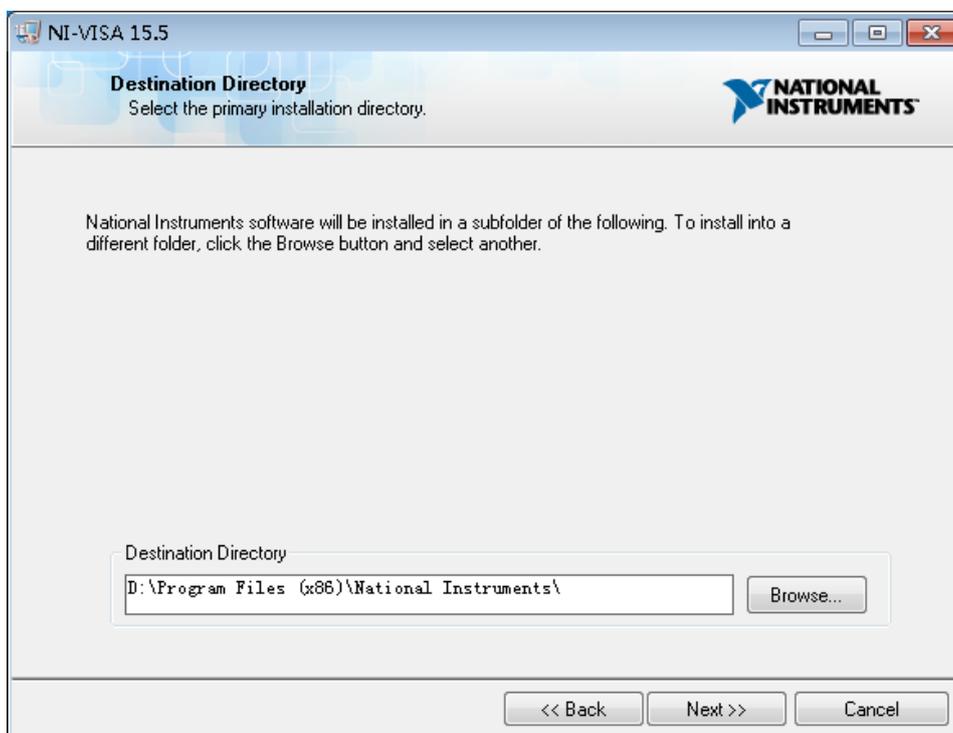
- a. Double click the NIVISA 15.5 full.exe, a dialog will be shown as below:



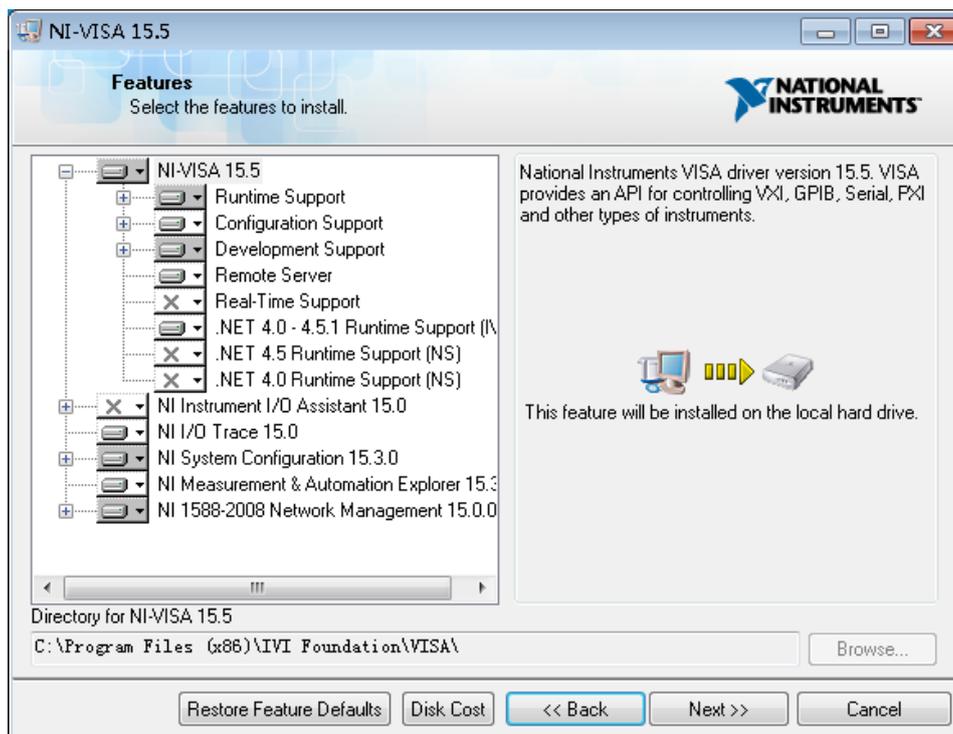
- b. Click Unzip, the installation process will automatically launch after unzipping files. If your computer needs to install .NET Framework 4, it may auto start.



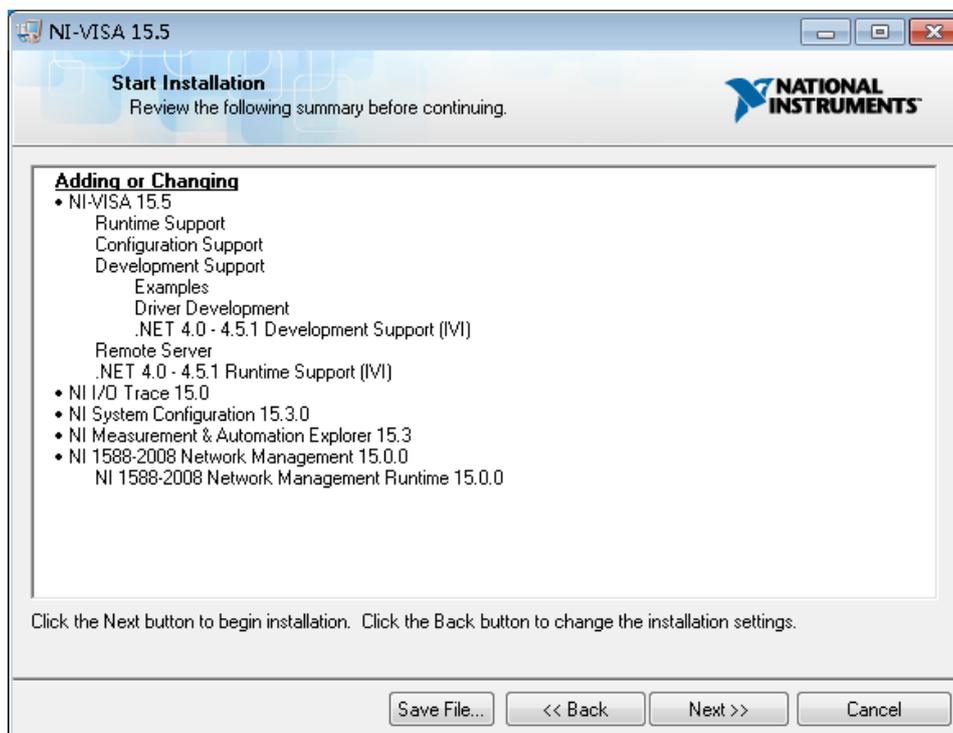
- c. The NI-VISA installingdialog is shown above. Click Next to start the installation process.



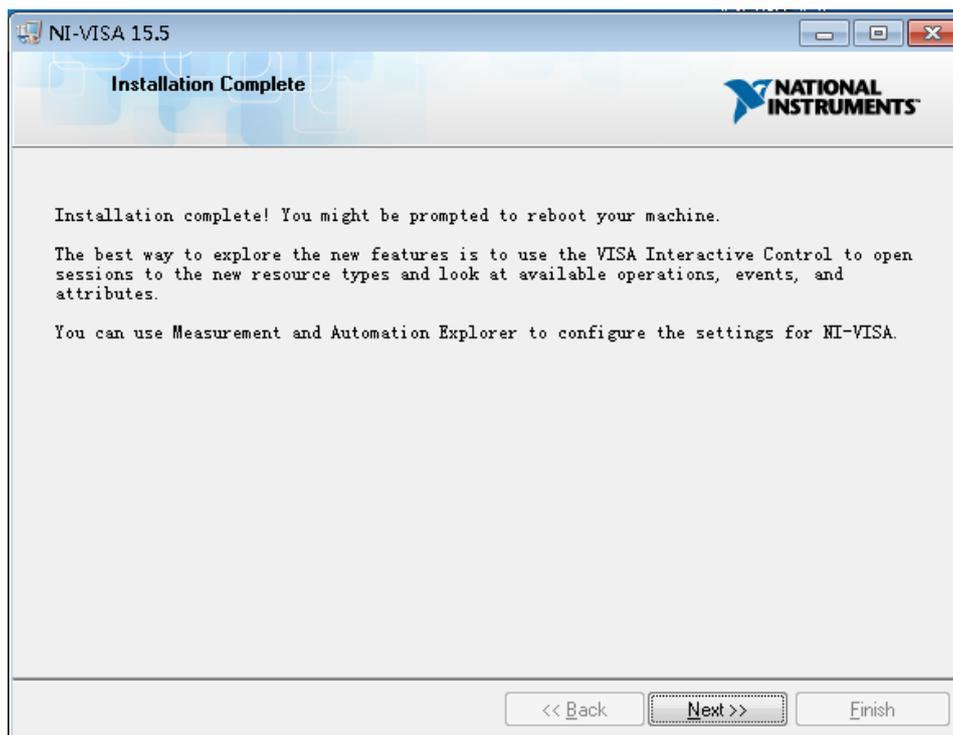
- d. Set the install path. The default path is “C:\Program Files\National Instruments\”. You can change it. Click Next.



- e. Click Next twice, in the License Agreement dialog, select “I accept the above 2 License Agreement(s).”, and click Next.



- f. Click Next to begin the installation.



- g. Wait until the installation is completed, and then reboot your PC.

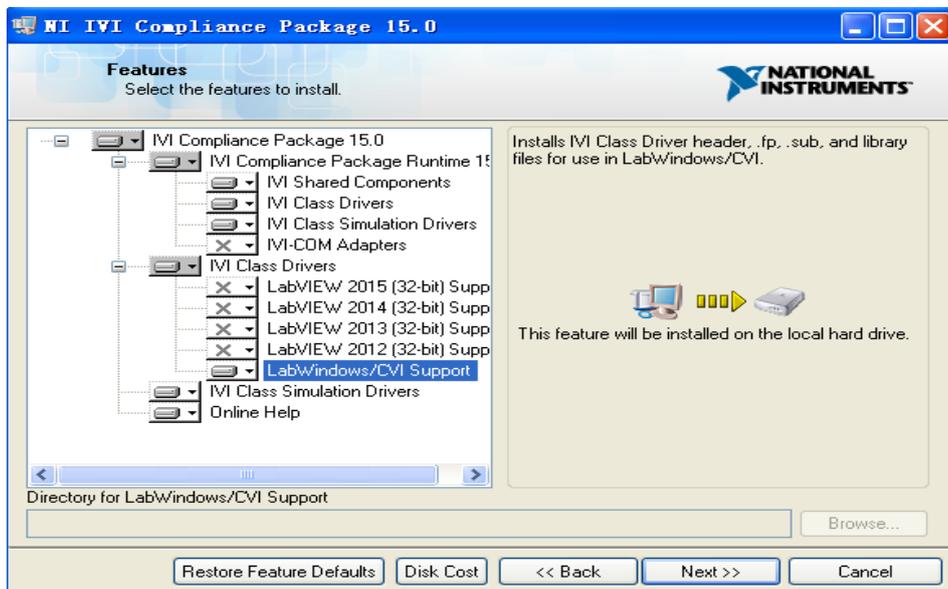
Install the IVI Compliance Package

The IVI Compliance Package contains the IVI class drivers and supported libraries for developing and leveraging IVI-based applications.

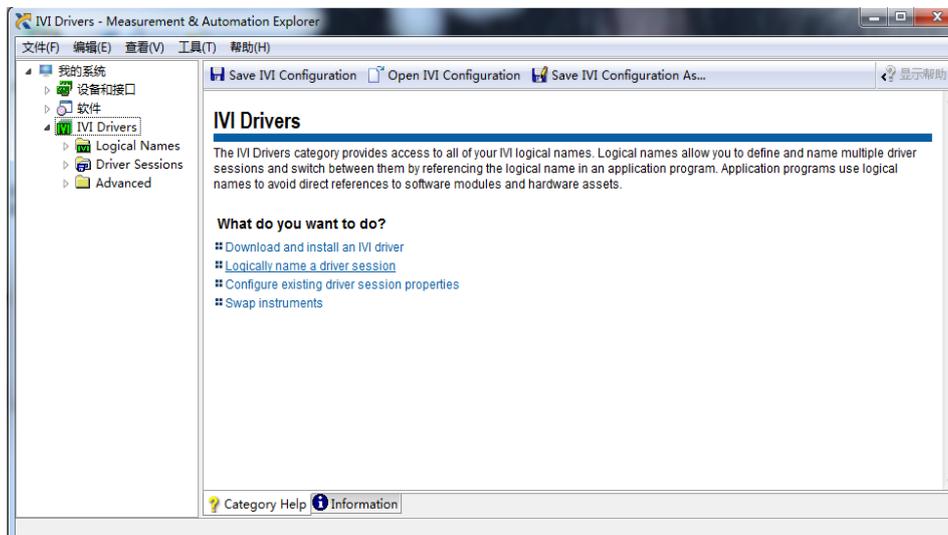
You can get the IVI Compliance Package from :

<https://www.ni.com/zh-cn/support/downloads/drivers/download.ivi-compliance-package.html#329444>

- a. If the IVI Compliance Package is not installed, there is no IVI Drivers option in "My System".
- b. Install the IVI Compliance Package (ICP).



- c. Restart your computer after the installation. After the reboot, the IVI Drivers option appears.



SPS IVI-C Driver Package List

The SPS IVI-C driver package provides three kinds of files: SPS_x32 file, SPS.h file and SPS_x64 file.

File	Description
SPS_x32.dll/ SPS_x64.dll	Adynamic link library file,including variables, functions, and data interfaces for various attributes.
SPS_x32.lib/ SPS_x64.lib	An import libraryfile, including the symbolic name and optional identification number of each exported function in the SPS_x32.dll file.
SPS.h	A header file,includingdeclarations of variables, functions, and data interfaces.

You include the SPS.h when programming the Siglent DC Power Supply with the IVI driver, and load the SPS.dll dynamic file or SPS_x32.lib import library file into your own project.

You will find an example that show you how to use these files at the end of this document.

Introduction to IVI

IVI (Interchangeable Virtual Instruments) is a new generation of instrument driver technology specifications introduced by the IVI Foundation. IVI can realize the interchangeability with the instrument, the instrument simulation, and the instrument state tracking and buffer function. All references to IVI drivers in this document refer to IVI-C drivers that are created using NI tools and that rely on the IVI Engine.

IVI Data Type

There are six data types for the attributes of the IVI Engine: ViInt32, ViReal64, ViString, ViBoolean, ViSession and ViAddr.

Table 1 Data Type

Data Type	Description
ViInt32	32-bit signed integer
ViReal64	64-bit floating-point number
ViString	String type
ViBoolean	Boolean value
ViSession	A VISA session handle
ViAddr	Logical address type

Attribute

This chapter describes the attributes of the SIGLENT IVI driver. The following table lists the supported IVI base class attributes and SIGLENT custom attributes.

System	Attribute
Basic Operation	SPS_ATTR_OUTPUT_ENABLED
	SPS_ATTR_OVP_ENABLED
	SPS_ATTR_OVP_LIMIT
	SPS_ATTR_CURRENT_LIMIT_BEHAVIOR
	SPS_ATTR_CURRENT_LIMIT
	SPS_ATTR_VOLTAGE_LEVEL
Source	SPS_ATTR_VOLTAGE_SET
	SPS_ATTR_OVP_SET
	SPS_ATTR_VOLTAGE_RISE_SLOPE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_VOLTAGE_FALL_SLOPE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_VOLTAGE_RESPONSE_MODE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_CURRENT_SET
	SPS_ATTR_OCP_SET
	SPS_ATTR_CURRENT_RISE_SLOPE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_CURRENT_FALL_SLOPE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_CURRENT_RESPONSE_MODE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_OPP_SET (Supported only by SPS6000X series)
	SPS_ATTR_VOLTAGE_CONTROL_MODE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_CURRENT_CONTROL_MODE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_EXTERNAL_CONTROL_STATE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_OUTPUT_STATE
	SPS_ATTR_OUTPUT_ON_DELAY_SET

	SPS_ATTR_OUTPUT_OFF_DELAY_SET
	SPS_ATTR_OUTPUT_MODE (Supported only by SPS6000X series)
	SPS_ATTR_CC_CV_PRIORITY (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_MEASURE_AVERAGE_GRADE (Only the SPS5000X and SPS6000X series are supported)
	SPS_ATTR_RESET_PROTECT_STATE
	SPS_ATTR_OCP_STATE_SET
	SPS_ATTR_LOAD_MODE (Supported only by SPS5000X series)
	SPS_ATTR_CHANNEL_ENABLE_STATE (Supported only by SPS5000X series)
	SPS_ATTR_OUTPUT_LOCAL_ENABLE_STATE (Supported only by SPS5000X series)
	SPS_ATTR_RESISTANCE_SET (Supported only by SPS5000X series)
	SPS_ATTR_OCP_DELAY_SET (Supported only by SPD4000X series)
	SPS_ATTR_OPERATION_MODE (Supported only by SPD4000X series)
	SPS_ATTR_SENSE_MODE (Supported only by SPD4000X series)
Measure	SPS_ATTR_VOLTAGE_MEASURE
	SPS_ATTR_CURRENT_MEASURE
	SPS_ATTR_POWER_MEASURE
	SPS_ATTR_CHANNEL_RUN_CCCV_MODE

Basic Operation

Attributes that control the basic features of the DC Power Supply. The basic group has the following attributes:

- **SPS_ATTR_OUTPUT_ENABLED**
- **SPS_ATTR_OVP_ENABLED**
- **SPS_ATTR_OVP_LIMIT**
- **SPS_VAL_CURRENT_LIMIT_BEHAVIOR_DISABLE_OUTPUT**
- **SPS_ATTR_CURRENT_LIMIT**
- **SPS_ATTR_VOLTAGE_LEVEL**

SPS_ATTR_OUTPUT_ENABLED

Description Set the output status of all channels.

Data type ViBoolean

Access R/W

Common Control Functions

```
ViStatus SPS_SetAttributeViBoolean (ViSession vi,
ViConstString channelName, ViAttr attribute, ViBoolean value);
ViStatus SPS_GetAttributeViBoolean (ViSession vi,
ViConstString channelName, ViAttr attribute, ViBoolean *value);
```

Notes:

vi is the instrument handle.

channelName: "CH1"~ "CH4"

attributeId is **SPS_ATTR_OUTPUT_ENABLED** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
Disable	VI_FALSE	0
Enable	VI_TRUE	1

SPS_ATTR_OVP_ENABLED

Description Set the overvoltage protection enable value.

Data type ViBoolean

Access R/W

Common Control Functions

ViStatus SPS_SetAttributeViBoolean (ViSession vi, ViConstString channelName, ViAttr attribute, ViBoolean value);

ViStatus SPS_GetAttributeViBoolean (ViSession vi, ViConstString channelName, ViAttr attribute, ViBoolean *value);

Notes:

vi is the instrument handle.

channelName: "CH1"~ "CH4"

attributeld is **SPS_ATTR_OVP_ENABLED** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Disable	VI_FALSE	0
Enable	VI_TRUE	1

Notes: Currently supported models will always enable overvoltage protection.

SPS_ATTR_OVP_LIMIT

Description	Set overvoltage protection value.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_OVP_LIMIT macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	The settable range is 10% to 110% of the rated voltage of the current model.

SPS_VAL_CURRENT_LIMIT_BEHAVIOR_DISABLE_OUTPUT

Description	Set the current behavior mode during overcurrent.
Data Type	ViInt32
Access	R
Common Control Functions	<pre>ViStatus SPS_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attribute, ViInt32 *value); ViStatus SPS_SetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attribute, ViInt32 value);</pre> <p>Notes: vi is the instrument handle. channelName is "CH1"~ "CH4". attributeld is SPS_VAL_CURRENT_LIMIT_BEHAVIOR_DISABLE_OUTPUT macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	NULL Notes: The current model only supports turning off output.

SPS_ATTR_CURRENT_LIMIT

Description	Set overcurrent protection value.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_CURRENT_LIMIT macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	The settable range is 10% to 110% of the rated current of the current model.

SPS_ATTR_VOLTAGE_LEVEL

Description	Set output voltage.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_VOLTAGE_LEVEL macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	The settable range is 0 to rated voltage.

Source

- **SPS_ATTR_VOLTAGE_SET**
- **SPS_ATTR_OVP_SET**
- **SPS_ATTR_VOLTAGE_RISE_SLOPE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_VOLTAGE_FALL_SLOPE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_VOLTAGE_RESPONSE_MODE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_CURRENT_SET**
- **SPS_ATTR_OCP_SET**
- **SPS_ATTR_CURRENT_RISE_SLOPE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_CURRENT_FALL_SLOPE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_CURRENT_RESPONSE_MODE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_OPP_SET** (Supported only by SPS6000X series)
- **SPS_ATTR_VOLTAGE_CONTROL_MODE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_CURRENT_CONTROL_MODE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_EXTERNAL_CONTROL_STATE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_OUTPUT_STATE**
- **SPS_ATTR_OUTPUT_ON_DELAY_SET**
- **SPS_ATTR_OUTPUT_OFF_DELAY_SET**
- **SPS_ATTR_OUTPUT_MODE** (Supported only by SPS6000X series)
- **SPS_ATTR_CC_CV_PRIORITY** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_MEASURE_AVERAGE_GRADE** (Only the SPS5000X and SPS6000X series are supported)
- **SPS_ATTR_RESET_PROTECT_STATE**
- **SPS_ATTR_OCP_STATE_SET**
- **SPS_ATTR_LOAD_MODE** (Supported only by SPS5000X series)
- **SPS_ATTR_CHANNEL_ENABLE_STATE** (Supported only by SPS5000X series)
- **SPS_ATTR_OUTPUT_LOCAL_ENABLE_STATE** (Supported only by SPS5000X series)

- **SPS_ATTR_RESISTANCE_SET** (Supported only by SPS5000X series)
- **SPS_ATTR_OCP_DELAY_SET** (Supported only by SPD4000X series)
- **SPS_ATTR_OPERATION_MODE** (Supported only by SPD4000X series)
- **SPS_ATTR_SENSE_MODE** (Supported only by SPD4000X series)

SPS_ATTR_VOLTAGE_SET

Description Set output voltage.

Data type ViReal64

Access R/W

Common Control Functions ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value);
ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);

Notes:

vi is the instrument handle.

channelName: "CH1"~ "CH4"

attributeld is **SPS_ATTR_VOLTAGE_SET** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range The settable range is 0 to rated voltage.

SPS_ATTR_OVP_SET

Description	Set overvoltage protection value.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_OVP_SET macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	The settable range is 10% to 110% of the rated voltage of the current model.

SPS_ATTR_VOLTAGE_RISE_SLOPE

Description	Set voltage rise slope (Only the SPS5000X and SPS6000X series are supported).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre>
	Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_VOLTAGE_RISE_SLOPE macro. value is used to store or set the value of function represented by attributeld .
Value Range	See user manual for range. SPS5000 series, unit V/s SPS6000 series, unit V/ms

SPS_ATTR_VOLTAGE_FALL_SLOPE

Description	Set voltage fall slope (Only the SPS5000X and SPS6000X series are supported).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_VOLTAGE_FALL_SLOPE macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	See user manual for range. SPS5000 series, unit V/s SPS6000 series, unit V/ms

SPS_ATTR_VOLTAGE_RESPONSE_MODE

Description Set voltage response mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeId is **SPS_ATTR_VOLTAGE_RESPONSE_MODE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
FAST MODE	SPS_VAL_VOLTAGE_CURRENT_RESPONSE_FAST	0
SLOPE MODE	SPS_VAL_VOLTAGE_CURRENT_RESPONSE_SLOPE	1

Notes:

In CV state, the mode takes effect.

SPS_ATTR_CURRENT_SET

Description	Set output current.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_CURRENT_SET macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	The settable range is 0 to rated current.

SPS_ATTR_OCP_SET

Description	Set overcurrent protection value.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre>
	<p>Notes:</p> <p>vi is the instrument handle.</p> <p>channelName: "CH1"~ "CH4"</p> <p>attributeId is SPS_ATTR_OCP_SET macro.</p> <p>value is used to store or set the value of function represented by attributeId.</p>
Value Range	The settable range is from 10% to 110% of the rated current value of the current model.

SPS_ATTR_CURRENT_RISE_SLOPE

Description	Set current rise slope (Only the SPS5000X and SPS6000X series are supported).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_CURRENT_RISE_SLOPE macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	See user manual for range. SPS5000 series, unit A/s SPS6000 series, unit A/ms

SPS_ATTR_CURRENT_FALL_SLOPE

Description	Set current fall slope (Only the SPS5000X and SPS6000X series are supported).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_CURRENT_FALL_SLOPE macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	See user manual for range. SPS5000 series, unit A/s SPS6000 series, unit A/ms

SPS_ATTR_CURRENT_RESPONSE_MODE

Description Set current response mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeId is **SPS_ATTR_CURRENT_RESPONSE_MODE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
FAST MODE	SPS_VAL_VOLTAGE_CURRENT_RESPONSE_FAST	0
SLOPE MODE	SPS_VAL_VOLTAGE_CURRENT_RESPONSE_SLOPE	1

Notes:

In CC state, the mode takes effect.

SPS_ATTR_OPP_SET

Description	Set power protection value (Supported only by SPS6000X series).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_OPP_SET macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	The settable range is from 10% to 110% of the rated power value of the current model.

SPS_ATTR_VOLTAGE_CONTROL_MODE

Description Set voltage control mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeId is **SPS_ATTR_VOLTAGE_CONTROL_MODE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
Local mode	SPS_VAL_VOLT_CURR_CTRL_LOCAL_MODE	0
Voltage mode	SPS_VAL_VOLT_CURR_CTRL_EXTERNAL_VOLTAGE_MODE	1
Resistance mode	SPS_VAL_5000X_VOLT_CURR_CTRL_EXTERNAL_RESISTANCE_MODE (Supported only by SPS5000X series)	2

SPS_ATTR_CURRENT_CONTROL_MODE

Description Set current control mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeId is **SPS_ATTR_CURRENT_CONTROL_MODE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
Local mode	SPS_VAL_VOLT_CURR_CTRL_LOCAL_MODE	0
Voltage mode	SPS_VAL_VOLT_CURR_CTRL_EXTERNAL_VOLTAGE_MODE	1
Resistance mode	SPS_VAL_5000X_VOLT_CURR_CTRL_EXTERNAL_RESISTANCE_MODE (Supported only by SPS5000X series)	2

SPS_ATTR_EXTERNAL_CONTROL_STATE

Description Set external control mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeId is **SPS_ATTR_EXTERNAL_CONTROL_STATE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
External control on	SPS_VAL_EXTERNAL_CTRL_OFF	0
External control off	SPS_VAL_EXTERNAL_CTRL_ON	1
External control trigger	SPS_VAL_EXTERNAL_CTRL_TRIGGER	2

SPS_ATTR_OUTPUT_STATE

Description Set output status.

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_OUTPUT_STATE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Output off	SPS_VAL_OUTPUT_OFF	0
Output on	SPS_VAL_OUTPUT_ON	1

SPS_ATTR_OUTPUT_ON_DELAY_SET

Description	Set the turn-on output delay.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre>
	<p>Notes:</p> <p>vi is the instrument handle.</p> <p>channelName: "CH1"~ "CH4"</p> <p>attributeld is SPS_ATTR_OUTPUT_ON_DELAY_SET macro.</p> <p>value is used to store or set the value of function represented by attributeld.</p>
Value Range	See user manual for range. (unit: seconds)

SPS_ATTR_OUTPUT_OFF_DELAY_SET

Description	Set the turn-off output delay.
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_OUTPUT_OFF_DELAY_SET macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	See user manual for range. (unit: seconds)

SPS_ATTR_OUTPUT_MODE

Description Set output mode(Supported only by SPS6000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_OUTPUT_MODE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Normal mode	SPS_VAL_OUTPUT_NORMAL_MODE	0
List mode	SPS_VAL_OUTPUT_LIST_MODE	1

SPS_ATTR_CC_CV_PRIORITY

Description Set channel CC/CV priority mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_CC_CV_PRIORITY** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
CC priority	SPS_VAL_CC_PRIORITY	0
CV priority	SPS_VAL_CV_PRIORITY	1

SPS_ATTR_MEASURE_AVERAGE_GRADE

Description Set measurement average mode
(Only the SPS5000X and SPS6000X series are supported).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeId is **SPS_ATTR_MEASURE_AVERAGE_GRADE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
Low grade	SPS_VAL_MEASURE_LOW_GRADE	0
Middle grade	SPS_VAL_MEASURE_MIDDLE_GRADE	1
High grade	SPS_VAL_MEASURE_HIGH_GRADE	2

SPS_ATTR_RESET_PROTECT_STATE

Description Restore the circuit protection status of the channel.

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeld is **SPS_ATTR_RESET_PROTECT_STATE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Not reset protect	SPS_VAL_NOT_RESET_PROTECT	0
Reset protect	SPS_VAL_RESET_PROTECT	1

SPS_ATTR_OCP_STATE_SET

Description	Set the overcurrent protection enable state.
Data Type	ViInt32
Access	R/W
Common Control Functions	<pre>ViStatus SPS_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attribute, ViInt32 *value); ViStatus SPS_SetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attribute, ViInt32 value);</pre>

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_OCP_STATE_SET** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
OCP disable	SPS_VAL_OCP_STATE_OFF	0
OCP enable	SPS_VAL_OCP_STATE_ON	1

SPS_ATTR_LOAD_MODE

Description Set bleeder circuit mode
(Supported only by SPS5000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_LOAD_MODE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Load disable	SPS_VAL_LOAD_MODE_OFF	0
Load enable	SPS_VAL_LOAD_MODE_ON	1

SPS_ATTR_CHANNEL_ENABLE_STATE

Description Set channel enable status
(Supported only by SPS5000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~"CH4".

attributeId is **SPS_ATTR_CHANNEL_ENABLE_STATE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
Channel disable	SPS_VAL_CHANNEL_DISABLE	0
Channel enable	SPS_VAL_CHANNEL_ENABLE	1

SPS_ATTR_OUTPUT_LOCAL_ENABLE_STATE

Description Set the channel external control enable status (Supported only by SPS5000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is

SPS_ATTR_OUTPUT_LOCAL_ENABLE_STATE macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Output local disable	SPS_VAL_OUTPUT_LOCAL_DISABLE	0
Output local enable	SPS_VAL_OUTPUT_LOCAL_ENABLE	1

SPS_ATTR_RESISTANCE_SET

Description	Set channel internal resistance (Supported only by SPS5000X series).
Data type	ViReal64
Access	R/W
Common Control Functions	<pre>ViStatus SPS_SetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 value); ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_RESISTANCE_SET macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	See user manual for range (unit: Ω).

SPS_ATTR_OCP_DELAY_SET

Description Set OCP trigger delay (Supported only by SPD4000X series).

Data type ViReal64

Access R/W

Common Control Functions

```
ViStatus SPS_SetAttributeViReal64 (ViSession vi,  
ViConstString channelName, ViAttr attribute, ViReal64 value);  
ViStatus SPS_GetAttributeViReal64 (ViSession vi,  
ViConstString channelName, ViAttr attribute, ViReal64 *value);
```

Notes:

vi is the instrument handle.

channelName: "CH1"~ "CH4"

attributeId is **SPS_ATTR_OCP_DELAY_SET** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range See user manual for range (unit: S).

SPS_ATTR_OPERATION_MODE

Description Set operating mode (Supported only by SPD4000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_OPERATION_MODE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
Independent mode	SPS_VAL_INDEPENDENT_OPERATION_MODE	0
Series mode	SPS_VAL_SERIES_OPERATION_MODE	1
Parallel mode	SPS_VAL_PARALLEL_OPERATION_MODE	2

SPS_ATTR_SENSE_MODE

Description Set working mode (Supported only by SPD4000X series).

Data Type ViInt32

Access R/W

Common Control Functions

```
ViStatus SPS_GetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 *value);
ViStatus SPS_SetAttributeViInt32 (ViSession vi,
ViConstString channelName, ViAttr attribute, ViInt32 value);
```

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeld is **SPS_ATTR_SENSE_MODE** macro.

value is used to store or set the value of function represented by **attributeld**.

Value Range

Explanation	Discrete Value	Value
2W sense mode	SPS_VAL_2W_SENSE_MODE	0
4W sense mode	SPS_VAL_4W_SENSE_MODE	1

Measure

- SPS_ATTR_VOLTAGE_MEASURE
- SPS_ATTR_CURRENT_MEASURE
- SPS_ATTR_POWER_MEASURE
- SPS_ATTR_CHANNEL_RUN_CCCV_MODE

SPS_ATTR_VOLTAGE_MEASURE

Description	Get voltage output measurement.
Data type	ViReal64
Access	R
Common Control Functions	<pre>ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre>
	<p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeld is SPS_ATTR_VOLTAGE_MEASURE macro. value is used to store or set the value of function represented by attributeld.</p>
Value Range	NULL

SPS_ATTR_CURRENT_MEASURE

Description	Get current output measurement.
Data type	ViReal64
Access	R
Common Control Functions	<pre>ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre> <p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_CURRENT_MEASURE macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	NULL

SPS_ATTR_POWER_MEASURE

Description	Get power output measurement.
Data type	ViReal64
Access	R
Common Control Functions	<pre>ViStatus SPS_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attribute, ViReal64 *value);</pre>
	<p>Notes: vi is the instrument handle. channelName: "CH1"~ "CH4" attributeId is SPS_ATTR_POWER_MEASURE macro. value is used to store or set the value of function represented by attributeId.</p>
Value Range	NULL

SPS_ATTR_CHANNEL_RUN_CCCV_MODE

Description Get channel running status.

Data Type ViInt32

Access R

Common Control Functions ViStatus SPS_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attribute, ViInt32 *value);

Notes:

vi is the instrument handle.

channelName is "CH1"~ "CH4".

attributeId is **SPS_ATTR_CHANNEL_RUN_CCCV_MODE** macro.

value is used to store or set the value of function represented by **attributeId**.

Value Range

Explanation	Discrete Value	Value
CC priority	SPS_VAL_CC_PRIORITY	0
CV priority	SPS_VAL_CV_PRIORITY	1

High Level Functions

- **SPS_InitWithOptions (ViRsrcresourceName, ViBooleanIDQuery, ViBooleanresetDevice, ViConstStringoptionString, ViSession *newVi)**

This function creates a new IVI session.

Parameter	Description
resourceName	This parameter specifies the resource name of the instrument
IDQuery	To perform ID query or not
resetDevice	To reset the device or not
optionString	This parameter is the option string set to the InitWithOptions function of the instrument driver. It includes settings for Simulate, RangeCheck, QueryInstrStatus and Cache
*newVi	Instrument handle
Example: SPS_InitWithOptions ("TCPIP0::10.11.13.218::inst0::INSTR ", VI_TRUE, VI_FALSE, "Simulate=0,RangeCheck=1,QueryInstrStatus=0,Cache=0", &session);	

- **SPS_close (ViSession vi)**

This function closes the instrument.

Parameter	Description
vi	Instrument handle
Example: SPS_close(vi);	

- **SPS_ConfigureOutputEnabled (ViSession vi, ViConstString channelName, ViBoolean enabled);**

Parameter	Description
vi	Instrument handle
channelName	"CH1"~"CH4"
enabled	Whether to enable output
Example: SPS_ConfigureOutputEnabled (vi, "CH1", VI_TRUE);	

➤ **SPS_ConfigureOutputRange (ViSession vi, ViConstString channelName, ViInt32 rangeType, ViReal64 range);**

This function is used to determine whether a value is within the settable voltage range or within the settable current range.

Parameter	Description									
vi	Instrument handle									
channelName	“CH1”~ “CH4”									
rangeType	The type of the value being judged. <table border="1" data-bbox="555 613 1347 772"> <thead> <tr> <th>Explanation</th> <th>Discrete Value</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>SPS_VAL_RANGE_CURRENT</td> <td>0</td> </tr> <tr> <td>Voltage</td> <td>SPS_VAL_RANGE_VOLTAGE</td> <td>1</td> </tr> </tbody> </table>	Explanation	Discrete Value	Value	Current	SPS_VAL_RANGE_CURRENT	0	Voltage	SPS_VAL_RANGE_VOLTAGE	1
Explanation	Discrete Value	Value								
Current	SPS_VAL_RANGE_CURRENT	0								
Voltage	SPS_VAL_RANGE_VOLTAGE	1								
range	The value being judged.									
Example: SPS_ConfigureOutputRange (vi, “CH1”, SPS_VAL_RANGE_CURRENT, 4);										

➤ **SPS_ConfigureCurrentLimit (ViSession vi, ViConstString channelName, ViInt32 behavior, ViReal64 limit);**

This function is used to set the current value.

Parameter	Description
vi	Instrument handle
channelName	“CH1”~ “CH4”
behavior	Current behavior pattern during overcurrent conditions. This value is SPS_VAL_CURRENT_LIMIT_BEHAVIOR_DISABLE_OUTPUT (0) and cannot be modified.
limit	Set current value.
Example: SPS_ConfigureCurrentLimit (vi, “CH1”, SPS_VAL_CURRENT_LIMIT_BEHAVIOR_DISABLE_OUTPUT, 5);	

➤ **SPS_ConfigureOVP (ViSession vi, ViConstString channelName, ViBoolean enabled, ViReal64 limit);**

This function is used to set the overvoltage protection value.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"
enabled	Overvoltage protection enable value. When this value is set to VI_TRUE, the overvoltage protection can be set normally. When set to VI_FALSE, the overvoltage protection value cannot be modified. No matter what value is set, overvoltage protection is always on.
limit	overvoltage protection value
Example: SPS_ConfigureOVP (vi, "CH1", VI_TRUE, 8);	

- **SPS_ConfigureVoltageLevel (ViSession vi, ViConstString channelName, ViReal64 level);**

This function is used to set the voltage value of the channel.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"
level	Set voltage value
Example: SPS_ConfigureVoltageLevel (vi,"CH1",8);	

- **SPS_QueryOutputState (ViSession vi, ViConstString channelName, ViInt32 outputState, ViBoolean* inState);**

This function is used to query the channel output status.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"

outputState	Need to query the type of output status		
	Explanation	Discrete Value	Value
	Constant Voltage (CV)	SPS_VAL_OUTPUT_CONSTANT_VOLTAGE	0
	Constant Current (CC)	SPS_VAL_OUTPUT_CONSTANT_CURRENT	1
	overvoltage	SPS_VAL_OUTPUT_OVER_VOLTAGE	2
	overcurrent	SPS_VAL_OUTPUT_OVER_CURRENT	3
inState	Provide a return value, VI_TRUE indicates that it is in this state VI_FALSE means not in this state.		
Example: SPS_QueryOutputState (vi, "CH1", SPS_VAL_OUTPUT_CONSTANT_VOLTAGE, &valueboolean);			

➤ **SPS_QueryMaxCurrentLimit (ViSession vi, ViConstString channelName, ViReal64 voltageLevel, ViReal64* maxCurrentLimit);**

This function is used to obtain the maximum current value that the channel can set at a specific voltage.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"
voltageLevel	set voltage value
maxCurrentLimit	Returns the set maximum current value.
Example: SPS_QueryMaxCurrentLimit (vi, "CH1", voltagelevel, &maxCurrentLimit);	

➤ **SPS_QueryMaxVoltageLevel (ViSession vi, ViConstString channelName, ViReal64 currentLimit, ViReal64* maxVoltageLevel);**

This function is used to obtain the maximum voltage value that the channel can set at a specific current value.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"
currentLimit	set current value

maxVoltageLevel	Returns the maximum voltage value that can be set.
Example: SPS_QueryMaxVoltageLevel (vi, "CH1", currentlimit, &maxVoltageLevel);	

➤ **SPS_ResetOutputProtection (ViSession vi, ViConstString channelName);**

This function is used to restore the circuit protection status of the specified channel.

Parameter	Description
vi	Instrument handle
channelName	"CH1"~ "CH4"
Example: SPS_ResetOutputProtection (session, "CH1");	

➤ **SPS_Measure (ViSession vi, ViConstString channelName, ViInt32 measurementType, ViReal64* measurement);**

This function is used to measure output voltage or current.

Parameter	Description									
vi	Instrument handle									
channelName	"CH1"~ "CH4"									
measurementType	Type of measurement required. <table border="1" data-bbox="550 1288 1300 1444"> <thead> <tr> <th>Explanation</th> <th>Discrete Value</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Measure Current</td> <td>SPS_VAL_MEASURE_CURRENT</td> <td>0</td> </tr> <tr> <td>Measure Voltage</td> <td>SPS_VAL_MEASURE_VOLTAGE</td> <td>1</td> </tr> </tbody> </table>	Explanation	Discrete Value	Value	Measure Current	SPS_VAL_MEASURE_CURRENT	0	Measure Voltage	SPS_VAL_MEASURE_VOLTAGE	1
Explanation	Discrete Value	Value								
Measure Current	SPS_VAL_MEASURE_CURRENT	0								
Measure Voltage	SPS_VAL_MEASURE_VOLTAGE	1								
measurement	Returns a measurement of the specified type.									
Example: SPS_Measure (vi, "CH1", SPS_VAL_MEASURE_VOLTAGE, &value);										

➤ **SPS_GetChannelName (ViSession vi, ViInt32 index, ViInt32 bufferSize, ViChar name[])**

This function returns the highest-level channel name that corresponds to the specific driver channel string that is in the channel table at an index you specify.

Parameter	Description
vi	Instrument handle
index	Specified index
bufferSize	The length of the channel name
name	Channel name storage location
Example: SPS_GetChannelName (vi, 1, 256, str);	

Programming Example

The example is running in an environment where NI-VISA 20.0, LabWindow/CVI 2017, and IVI Compliance Package 20.0 are installed.

Using dynamic link library

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <Windows.h>
#include "SPS.h"

#define USBTMC_RESOURCE
"USB0::0xF4EC::0x1460::SPS20221013::INSTR"

#pragma comment(lib,"SPS_x32.lib")

static ViSession session;

void check_error_state(ViStatus state, ViReal64 value, const char * tip)
{
    char buf[100] = {0};
    char buf2[100] = {0};
    strcpy(buf, tip);
    if (state != VI_SUCCESS)
    {
        sprintf(buf2, " failure, error code: %x \n", state);
    }
    else
    {
        sprintf(buf2, " success, value = %f \n", value);
    }
    strcat(buf, buf2);
    printf(buf);
}

void basic_operation_test_case(void)
{
```

```
ViString channel = "CH1";
ViStatus status = VI_SUCCESS;
ViReal64 real64_value = 0.0;
ViBoolean bool_state = VI_FALSE;
ViReal64 volt = 5.0f;
ViReal64 curr = 1.0f;
ViBoolean output_state = VI_FALSE; //VI_FALSE VI_TRUE

status = SPS_SetAttributeViReal64(session, channel,
SPS_ATTR_VOLTAGE_LEVEL, volt);
check_error_state(status, volt, "Set SPS_ATTR_VOLTAGE_LEVEL");
status = SPS_GetAttributeViReal64(session, channel,
SPS_ATTR_VOLTAGE_LEVEL, &real64_value);
check_error_state(status, real64_value, "Get SPS_ATTR_VOLTAGE_LEVEL");

status = SPS_SetAttributeViReal64(session, channel, SPS_ATTR_CURRENT_SET,
curr);
check_error_state(status, curr, "Set SPS_ATTR_CURRENT_SET");
status = SPS_GetAttributeViReal64(session, channel, SPS_ATTR_CURRENT_SET,
&real64_value);
check_error_state(status, real64_value, "Set SPS_ATTR_CURRENT_SET");

status = SPS_SetAttributeViBoolean(session, channel,
SPS_ATTR_OUTPUT_ENABLED, output_state);
check_error_state(status, output_state, "Set SPS_ATTR_OUTPUT_ENABLED");
status = SPS_GetAttributeViBoolean(session, channel,
SPS_ATTR_OUTPUT_ENABLED, &bool_state);
check_error_state(status, bool_state, "Set SPS_ATTR_OUTPUT_ENABLED");

printf("Basic operation test case execution completed!\n");
}

void main(void)
{
ViString channel = "CH1";
ViStatus status = VI_SUCCESS; //VI_FALSE
ViChar buff[100] = {0};
status = SPS_InitWithOptions(USBTMC_RESOURCE, VI_TRUE, VI_TRUE,
```

```
"Simulate=0,RangeCheck=1,QueryInstrStatus=0,Cache=0", &session);
    if (status != VI_SUCCESS)
    {
        printf("InitWithOptions failure, error code: %x \n", status);
    }

    status = SPS_GetAttributeViString(session, channel,
    SPS_ATTR_ID_QUERY_RESPONSE, sizeof(buff), buff);
    printf(buff);

    basic_operation_test_case();

    printf("IVI driver test completed!\n");
    system("cmd /C pause");
}
```



About SIGLENT

SIGLENT is an international high-tech company, concentrating on R&D, sales, production and services of electronic test & measurement instruments.

SIGLENT first began developing digital oscilloscopes independently in 2002. After more than a decade of continuous development, SIGLENT has extended its product line to include digital oscilloscopes, isolated handheld oscilloscopes, function/arbitrary waveform generators, RF/MW signal generators, spectrum analyzers, vector network analyzers, digital multimeters, DC power supplies, electronic loads and other general purpose test instrumentation. Since its first oscilloscope was launched in 2005, SIGLENT has become the fastest growing manufacturer of digital oscilloscopes. We firmly believe that today SIGLENT is the best value in electronic test & measurement.

Headquarters:

SIGLENT Technologies Co., Ltd
Add: Bldg No.4 & No.5, Antongda Industrial
Zone, 3rd Liuxian Road, Bao'an District,
Shenzhen, 518101, China
Tel: + 86 755 3688 7876
Fax: + 86 755 3359 1582
Email: sales@siglent.com
Website: int.siglent.com

North America:

SIGLENT Technologies America, Inc
6557 Cochran Rd Solon, Ohio 44139
Tel: 440-398-5800
Toll Free: 877-515-5551
Fax: 440-399-1211
Email: info@siglentna.com
Website: www.siglentna.com

Europe:

SIGLENT Technologies Germany GmbH
Add: Staetzlinger Str. 70
86165 Augsburg, Germany
Tel: +49(0)-821-666 0 111 0
Fax: +49(0)-821-666 0 111 22
Email: info-eu@siglent.com
Website: www.siglenteu.com

Follow us on
Facebook: SiglentTech

