

# Function Generator

## IVI-C Programming Guide

E02A  
Jan, 2024



## 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 Function Generator supporting this IVI-C driver is shown below.

| Series       | Release Version Supporting IVI-C Driver |
|--------------|---|
| SDG1000X     | 1.01.01.33R1B5 and higher               |
| SDG1000XPLUS | V1P.1.1.1.41R2 and higher               |
| SDG2000X     | 2.01.01.23R5 and higher                 |
| SDG6000X     | 6.01.01.33R2 and higher                 |
| SDG7000A     | 1.1.1.26 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.

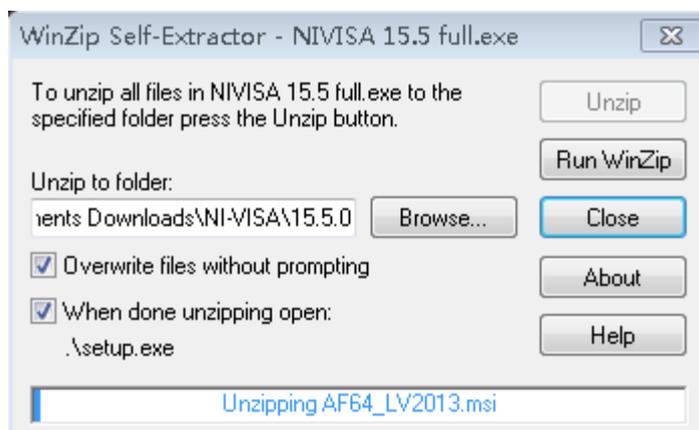
## Install NI-MAX

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>.

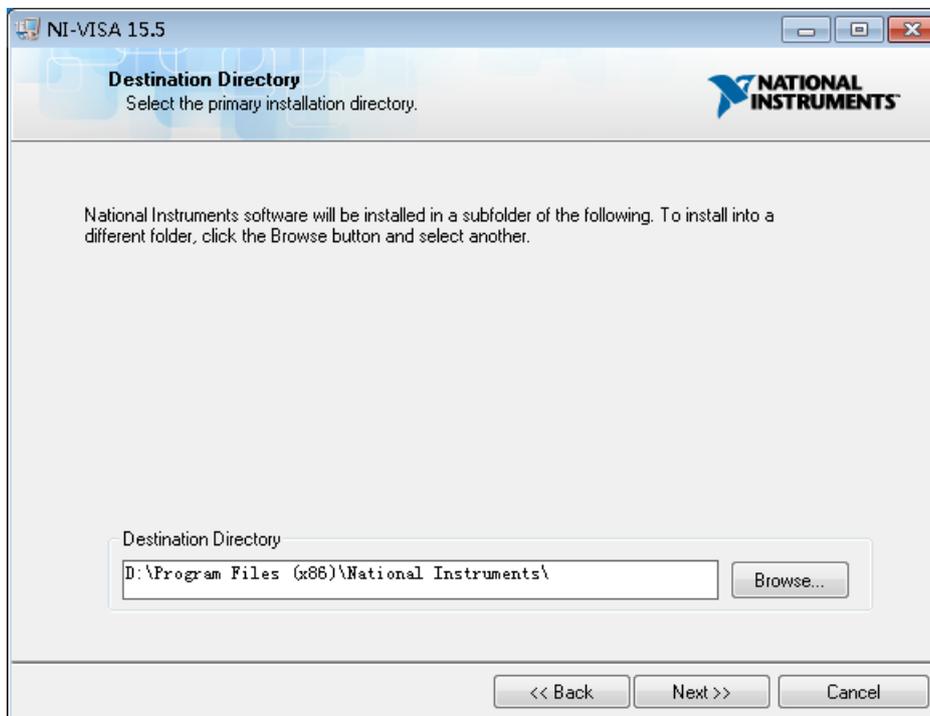
- a. Double click the NIVISA 15.5 full.exe, a dialog will be shown as below:



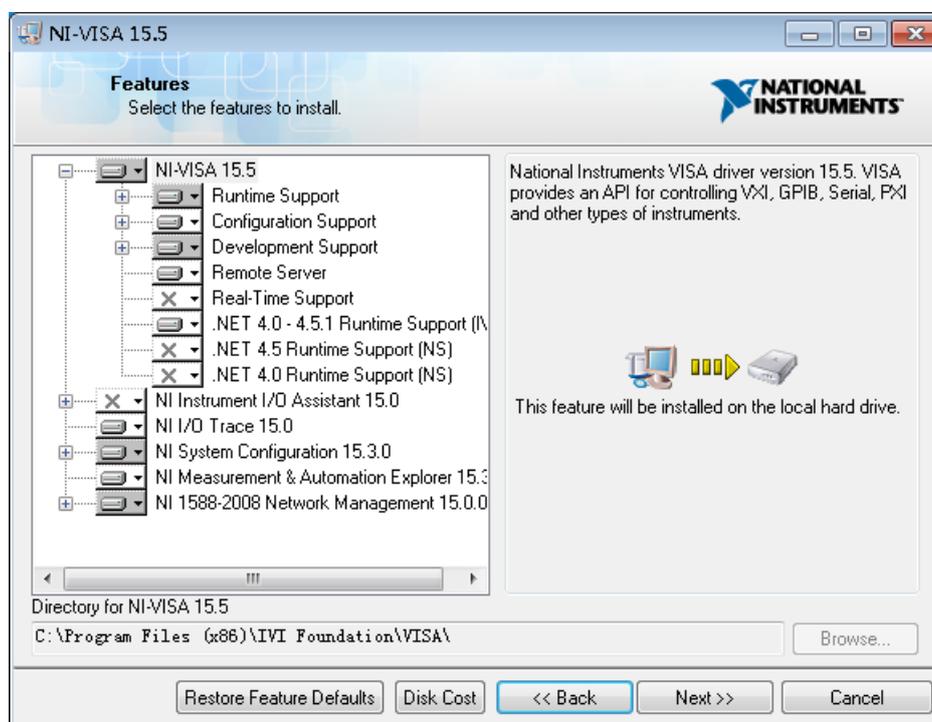
- b. Click Unzip, the installation process will automatically launch after unzipping files. If your computer needs to install .NET Framework 4, it may auto start.



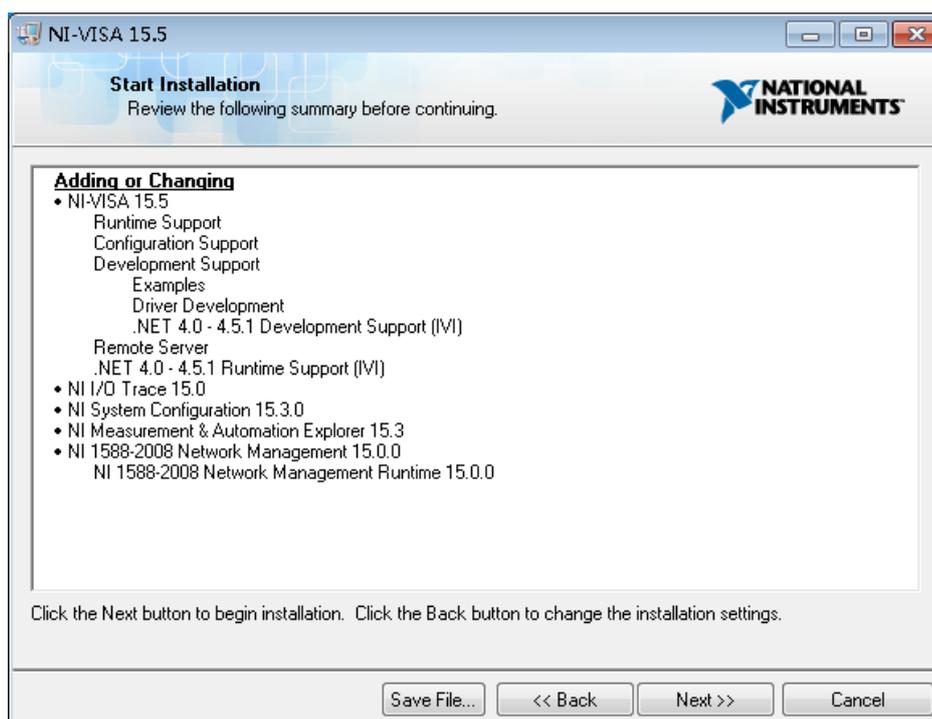
- c. The NI-VISA installing dialog is shown above. Click Next to start the installation process.



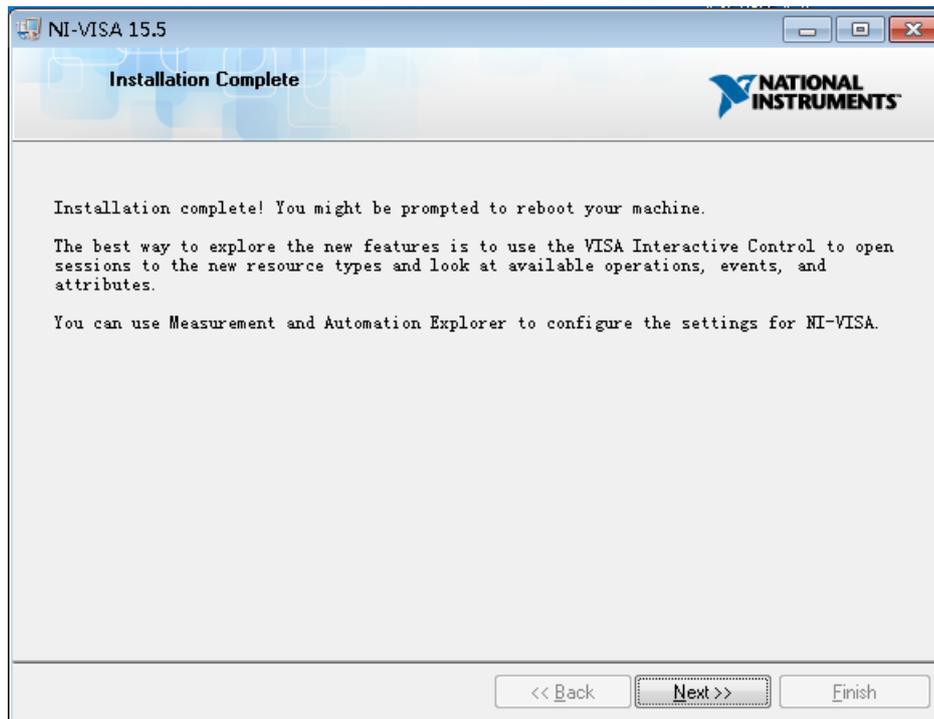
- d. Set the install path. The default path is “C:\Program Files\National Instruments\”. You can change it. Click Next.



- 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.

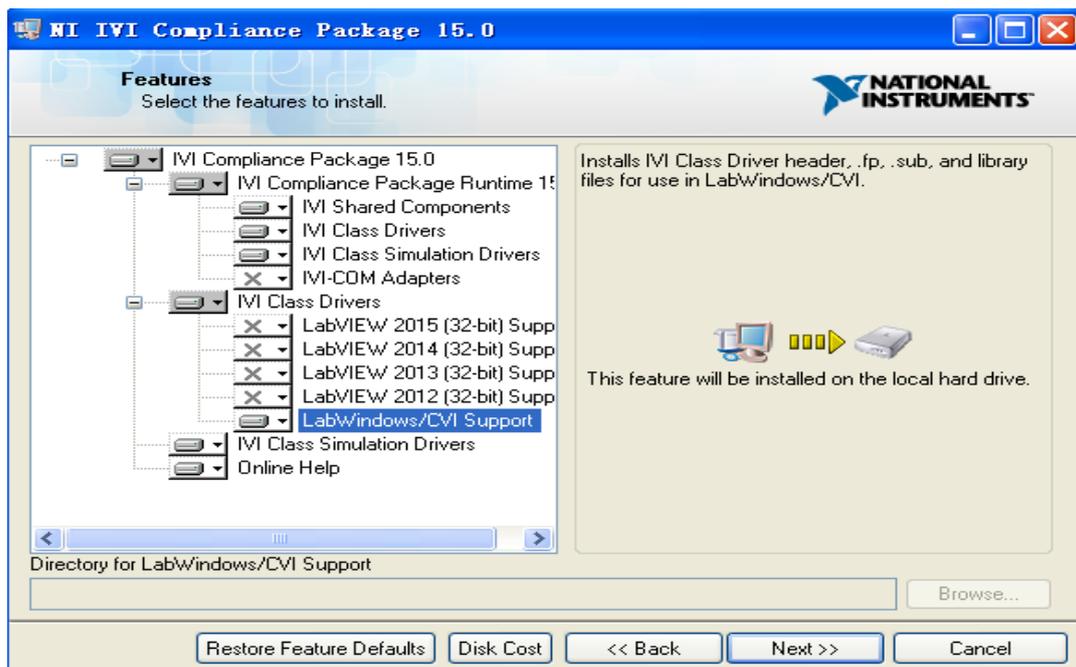
## Install the IVI Compliance Package

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

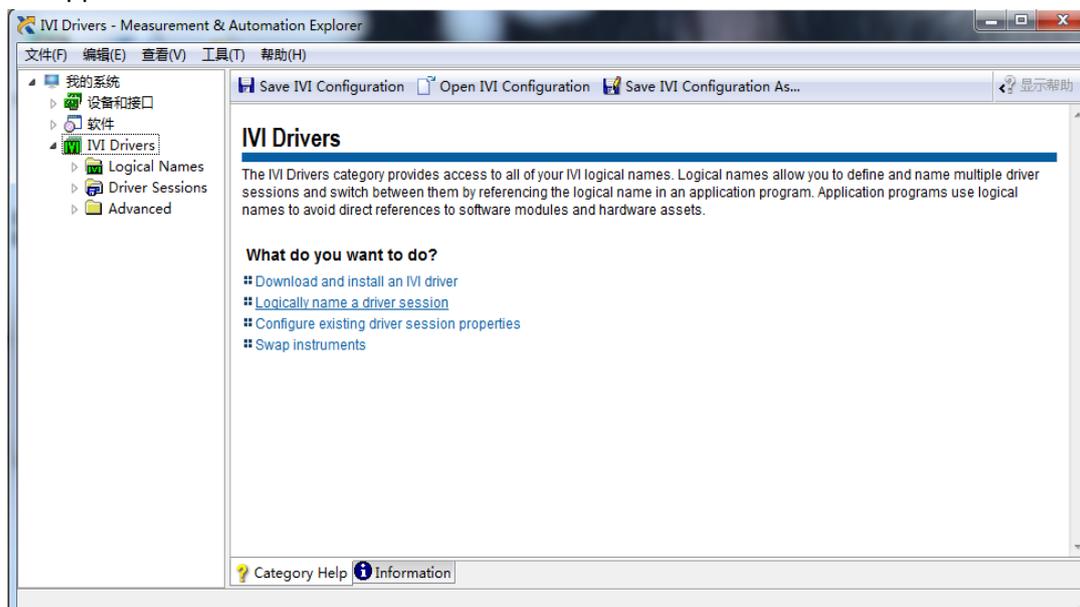
You can get the IVI Compliance Package from

<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.



## SDG IVI-C Driver Package List

The SDG IVI-C driver package provides three kinds of files: sdg.dll file, sdg.h file and sdg.lib file.

| File               | Description   |
|--------------------|---|
| sdg.dll/sdg_64.dll | A dynamic link library file, including variables, functions, and data interfaces for various attributes.                              |
| sdg.lib/sdg_64.lib | An import library file, including the symbolic name and optional identification number of each exported function in the sdg.dll file. |
| sdg.h              | A header file, including declarations of variables, functions, and data interfaces.   |

You include the sdg.h when programming the Siglent oscilloscope with the IVI driver, and load the sdg.dll dynamic file or sdg.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         |

## Access IVI Attribute

User-callable functions are typically implemented by manipulating attributes. You can call `sdg_SetAttribute` or `sdg_GetAttribute` functions.

### SetAttribute Function Group

- `sdg_SetAttributeViInt32` (`ViSession vi`, `ViConstString channelName`, `ViAttr attributId`, `ViInt32 value`)

Example: When you want to set the burst count, you can call the `SetAttribute` function to change the burst count.

```
sdg_SetAttributeViInt32 (vi, channelName, SDG_ATTR_OPERATION_MODE,  
SDG_VAL_OPERATE_BURST);
```

where,

**vi:** The instrument handle.

**channelName:** A constant string that represents the `channelName` and shows that this `SDG_ATTR_OPERATION_MODE` attribute is corresponding to that specific mode.

**SDG\_VAL\_OPERATE\_BURST:** Set the value to mode.

- `sdg_SetAttributeViReal64` (`ViSession vi`, `ViConstString channelName`, `ViAttr attributId`, `ViReal64 value`)

Example: When you want to set the rate at which the function generator's internal trigger source generates trigger signals, you can call `SetAttribute` or `GetAttribute` function to change or obtain the rate.

```
sdg_SetAttributeViReal64(vi, VI_NULL, SDG_ATTR_INTERNAL_TRIGGER_RATE,  
rate);
```

where,

**vi**: Instrument Handle.

**VI\_NULL**: A constant string that represents the rate at which the function generator's internal trigger source generates trigger signals and shows that this **SDG\_ATTR\_INTERNAL\_TRIGGER\_RATE** attribute is corresponding to this specific rate.

**rate**: Set the value to rate.

- `sdg_SetAttributeViString (ViSession vi, ViConstString channelName, ViAttr attributeld, ViConstString value)`

Example: When you want to get the string that describes the physical instrument ; from the configuration file, you can call SetAttribute function to get the string that describes the physical instrument.

```
sdg_SetAttributeViString(vi,"",IVI_ATTR_IO_RESOURCE_DESCRIPTOR,0,  
newResourceName);
```

where,

**vi**: The instrument handle.

**“CHAN1”**: A constant string that represents the analog channel 1 and shows that this **SDG\_ATTR\_CHANNEL\_LABLE\_TEXT** attribute is corresponding to this specific channel.

**newResourceName**: Get the string that describes the physical instrument.

- `sdg_SetAttributeViBoolean (ViSession vi, ViConstString channelName, ViAttr`

attributeld, ViBoolean value)

Example: When you want to set a channel on or off, you can call SetAttribute or GetAttribute function to change or obtain the state of the channel.

```
sdg_SetAttributeViBoolean(session,"CHAN1",SDG_ATTR_CHANNEL_ENABLED,  
VI_FALSE);
```

where,

**session:** The instrument handle.

**"CHAN1":** A constant string that represents the analog channel 1 and shows that this **SDG\_ATTR\_CHANNEL\_ENABLED** attribute is corresponding to this channel.

**VI\_FALSE:** This means turning channel 1 off.

### GetAttribute Function Group

- `sdg_GetAttributeViReal64 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViReal64 *value)`

Example: When you want to set the probe attenuation, you can call SetAttribute or GetAttribute function to change or obtain the probe attenuation value.

```
sdg_GetAttributeViReal64(session,"CHAN1",SDG_ATTR_PROBE_ATTENUATION,  
&value64);
```

where,

**session:** The instrument handle.

**"CHAN1":** A constant string that represents the analog channel 1 and shows that this

**SDG\_ATTR\_PROBE\_ATTENUATION** attribute is corresponding to this channel.

**value64:** A ViReal64 type variable which is used to store the returned value of the probe attenuation query.

- `sdg_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 *value)`

Example: When you want to set the channel coupling, you can call `SetAttribute` or `GetAttribute` function to change or obtain the channel coupling.

```
sdg_GetAttributeViInt32(session,"CHAN1",SDG_ATTR_VERTICAL_COUPLING,  
&value32);
```

where,

**session:** The instrument handle.

**"CHAN1":** A constant string that represents the analog channel 1 and shows that this **SDG\_ATTR\_VERTICAL\_COUPLING** attribute is corresponding to this specific channel.

**value32:** A ViInt32 type variable which is used to store the returned value of the coupling query.

- `sdg_GetAttributeViBoolean (ViSession vi, ViConstString channelName, ViAttr attributeld, ViBoolean *value)`

Example: When you want to set a channel on or off, you can call `SetAttribute` or `GetAttribute` function to change or obtain the state of channel.

```
sdg_GetAttributeViBoolean(session,"CHAN1",SDG_ATTR_CHANNEL_ENABLED,  
&boolean);
```

where,

**session:** The instrument handle.

**“CHAN1”:** A constant string that represents the analog channel 1 and shows that this **SDG\_ATTR\_CHANNEL\_ENABLED** attribute is corresponding to this specific channel.

**boolean:** A ViBoolean type variable which is used to store the returned value.

- `sdg_GetAttributeViString (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 bufSize, ViChar value[])`

Example: When you want to set the channel label text, you can call `SetAttribute` or `GetAttribute` function to change or obtain the channel label text.

```
sdg_GetAttributeViString(session,"CHAN1",SDG_ATTR_CHANNEL_LABEL_TE  
XT, buffersize,str);
```

where,

**session:** The instrument handle.

**“CHAN1”:** A constant string that represents the analog channel 1 and shows that this **SDG\_ATTR\_CHANNEL\_LABEL\_TEXT** attribute is corresponding to this specific channel.

**buffersize:** A ViInt32 type variable.

**str:** A ViString type variable which is used to store the returned value.

## 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           | SDG_ATTR_MACHINE_MODEL            |
|                           | SDG_ATTR_OPERATION_MODE           |
|                           | SDG_ATTR_OUTPUT_COUNT             |
|                           | SDG_ATTR_REF_CLOCK_SOURCE         |
|                           | SDG_ATTR_OUTPUT_ENABLED           |
|                           | SDG_ATTR_OUTPUT_IMPEDANCE         |
| Standard Function Output  | SDG_ATTR_FUNC_WAVEFORM            |
|                           | SDG_ATTR_FUNC_AMPLITUDE           |
|                           | SDG_ATTR_FUNC_DC_OFFSET           |
|                           | SDG_ATTR_FUNC_FREQUENCY           |
|                           | SDG_ATTR_FUNC_START_PHASE         |
|                           | SDG_ATTR_FUNC_DUTY_CYCLE_HIGH     |
|                           | SDG_ATTR_FUNC_RAMP_SYMMETRY       |
| Arbitrary Waveform Output | SDG_ATTR_ARB_GAIN                 |
|                           | SDG_ATTR_ARB_OFFSET               |
|                           | SDG_ATTR_ARB_SAMPLE_RATE          |
|                           | SDG_ATTR_ARB_WAVEFORM_HANDLE      |
|                           | SDG_ATTR_MAX_NUM_WAVEFORMS        |
|                           | SDG_ATTR_MAX_WAVEFORM_SIZE        |
|                           | SDG_ATTR_MIN_WAVEFORM_SIZE        |
|                           | SDG_ATTR_WAVEFORM_QUANTUM         |
|                           | SDG_ATTR_ARB_FREQUENCY            |
| Harmonic                  | SDG_ATTR_HARMONIC_ENABLED         |
|                           | SDG_ATTR_HARMONIC_ORDER           |
|                           | SDG_ATTR_HARMONIC_SN              |
|                           | SDG_ATTR_HARMONIC_AMPL            |
|                           | SDG_ATTR_HARMONIC_PHASE           |
|                           | SDG_ATTR_HARMONIC_TYPE            |
|                           | SDG_ATTR_COUNTER_STATISTICS_STATE |
|                           | SDG_ATTR_COUNTER_STATE            |
|                           | SDG_ATTR_COUNTER_HF               |
|                           | SDG_ATTR_COUNTER_COUPING          |
|                           | SDG_ATTR_COUNTER_IMPEDANCE        |
|                           | SDG_ATTR_COUNTER_LEVE             |
| Triggering                | SDG_ATTR_TRIGGER_SOURCE           |
|                           | SDG_ATTR_INTERNAL_TRIGGER_RATE    |
|                           | SDG_ATTR_BURST_NCYCLES_ENABLED    |

|            |                                       |
|------------|---------------------------------------|
| Bursting   | SDG_ATTR_BURST_GATED_ENABLED          |
|            | SDG_ATTR_BURST_INFINITY_ENABLED       |
|            | SDG_ATTR_BURST_GATE_POLARITY          |
|            | SDG_ATTR_BURST_INTERNAL_PERIOD        |
|            | SDG_ATTR_BURST_PHASE                  |
|            | SDG_ATTR_BURST_TDELAY                 |
|            | SDG_ATTR_BURST_TRIGGER_SLOPE          |
|            | SDG_ATTR_BURST_TRIGGER_SOURCE         |
|            | SDG_ATTR_BURST_TRIGGER_TRIGOUT        |
|            | SDG_ATTR_BURST_COUNT                  |
| Sweep      | SDG_ATTR_SWEEP_STATE                  |
|            | SDG_ATTR_SWEEP_TIME                   |
|            | SDG_ATTR_SWEEP_TRIGGER_SLOPE          |
|            | SDG_ATTR_SWEEP_TRIGGER_SOURCE         |
|            | SDG_ATTR_SWEEP_TRIGGER_TRIGOUT        |
|            | SDG_ATTR_SWEEP_TYPE                   |
|            | SDG_ATTR_SWEEP_DIRECTION              |
|            | SDG_ATTR_SWEEP_FREQUENCY_CENTER       |
|            | SDG_ATTR_SWEEP_FREQUENCY_SPAN         |
|            | SDG_ATTR_SWEEP_FREQUENCY_START        |
|            | SDG_ATTR_SWEEP_FREQUENCY_STOP         |
|            | SDG_ATTR_SWEEP_FREQUENCY_START_RECORD |
|            | SDG_ATTR_SWEEP_FREQUENCY_STOP_RECORD  |
| IQ         | SDG_ATTR_IQ_FREQUENCY_CENTER          |
|            | SDG_ATTR_IQ_SAMPLE_RATE               |
|            | SDG_ATTR_IQ_SYMBOL_RATE               |
|            | SDG_ATTR_IQ_AMPLITUDE                 |
|            | SDG_ATTR_IQ_TRIGGER_SOURCE            |
|            | SDG_ATTR_IQ_ADJ_GAIN                  |
|            | SDG_ATTR_IQ_ADJ_I_OFFSET              |
|            | SDG_ATTR_IQ_ADJ_Q_OFFSET              |
|            | SDG_ATTR_IQ_ADJ_Q_SKEW                |
|            | SDG_ATTR_IQ_WAVE_BUILT_IN             |
|            | SDG_ATTR_IQ_WAVE_USER                 |
| PRBS       | SDG_ATTR_PRBS_BITRATE                 |
|            | SDG_ATTR_PRBS_AMPLITUDE               |
|            | SDG_ATTR_PRBS_OFFSET                  |
|            | SDG_ATTR_PRBS_LENGTH                  |
|            | SDG_ATTR_PRBS_EDGE                    |
|            | SDG_ATTR_PRBS_DIFFSTATE               |
|            | SDG_ATTR_PRBS_LOGIC_STATE             |
|            |                                       |
| Modulation | SDG_ATTR_MOD_TYPE                     |
|            | SDG_ATTR_MOD_STATE                    |
|            | SDG_ATTR_AM_ENABLED                   |

|                                 |
|---------------------------------|
| SDG_ATTR_AM_SOURCE              |
| SDG_ATTR_AM_INTERNAL_DEPTH      |
| SDG_ATTR_AM_INTERNAL_WAVEFORM   |
| SDG_ATTR_AM_INTERNAL_FREQUENCY  |
| SDG_ATTR_FM_ENABLED             |
| SDG_ATTR_FM_SOURCE              |
| SDG_ATTR_FM_INTERNAL_DEVIATION  |
| SDG_ATTR_FM_INTERNAL_WAVEFORM   |
| SDG_ATTR_FM_INTERNAL_FREQUENCY  |
| SDG_ATTR_PM_ENABLED             |
| SDG_ATTR_PM_SOURCE              |
| SDG_ATTR_PM_DEVIATION           |
| SDG_ATTR_PM_INTERNAL_WAVEFORM   |
| SDG_ATTR_PM_INTERNAL_FREQUENCY  |
| SDG_ATTR_ASK_ENABLED            |
| SDG_ATTR_ASK_SOURCE             |
| SDG_ATTR_ASK_INTERNAL_RATE      |
| SDG_ATTR_ASK_AMPLITUDE          |
| SDG_ATTR_FSK_ENABLED            |
| SDG_ATTR_FSK_SOURCE             |
| SDG_ATTR_FSK_FREQUENCY          |
| SDG_ATTR_FSK_INTERNAL_RATE      |
| SDG_ATTR_PSK_ENABLED            |
| SDG_ATTR_PSK_SOURCE             |
| SDG_ATTR_PSK_POLARITY           |
| SDG_ATTR_PSK_PHASE              |
| SDG_ATTR_PSK_INTERNAL_RATE      |
| SDG_ATTR_PWM_ENABLED            |
| SDG_ATTR_PWM_SOURCE             |
| SDG_ATTR_PWM_DEVIATION_WIDTH    |
| SDG_ATTR_PWM_INTERNAL_WAVEFORM  |
| SDG_ATTR_PWM_INTERNAL_FREQUENCY |
| SDG_ATTR_PULSE_WIDTH            |
| SDG_ATTR_PULSE_PERIOD           |

## Basic Operation

Attributes that control the basic features of the function generator. The basic group has the following attributes:

- ◆ **SDG\_ATTR\_MACHINE\_MODEL**
- ◆ **SDG\_ATTR\_OPERATION\_MODE**
- ◆ **SDG\_ATTR\_OUTPUT\_COUNT**
- ◆ **SDG\_ATTR\_REF\_CLOCK\_SOURCET**
- ◆ **SDG\_ATTR\_OUTPUT\_ENABLED**
- ◆ **SDG\_ATTR\_OUTPUT\_IMPEDANCE**

## SDG\_ATTR\_MACHINE\_MODEL

**Description** This attribute is used to store the specific machine model.

**Data type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViReal64` (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 value)

`sdg_GetAttributeViReal64` (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 \*value)

**Notes:**

**vi** is the instrument handle.

**channelName** is one of the following analog inputs:

SDG1012X,SDG1032X,SDG1062X,SDG2042X,  
SDG2082X,SDG6022X,SDG6032X,SDG6052X,  
SDG6052X,SDG7102A,SDG7052A,SDG7032A.

**attributeld** is SDG\_ATTR\_MACHINE\_MODEL macro.

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

**Value Range**

| Type     | Discrete Value                 | Value |
|----------|--------------------------------|-------|
| SDG1012X | SDG_VAL_MACHINE_MODEL_SDG1012X | 0     |
| SDG1032X | SDG_VAL_MACHINE_MODEL_SDG1032X | 1     |
| SDG1062X | SDG_VAL_MACHINE_MODEL_SDG1062X | 2     |
| SDG2042X | SDG_VAL_MACHINE_MODEL_SDG2042X | 3     |
| SDG2082X | SDG_VAL_MACHINE_MODEL_SDG2082X | 4     |
| SDG2122X | SDG_VAL_MACHINE_MODEL_SDG2122X | 5     |
| SDG6022X | SDG_VAL_MACHINE_MODEL_SDG6022X | 6     |
| SDG6032X | SDG_VAL_MACHINE_MODEL_SDG6032X | 7     |
| SDG6052X | SDG_VAL_MACHINE_MODEL_SDG6052X | 8     |
| SDG7102A | SDG_VAL_MACHINE_MODEL_SDG7102A | 9     |
| SDG7052A | SDG_VAL_MACHINE_MODEL_SDG7052A | 10    |
| SDG7032A | SDG_VAL_MACHINE_MODEL_SDG7032A | 11    |

**Related Attribute** NULL

**High Level Functions** NULL

## SDG\_ATTR\_OPERATION\_MODE

**Description** This channel-based attribute determines how the function generator produces waveforms.

**Data type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**channelName** is one of the following analog inputs: CHAN1, CHAN2.

**attributeld** is SDG\_ATTR\_OPERATION\_MODE macro.

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

**Value Range**

| Type  | Discrete Value             | Value |
|-------|----------------------------|-------|
| CONT  | SDG_VAL_OPERATE_CONTINUOUS | 0     |
| BURST | SDG_VAL_OPERATE_BURST      | 1     |

**Related Attribute**

SDG\_ATTR\_OUTPUT\_IMPEDANCE  
SDG\_ATTR\_OUTPUT\_ENABLED  
SDG\_ATTR\_REF\_CLOCK\_SOURCET  
SDG\_ATTR\_OUTPUT\_COUNT

**High Level Functions**

`sdg_ConfigureOperationMode`  
`sdg_ConfigureBurstCount`  
`sdg_ConfigureBurstStartPhase`

## SDG\_ATTR\_OUTPUT\_COUNT

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Returns the number of available output channels.  |
| <b>Data type</b>                | ViInt32   |
| <b>Access</b>                   | R   |
| <b>Common Control Functions</b> | <code>sdg_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>channelName</b> is one of the following analog inputs: CHAN1, CHAN2.<br><b>attributeld</b> is SDG_ATTR_OUTPUT_COUNT macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | NULL.   |
| <b>Related Attribute</b>        | SDG_ATTR_OUTPUT_IMPEDANCE<br>SDG_ATTR_OUTPUT_ENABLED<br>SDG_ATTR_REF_CLOCK_SOURCET<br>SDG_ATTR_OPERATION_MODE   |
| <b>High Level Functions</b>     | NULL  |

## SDG\_ATTR\_REF\_CLOCK\_SOURCE

**Description** Specifies the reference clock source. The function generator derives frequencies and sample rates that it uses to generate waveforms from the source you specify.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32 (ViSession vi, ViConstString channelName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is NULL.

**attributeld** is SDG\_ATTR\_REF\_CLOCK\_SOURCE macro.

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

**Value Range**

| Type           | Discrete Value             | Value |
|----------------|----------------------------|-------|
| INT            | SDG_VAL_REF_CLOCK_INTERNAL | 0     |
| EXT            | SDG_VAL_REF_CLOCK_EXTERNAL | 1     |
| INT,10MOUT,ON  | SDG_VAL_REF_CLOCK_INTERNAL | 2     |
| EXT,10MOUT,ON  | SDG_VAL_REF_CLOCK_EXTERNAL | 3     |
| INT,10MOUT,OFF | SDG_VAL_REF_CLOCK_INTERNAL | 4     |
| EXT,10MOUT,OFF | SDG_VAL_REF_CLOCK_EXTERNAL | 5     |

**Notes:**

When the SDG\_ATTR\_MACHINE\_MODEL is SDG7000A&SDG6000X, all types are satisfied, otherwise only "INT" and "EXT" are included

**Related Attribute**

SDG\_ATTR\_OUTPUT\_IMPEDANCE  
SDG\_ATTR\_OPERATION\_MODE  
SDG\_ATTR\_OUTPUT\_ENABLED  
SDG\_ATTR\_OUTPUT\_COUNT

**High Level Functions** `sdg_ConfigureChannel`

## SDG\_ATTR\_OUTPUT\_ENABLED

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies whether the signal the function generator produces appears at the output connector.   |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_OUTPUT_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means Output is on<br>VI_FALSE means Output is off   |
| <b>Related Attribute</b>        | SDG_ATTR_OUTPUT_IMPEDANCE<br>SDG_ATTR_OPERATION_MODE<br>SDG_ATTR_REF_CLOCK_SOURCET<br>SDG_ATTR_OUTPUT_COUNT  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureOutputEnabled</code>  |

## SDG\_ATTR\_OUTPUT\_IMPEDANCE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the function generator's output impedance at the output connector.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_OUTPUT_IMPEDANCE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | Its value must be between 50 and 1000000. 1000000 indicates that the function generator is connected to a high impedance load.   |
| <b>Related Attribute</b>        | SDG_ATTR_OUTPUT_ENABLED<br>SDG_ATTR_OPERATION_MODE<br>SDG_ATTR_REF_CLOCK_SOURCET<br>SDG_ATTR_OUTPUT_COUNT  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureOutputImpedance</code>  |

## Standard Function Output

Attributes for generating standard function waveform output. You use this group of attributes when you set the `SDG_ATTR_OUTPUT_MODE` attribute to `SDG_VAL_OUTPUT_FUNC`. You can use the `sdg_ConfigureStandardWaveform` function to set all these attributes except `SDG_ATTR_FUNC_DUTY_CYCLE_HIGH`. The Standard Function group has the following attributes:

- ◆ **SDG\_ATTR\_FUNC\_WAVEFORM**
- ◆ **SDG\_ATTR\_FUNC\_AMPLITUDE**
- ◆ **SDG\_ATTR\_FUNC\_DC\_OFFSET**
- ◆ **SDG\_ATTR\_FUNC\_FREQUENCY**
- ◆ **SDG\_ATTR\_FUNC\_START\_PHASE**
- ◆ **SDG\_ATTR\_FUNC\_DUTY\_CYCLE\_HIGH**
- ◆ **SDG\_ATTR\_FUNC\_RAMP\_SYMMETRY**

## SDG\_ATTR\_FUNC\_WAVEFORM

**Description** This channel-based attribute specifies which standard waveform the function generator produces.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_FUNC\_WAVEFORM macro.

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

**Value Range**

| Type   | Discrete Value        | Value |
|--------|-----------------------|-------|
| SINE   | SDG_VAL_WFM_SINE      | 1     |
| SQUARE | SDG_VAL_WFM_SQUARE    | 2     |
| RAMP   | SDG_VAL_WFM_RAMP      | 3     |
| RAMP   | SDG_VAL_WFM_RAMP_UP   | 4     |
| RAMP   | SDG_VAL_WFM_RAMP_DOWN | 5     |
| PULSE  | SDG_VAL_WFM_PULSE     | 1001  |
| NOISE  | SDG_VAL_WFM_NOISE     | 1002  |
| ARB    | SDG_VAL_WFM_ARB       | 1003  |
| DC     | SDG_VAL_WFM_DC        | 6     |
| PRBS   | SDG_VAL_WFM_PRBS      | 7     |

**Related Attribute**

SDG\_ATTR\_FUNC\_AMPLITUDE  
SDG\_ATTR\_FUNC\_FREQUENCY  
SDG\_ATTR\_FUNC\_START\_PHSAE

**High Level Functions**

`sdg_ConfigureSquareWaveformSpec`  
`sdg_ConfigureRampWaveformSpec`

## SDG\_ATTR\_FUNC\_AMPLITUDE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the amplitude of the standard waveform the function generator produces. This value is the amplitude at the output terminal.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FUNC_AMPLITUDE macro.<br><b>value</b> is used to store the value of function returned by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [2mV,24V]<br><br><b>Notes:</b><br>Only SDS7000A maximum amplitude is 24V, other models are 20V  |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_FREQUENCY<br>SDG_ATTR_FUNC_WAVEFORM<br>SDG_ATTR_FUNC_START_PHSAE   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureStandardWaveform</code>   |

## SDG\_ATTR\_FUNC\_DC\_OFFSET

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the DC offset of the standard waveform the function generator produces. This value is the offset at the output terminal. The units are volts (V).   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FUNC_AMPLITUDE macro.<br><b>value</b> is used to store the value of function returned by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [-12V,12V]<br><br><b>Notes:</b><br>Only SDS7000A maximum offset is $\pm 12V$ , other models are $\pm 10V$   |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_WAVEFORM   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureStandardWaveform</code>   |

## SDG\_ATTR\_FUNC\_FREQUENCY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the frequency of the standard waveform the function generator produces. The units are hertz (Hz).   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_FUNC_FREQUENCY macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1uHZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1uHZ,60M]<br/>SDG1000XPlus range is [1uHZ,60M]<br/>SDG2000X range is [1uHZ,120M]<br/>SDG6000X range is [1uHZ,500M]<br/>SDG7000A range is [1uHZ,1G]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_AMPLITUDE<br>SDG_ATTR_FUNC_WAVEFORM<br>SDG_ATTR_FUNC_START_PHSAE   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureAcquisitionRecord</code>  |

## SDG\_ATTR\_FUNC\_START\_PHASE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the horizontal offset of the standard waveform the function generator produces. You specify this attribute in degrees of one waveform cycle.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FUNC_START_PHASE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [-360° , 360° ].  |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_AMPLITUDE<br>SDG_ATTR_FUNC_FREQUENCY<br>SDG_ATTR_FUNC_WAVEFORM   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureStandardWaveform</code>   |

## SDG\_ATTR\_FUNC\_DUTY\_CYCLE\_HIGH

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies the length of time the output voltage level remains high in a square waveform. You specify this attribute as a percentage of one waveform cycle.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName (one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FUNC_DUTY_CYCLE_HIGH macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [0.01%, 99.9%].  |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_WAVEFORM  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureSquareWaveformSpec</code>  |

## SDG\_ATTR\_FUNC\_RAMP\_SYMMETRY

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based property specifies the symmetry of the output triangle wave. It can be set only when WVTP is a triangle wave, the unit is "%".   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</pre><br><pre>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_FUNC_RAMP_SYMMETRY macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0%,100%]  |
| <b>Related Attribute</b>        | SDG_ATTR_FUNC_WAVEFORM  |
| <b>High Level Functions</b>     | sdg_ConfigureRampWaveformSpec   |

## Arbitrary Waveform Output

Attributes for generating arbitrary waveform output. You use this group of attributes when you set the `SDG_ATTR_OUTPUT_MODE` attribute to `SDG_VAL_OUTPUT_ARB..` The Arbitrary Waveform group has the following attributes:

- ◆ **SDG\_ATTR\_ARB\_GAIN**
- ◆ **SDG\_ATTR\_ARB\_OFFSET**
- ◆ **SDG\_ATTR\_ARB\_SAMPLE\_RATE**
- ◆ **SDG\_ATTR\_ARB\_WAVEFORM\_HANDLE**
- ◆ **SDG\_ATTR\_MAX\_NUM\_WAVEFORMS**
- ◆ **SDG\_ATTR\_MAX\_WAVEFORM\_SIZE**
- ◆ **SDG\_ATTR\_MIN\_WAVEFORM\_SIZE**
- ◆ **SDG\_ATTR\_WAVEFORM\_QUANTUM**
- ◆ **SDG\_ATTR\_ARB\_FREQUENCY**

## SDG\_ATTR\_ARB\_GAIN

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the factor by which the function generator scales the arbitrary waveform data.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ARB_GAIN macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mV,24V]</p> <p><b>Notes:</b><br/>SDG1000X range is [2mV,20V]<br/>SDG1000XPlus range is [2mV,20V]<br/>SDG2000X range is [2mV,20V]<br/>SDG6000X range is [2mV,20V]<br/>SDG7000A range is [1mV,24V]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_ARB_WAVEFORM_HANDLE<br>SDG_ATTR_ARB_OFFSET  |
| <b>High Level Functions</b>     | sdg_ConfigureArbWaveform   |

## SDG\_ATTR\_ARB\_OFFSET

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the value the function generator adds to the arbitrary waveform data.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ARB_OFFSET macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [-12V,12V]</p> <p><b>Notes:</b><br/>SDG1000X range is [-9.999V,9.999V]<br/>SDG1000XPlus range is [-9.999V,9.999V]<br/>SDG2000X range is [-9.999V,9.999V]<br/>SDG6000X range is [-9.999V,9.999V]<br/>SDG7000A range is [-12V,12V]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_ARB_GAIN<br>SDG_ATTR_ARB_WAVEFORM_HANDLE  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureArbWaveform</code>  |

## SDG\_ATTR\_ARB\_SAMPLE\_RATE

|   |   |
|---|---|
| <b>Description</b>                            | Specifies the rate at which the function generator outputs the points in arbitrary waveforms. The units are samples per second.   |
| <b>Data Type</b>                              | ViReal64  |
| <b>Access</b>                                 | R/W   |
| <b>Common Control Functions</b>               | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ARB_SAMPLE_RATE macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>                            | <p>The value range is [1uSa/s,2.5GSa/s]</p> <p><b>Notes:</b><br/>SDG1000X range is [1uSa/s,30MSa/s]<br/>SDG1000XPlus range is [1uSa/s,250MSa/s]<br/>SDG2000X range is [1uSa/s,75MSa/s]<br/>SDG6000X range is [1uSa/s,300MSa/s]<br/>SDG7000A range is [10mSa/s, 2.5GSa/s]</p>  |
| <b>Related Attribute High Level Functions</b> | <p>None</p> <p><code>sdg_ConfigureSampleRate</code></p>   |



## SDG\_ATTR\_ARB\_WAVEFORM\_HANDLE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute identifies which arbitrary waveform the function generator produces.   |
| <b>Data Type</b>                | Vilnt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ARB_WAVEFORM_HANDLE macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | None  |
| <b>Related Attribute</b>        | SDG_ATTR_ARB_GAIN<br>SDG_ATTR_ARB_OFFSET  |
| <b>High Level Functions</b>     | sdg_ConfigureArbWaveform  |

## SDG\_ATTR\_MAX\_NUM\_WAVEFORMS

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Returns the maximum number of arbitrary waveforms that the function generator allows. Typically, this value is constant for the function generator.  |
| <b>Data Type</b>                | ViInt32  |
| <b>Access</b>                   | R  |
| <b>Common Control Functions</b> | <code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_MAX_NUM_WAVEFORMS macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | None   |
| <b>Related Attribute</b>        | SDG_ATTR_WAVEFORM_QUANTUM<br>SDG_ATTR_MAX_WAVEFORM_SIZE<br>SDG_ATTR_MIN_WAVEFORM_SIZE<br>SDG_ATTR_WFM_STRUCTURE  |
| <b>High Level Functions</b>     | <code>sdg_VerifyWfmCreatable</code><br><code>sdg_QueryArbWfmCapabilities</code><br><code>sdg_CreateArbRecords</code><br><code>sdg_GetNumDefinedWaveforms</code><br><code>sdg_WfmExists</code><br><code>sdg_GetWfmHandleFromNameUtil</code><br><code>sdg_ClearDriverWfmRecord</code>  |

## SDG\_ATTR\_MAX\_WAVEFORM\_SIZE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Returns the maximum number of points the function generator allows in an arbitrary waveform. Typically, this value is constant for the function generator.   |
| <b>Data Type</b>                | ViInt32  |
| <b>Access</b>                   | R  |
| <b>Common Control Functions</b> | <code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_MAX_WAVEFORM_SIZE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | None   |
| <b>Related Attribute</b>        | SDG_ATTR_WAVEFORM_QUANTUM<br>SDG_ATTR_MIN_WAVEFORM_SIZE  |
| <b>High Level Functions</b>     | <code>sdg_VerifyWfmCreatable</code><br><code>sdg_QueryArbWfmCapabilities</code>  |

## SDG\_ATTR\_MIN\_WAVEFORM\_SIZE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Returns the minimum number of points the function generator allows in an arbitrary waveform. Typically, this value is constant for the function generator.   |
| <b>Data Type</b>                | ViInt32  |
| <b>Access</b>                   | R  |
| <b>Common Control Functions</b> | <code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_MIN_WAVEFORM_SIZE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | None   |
| <b>Related Attribute</b>        | SDG_ATTR_WAVEFORM_QUANTUM<br>SDG_ATTR_MAX_WAVEFORM_SIZE  |
| <b>High Level Functions</b>     | <code>sdg_VerifyWfmCreatable</code><br><code>sdg_QueryArbWfmCapabilities</code>  |

## SDG\_ATTR\_WAVEFORM\_QUANTUM

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | The size of each arbitrary waveform must be a multiple of a quantum value. This attribute returns the quantum value the function generator allows.  |
| <b>Data Type</b>                | ViInt32   |
| <b>Access</b>                   | R   |
| <b>Common Control Functions</b> | <code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_WAVEFORM_QUANTUM macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | None  |
| <b>Related Attribute</b>        | SDG_ATTR_MAX_WAVEFORM_SIZE<br>SDG_ATTR_MIN_WAVEFORM_SIZE  |
| <b>High Level Functions</b>     | <code>sdg_VerifyWfmCreatable</code><br><code>sdg_QueryArbWfmCapabilities</code>   |

## SDG\_ATTR\_ARB\_FREQUENCY

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies the rate at which the function generator produces an entire arbitrary waveform. The units are waveforms per second.  |
| <b>Data type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common control functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_ARB_FREQUENCY macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [1uHZ,100MHZ]<br><br><b>Notes:</b><br>SDG1000X range is [1uHZ,6MHZ]<br>SDG1000XPlus range is [1uHZ,20MHZ]<br>SDG2000X range is [1uHZ,20MHZ]<br>SDG6000X range is [1uHZ,50MHZ]<br>SDG7000A range is [1uHZ,100MHZ]   |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureArbFrequency</code>  |

## Harmonic

Attributes for configuring the harmonic mode. The harmonic mode group has the following attributes:

- ◆ **SDG\_ATTR\_HARMONIC\_ENABLED**
- ◆ **SDG\_ATTR\_HARMONIC\_ORDER**
- ◆ **SDG\_ATTR\_HARMONIC\_SN**
- ◆ **SDG\_ATTR\_HARMONIC\_AMPL**
- ◆ **SDG\_ATTR\_HARMONIC\_PHASE**
- ◆ **SDG\_ATTR\_HARMONIC\_TYPE**

## SDG\_ATTR\_HARMONIC\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute enable or disable the harmonic mode.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_HARMONIC_ENABLEDmacro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value range</b>              | VI_TRUE means harmonic is on<br>VI_FALSE means harmonic is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_HARMONIC\_ORDER

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the harmonic source.  |
| <b>Data Type</b>                | Vilnt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_HARMONIC_ORDER macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The Value range is [2,16]</p> <p><b>Notes:</b><br/>SDG1000X range is [2,16]<br/>SDG1000XPlus range is [2,16]<br/>SDG2000X range is [2,10]<br/>SDG6000X range is [2,16]<br/>SDG7000A range is [2,16]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_HARMONIC_TYPE<br>SDG_ATTR_HARMONIC_PHASE<br>SDG_ATTR_HARMONIC_AMPL<br>SDG_ATTR_HARMONIC_SN  |
| <b>High Level Functions</b>     | sdg_InitAttributes   |

## SDG\_ATTR\_HARMONIC\_SN

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute is just a assistant for the attribute SDG_ATTR_HARMONIC_PHASE. You can use this attribute to specify which order of harmonic you will specify its phase.   |
| <b>Data Type</b>                | Vilnt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_HARMONIC_SN macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | None  |
| <b>Related Attribute</b>        | SDG_ATTR_HARMONIC_TYPE<br>SDG_ATTR_HARMONIC_PHASE<br>SDG_ATTR_HARMONIC_AMPL<br>SDG_ATTR_HARMONIC_ORDER  |
| <b>High Level Functions</b>     | sdg_InitAttributes  |

## SDG\_ATTR\_HARMONIC\_AMPL

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the amplitude of the specified order of harmonic.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</pre><br><pre>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_HARMONIC_AMPL macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [2mV, 24V].<br><br><b>Notes:</b><br>Only SDS7000A maximum amplitude is 24V, other models are 20V  |
| <b>Related Attribute</b>        | SDG_ATTR_HARMONIC_TYPE<br>SDG_ATTR_HARMONIC_PHASE<br>SDG_ATTR_HARMONIC_ORDER<br>SDG_ATTR_HARMONIC_SN   |
| <b>High Level Functions</b>     | sdg_InitAttributes   |

## SDG\_ATTR\_HARMONIC\_PHASE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the phase of the specified order of harmonic.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_HARMONIC_PHASE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [0° , 360° ].   |
| <b>Related Attribute</b>        | SDG_ATTR_HARMONIC_TYPE<br>SDG_ATTR_HARMONIC_AMPL<br>SDG_ATTR_HARMONIC_ORDER<br>SDG_ATTR_HARMONIC_SN  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>  |

## SDG\_ATTR\_HARMONIC\_TYPE

**Description** This channel-based attribute specifies the harmonic type to EVEN, ODD, ALL or USER.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_HARMONIC\_TYPE macro.

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

**Value Range**

| Type | Discrete Value             | Value |
|------|----------------------------|-------|
| EVEN | SDG_VAL_HARMONIC_TYPE_EVEN | 0     |
| ODD  | SDG_VAL_HARMONIC_TYPE_ODD  | 1     |
| ALL  | SDG_VAL_HARMONIC_TYPE_ALL  | 2     |

**Related Attribute** SDG\_ATTR\_HARMONIC\_PHASE  
SDG\_ATTR\_HARMONIC\_AMPL  
SDG\_ATTR\_HARMONIC\_ORDER  
SDG\_ATTR\_HARMONIC\_SN

**High Level Functions** `sdg_InitAttributes`

## Counter

The counter group properties are used to set or read counter related parameters. The counter group has the following attributes:

- ◆ **SDG\_ATTR\_COUNTER\_STATISTICS\_DISPLAY**
- ◆ **SDG\_ATTR\_COUNTER\_STATISTICS\_STATE**
- ◆ **SDG\_ATTR\_COUNTER\_STATE**
- ◆ **SDG\_ATTR\_COUNTER\_GATE\_TIME**
- ◆ **SDG\_ATTR\_COUNTER\_HF**
- ◆ **SDG\_ATTR\_COUNTER\_ATTENUATION**
- ◆ **SDG\_ATTR\_COUNTER\_COUPING**
- ◆ **SDG\_ATTR\_COUNTER\_IMPEDANCE**
- ◆ **SDG\_ATTR\_COUNTER\_LEVE**
- ◆ **SDG\_ATTR\_COUNTER\_SENSITIVE**

## SDG\_ATTR\_COUNTER\_STATISTICS\_STATE

**Description** This attribute enable or disable the measurement results statistic function of the counter.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_COUNTER\_STATISTICS\_STATE macro.

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

**Value Range**

| Type | Discrete Value                 | Value |
|------|--------------------------------|-------|
| ON   | SDG_VAL_COUNTER_STATISTICS_ON  | 0     |
| OFF  | SDG_VAL_COUNTER_STATISTICS_OFF | 1     |

**Related Attribute** None

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_COUNTER\_STATE

**Description** This attribute enable or disable the counter function.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_COUNTER\_STATE macro.

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

**Value Range**

| Type | Discrete Value            | Value |
|------|---------------------------|-------|
| ON   | SDG_VAL_COUNTER_STATE_ON  | 0     |
| OFF  | SDG_VAL_COUNTER_STATE_OFF | 1     |

**Related Attribute** None

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_COUNTER\_HF

**Description** This attribute enable or disable the high-frequency reject of the counter.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_COUNTER\_HF macro.

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

**Value Range**

| Type | Discrete Value         | Value |
|------|------------------------|-------|
| OFF  | SDG_VAL_COUNTER_HF_OFF | 0     |
| ON   | SDG_VAL_COUNTER_HF_ON  | 1     |

**Related Attribute** None

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_COUNTER\_COUPING

**Description** This attribute specifies the coupling mode of the counter to AC or DC..

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_COUNTER\_COUPING macro

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

**Value Range**

| Type | Discrete Value             | Value |
|------|----------------------------|-------|
| AC   | SDG_VAL_COUNTER_COUPING_AC | 0     |
| DC   | SDG_VAL_COUNTER_COUPING_D  | 1     |

**Related Attribute** None

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_COUNTER\_IMPEDANCE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute specifies the input impedance of the counter to 50 $\Omega$ or 1 M $\Omega$ .  |
| <b>Data Type</b>                | ViInt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</code><br><br><code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_COUNTER\_IMPEDANCE macro

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

**Value Range**

| Type | Discrete Value               | Value |
|------|------------------------------|-------|
| 50   | SDG_VAL_COUNTER_IMPEDANCE_50 | 0     |
| 1M   | SDG_VAL_COUNTER_IMPEDANCE_1M | 1     |

|                             |                                 |
|-----------------------------|---------------------------------|
| <b>Related Attribute</b>    | None                            |
| <b>High Level Functions</b> | <code>sdg_InitAttributes</code> |

## SDG\_ATTR\_COUNTER\_LEVEL

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute specifies the trigger level of the counter.  |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_COUNTER_LEVEL macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is[-3V,2.5V]  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>   |

## Bursting

The bursting group properties are used to set or read bursting related parameters. The bursting group has the following attributes:

- ◆ **SDG\_ATTR\_BURST\_NCYCLES\_ENABLED**
- ◆ **SDG\_ATTR\_BURST\_GATED\_ENABLED**
- ◆ **SDG\_ATTR\_BURST\_INFINITY\_ENABLED**
- ◆ **SDG\_ATTR\_BURST\_GATE\_POLARITY**
- ◆ **SDG\_ATTR\_BURST\_INTERNAL\_PERIOD**
- ◆ **SDG\_ATTR\_BURST\_PHASE**

- ◆ SDG\_ATTR\_BURST\_TDELAY
- ◆ SDG\_ATTR\_BURST\_TRIGGER\_SLOPE
- ◆ SDG\_ATTR\_BURST\_TRIGGER\_SOURCE
- ◆ SDG\_ATTR\_BURST\_TRIGGER\_TRIGOUT
- ◆ SDG\_ATTR\_BURST\_COUNT

## SDG\_ATTR\_BURST\_NCYCLES\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator is configured to N cycles burst mode.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</p> <p>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</p> <p><b>Notes:</b><br/> <b>vi</b> is the instrument handle.<br/> <b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/> <b>attributeld</b> is SDG_ATTR_BURST_NCYCLES_ENABLED macro.<br/> <b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>VI_TRUE means Ncycles is on</p> <p>VI_FALSE means Ncycles is off</p>   |
| <b>Related Attribute</b>        | NULL  |
| <b>High Level Functions</b>     | <p>sdg_ConfigureBurstNcycEnabled</p> <p>sdg_InitAttributes</p>  |

## SDG\_ATTR\_BURST\_GATED\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator is configured to gated burst mode.  |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_BURST_GATED_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means Gated is on<br>VI_FALSE means Gated is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureBurstGateEnabled</code><br><code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_BURST\_INFINITY\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator is configured to infinite burst mode.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is<br>SDG_ATTR_BURST_INFINITY_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means Infinity is on<br>VI_FALSE means Infinity is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_BURST\_GATE\_POLARITY

**Description** Specify it's high level or low level from the rear panel [Mod/FSK/Trig] that triggers the function generator to produce pulse train. This attribute is effective only when the burst mode is gated.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_BURST\_GATE\_POLARITY macro

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

**Value Range**

| Type | Discrete Value                       | Value |
|------|--------------------------------------|-------|
| POS  | SDG_VAL_BURST_GATE_POLARITY_POSITIVE | 0     |
| NEG  | SDG_VAL_BURST_GATE_POLARITY_NEGATIVE | 1     |

**Related Attribute** None

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_BURST\_INTERNAL\_PERIOD

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specify the period of the burst pulse train which means the time interval between two adjacent N-cycles pulse train.  |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_BURST_INTERNAL_PERIOD macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | NULL  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_BURST\_PHASE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specify the start phase of the pulse train, whose unit is degree.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_BURST_PHASE macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0°,360°]  |
| <b>Related Attribute</b>        | SDG_ATTR_BURST_TDELAY<br>SDG_ATTR_BURST_COUNT   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureBurstStartPhase</code><br><code>sdg_InitAttributes</code>  |

## SDG\_ATTR\_BURST\_TDELAY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specify the time interval between the function generator just received a triggering and it starts to produce pulse train.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_BURST_TDELAY macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0s,100s]   |
| <b>Related Attribute</b>        | SDG_ATTR_BURST_PHASE<br>SDG_ATTR_BURST_COUNT   |
| <b>High Level Functions</b>     | sdg_InitAttributes   |

## SDG\_ATTR\_BURST\_TRIGGER\_SLOPE

**Description** Specify it's rising edge or failling edge that make the function generator produce pulse train.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_BURST\_TRIGGER\_SLOPE macro

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

**Value Range**

| Type | Discrete Value                       | Value |
|------|--------------------------------------|-------|
| NORM | SDG_VAL_BURST_TRIGGER_SLOPE_NORMAL   | 0     |
| INV  | SDG_VAL_BURST_TRIGGER_SLOPE_INVERTED | 1     |

**Related Attribute** SDG\_ATTR\_BURST\_TRIGGER\_TRIGOUT  
SDG\_ATTR\_BURST\_TRIGGER\_SOURCE

**High Level Functions** `sdg_InitAttributes`

## SDG\_ATTR\_BURST\_TRIGGER\_SOURCE

**Description** Specify the trigger source of the function generator, whose value can be internal, external or manual (software).

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_BURST\_TRIGGER\_SOURCE macro

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

**Value Range**

| Type  | Discrete Value                          | Value |
|-------|---|-------|
| INT   | SDG_VAL_BURST_TRIGGER_SOURCE_INTERNAL   | 0     |
| EXT   | SDG_VAL_BURST_TRIGGER_SOURCE_EXTERNAL   | 1     |
| MAN   | SDG_VAL_BURST_TRIGGER_SOURCE_MANUAL     | 2     |
| EXT_A | SDG_VAL_BURST_TRIGGER_SOURCE_EXTERNAL_A | 3     |
| EXT_B | SDG_VAL_BURST_TRIGGER_SOURCE_EXTERNAL_B | 4     |
| MAN_A | SDG_VAL_BURST_TRIGGER_SOURCE_MANUAL_A   | 5     |
| MAN_B | SDG_VAL_BURST_TRIGGER_SOURCE_MANUAL_B   | 6     |

**Related Attribute** SDG\_ATTR\_BURST\_TRIGGER\_TRIGOUT  
SDG\_ATTR\_BURST\_TRIGGER\_SLOPE

**High Level Functions** `sdg_InitAttributes`



## SDG\_ATTR\_BURST\_TRIGGER\_TRIGOUT

**Description** Specify the edge type of the Trigger output signal.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_BURST\_TRIGGER\_TRIGOUT macro

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

**Value Range**

| Type | Discrete Value                         | Value |
|------|--|-------|
| OFF  | SDG_VAL_BURST_TRIGGER_TRIGOUT_OFF      | 0     |
| RISE | SDG_VAL_BURST_TRIGGER_TRIGOUT_POSITIVE | 1     |
| FALL | SDG_VAL_BURST_TRIGGER_TRIGOUT_NEGATIVE | 2     |

**Related Attribute** SDG\_ATTR\_BURST\_TRIGGER\_SOURCE  
SDG\_ATTR\_BURST\_TRIGGER\_SLOPE

**High Level Functions** `sdg_ConfigureBurstTrigOut`  
`sdg_InitAttributes`

## SDG\_ATTR\_BURST\_COUNT

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies the number of cycles that the function generator produces when it is in the burst operation mode and receives a trigger.  |
| <b>Data Type</b>                | ViInt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</code><br><br><code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_BURST_COUNT macro<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [1,1M]  |
| <b>Related Attribute</b>        | SDG_ATTR_BURST_PHASE<br>SDG_ATTR_BURST_TDELAY  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureBurstCount</code><br><code>sdg_InitAttributes</code>  |

## Sweep

The sweep group properties are used to set or read sweep related parameters. The sweep group has the following attributes:

- ◆ **SDG\_ATTR\_SWEEP\_STATE**
- ◆ **SDG\_ATTR\_SWEEP\_TIME**
- ◆ **SDG\_ATTR\_SWEEP\_TRIGGER\_SLOPE**
- ◆ **SDG\_ATTR\_SWEEP\_TRIGGER\_SOURCE**
- ◆ **SDG\_ATTR\_SWEEP\_TRIGGER\_TRIGOUT**
- ◆ **SDG\_ATTR\_SWEEP\_TYPE**
- ◆ **SDG\_ATTR\_SWEEP\_DIRECTION**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_CENTER**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_SPAN**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_START**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_STOP**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_START\_RECORD**
- ◆ **SDG\_ATTR\_SWEEP\_FREQUENCY\_STOP\_RECORD**

### SDG\_ATTR\_SWEEP\_STATE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Enable or disable the sweep function.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_SWEEP\_STATE macro.

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

**Value Range**

VI\_TRUE means Sweep is on

VI\_FALSE means Sweep is off

**Related Attribute**

SDG\_ATTR\_SWEEP\_TIME

SDG\_ATTR\_SWEEP\_TYPE

SDG\_ATTR\_SWEEP\_DIRECTION

**High Level Functions**

None

## SDG\_ATTR\_SWEEP\_TIME

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the sweep time and the default unit is "s".  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_SWEEP_TIME macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1ms,1000s]</p> <p><b>Notes:</b><br/>Only SDS7000A maximum time is 1000s, other models are 500s</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_TYPE<br>SDG_ATTR_SWEEP_STATE<br>SDG_ATTR_SWEEP_DIRECTION  |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_SWEEP\_TRIGGER\_SLOPE

| <b>Description</b>              | Set the generator to enable the sweep output on the rising (POSitive) or falling (NEGative) edge of the external trigger signal.  |       |                |       |     |                                      |   |     |                                      |   |
|---------------------------------|---|-------|----------------|-------|-----|--------------------------------------|---|-----|--------------------------------------|---|
| <b>Data Type</b>                | ViInt32   |       |                |       |     |                                      |   |     |                                      |   |
| <b>Access</b>                   | R/W   |       |                |       |     |                                      |   |     |                                      |   |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</pre><br><pre>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_SWEEP_TRIGGER_SLOPE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |       |                |       |     |                                      |   |     |                                      |   |
| <b>Value Range</b>              | <table border="1"><thead><tr><th>Type</th><th>Discrete Value</th><th>Value</th></tr></thead><tbody><tr><td>POS</td><td>SDG_VAL_SWEEP_TRIGGER_SLOPE_POSITIVE</td><td>0</td></tr><tr><td>NEG</td><td>SDG_VAL_SWEEP_TRIGGER_SLOPE_NEGATIVE</td><td>1</td></tr></tbody></table>   | Type  | Discrete Value | Value | POS | SDG_VAL_SWEEP_TRIGGER_SLOPE_POSITIVE | 0 | NEG | SDG_VAL_SWEEP_TRIGGER_SLOPE_NEGATIVE | 1 |
| Type                            | Discrete Value  | Value |                |       |     |                                      |   |     |                                      |   |
| POS                             | SDG_VAL_SWEEP_TRIGGER_SLOPE_POSITIVE  | 0     |                |       |     |                                      |   |     |                                      |   |
| NEG                             | SDG_VAL_SWEEP_TRIGGER_SLOPE_NEGATIVE  | 1     |                |       |     |                                      |   |     |                                      |   |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_TRIGGER_SLOPE<br>SDG_ATTR_SWEEP_TRIGGER_SOURCE<br>SDG_ATTR_SWEEP_TRIGGER_TRIGOUT   |       |                |       |     |                                      |   |     |                                      |   |
| <b>High Level Functions</b>     | None  |       |                |       |     |                                      |   |     |                                      |   |

## SDG\_ATTR\_SWEEP\_TRIGGER\_SOURCE

**Description** Set the trigger source of the sweep to INTERNAL,EXTERNAL or MANUAL. Query the trigger source.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_SWEEP\_TRIGGER\_SOURCE macro

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

**Value Range**

| Type  | Discrete Value                          | Value |
|-------|---|-------|
| INT   | SDG_VAL_SWEEP_TRIGGER_SOURCE_INTERNAL   | 0     |
| EXT   | SDG_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL   | 1     |
| MAN   | SDG_VAL_SWEEP_TRIGGER_SOURCE_MANUAL     | 2     |
| EXT_A | SDG_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL_A | 3     |
| EXT_B | SDG_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL_B | 4     |
| MAN_A | SDG_VAL_SWEEP_TRIGGER_SOURCE_MANUAL_A   | 5     |
| MAN_B | SDG_VAL_SWEEP_TRIGGER_SOURCE_MANUAL_B   | 6     |

**Related Attribute** None

**High Level Functions** `sdg_ConfigureSweepTriggerSource`

## SDG\_ATTR\_SWEEP\_TRIGGER\_TRIGOUT

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the edge type of the sweep trigger output to rising (POSitive) or falling (NEGative) edge or disable the trigger output signal (OFF).   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code></p> <p><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_SWEEP_TRIGGER_TRIGOUT macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | NULL  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureSweepTriggerOut</code>   |

## SDG\_ATTR\_SWEEP\_TYPE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the type of the sweep to Log or Linear.   |
| <b>Data Type</b>                | ViInt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</code><br><br><code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_SWEEP\_TYPE macro

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

**Value Range**

| Type | Discrete Value           | Value |
|------|--------------------------|-------|
| LINE | SDG_VAL_SWEEP_TYPE_LINER | 0     |
| LOG  | SDG_VAL_SWEEP_TYPE_LOG   | 1     |

|                          |   |
|--------------------------|---|
| <b>Related Attribute</b> | SDG_ATTR_SWEEP_TIME<br>SDG_ATTR_SWEEP_STATE<br>SDG_ATTR_SWEEP_DIRECTION |
|--------------------------|---|

|                             |      |
|-----------------------------|------|
| <b>High Level Functions</b> | None |
|-----------------------------|------|

## SDG\_ATTR\_SWEEP\_DIRECTION

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the direction of the sweep to Up,Down or Up_and_Down.  |
| <b>Data Type</b>                | Vilnt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_SWEEP\_DIRECTION macro

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

**Value Range**

| Type    | Discrete Value                  | Value |
|---------|---------------------------------|-------|
| UP      | SDG_VAL_SWEEP_DIRECTION_UP      | 0     |
| DOWN    | SDG_VAL_SWEEP_DIRECTION_DOWN    | 1     |
| UP_DOWN | SDG_VAL_SWEEP_DIRECTION_UP_DOWN | 2     |

**Notes:**

SDG6000X&SDG7000A can set

SDG\_VAL\_SWEEP\_DIRECTION\_UP\_DOWN in linear mode

|                          |   |
|--------------------------|---|
| <b>Related Attribute</b> | <p>SDG_ATTR_SWEEP_TIME</p> <p>SDG_ATTR_SWEEP_TYPE</p> <p>SDG_ATTR_SWEEP_STATE</p> |
|--------------------------|---|

|                             |                             |
|-----------------------------|-----------------------------|
| <b>High Level Functions</b> | sdg_ConfigureSweepDirection |
|-----------------------------|-----------------------------|

## SDG\_ATTR\_SWEEP\_FREQUENCY\_CENTER

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the center frequency of the sweep and the default unit is "Hz".   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_SWEEP_FREQUENCY_CENTER macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is[1uHZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1uHZ, 60MHZ]<br/>SDG1000XPlus range is [1uHZ, 60MHZ]<br/>SDG2000X range is [1uHZ,120MHZ]<br/>SDG6000X range is [1uHZ, 500MHZ]<br/>SDG7000A range is [1uHZ,1GHZ]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_FREQUENCY_SPAN<br>SDG_ATTR_SWEEP_FREQUENCY_START<br>SDG_ATTR_SWEEP_FREQUENCY_STOP  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_SWEEP\_FREQUENCY\_SPAN

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the frequency span of the sweep and the default unit is "Hz".   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_SWEEP_FREQUENCY_SPAN macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is[0HZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [0HZ, 60MHZ]<br/>SDG1000XPlus range is [0HZ, 60MHZ]<br/>SDG2000X range is [0HZ,120MHZ]<br/>SDG6000X range is [0HZ, 500MHZ]<br/>SDG7000A range is [0HZ,1GHZ]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_FREQUENCY_START<br>SDG_ATTR_SWEEP_FREQUENCY_STOP<br>SDG_ATTR_SWEEP_FREQUENCY_CENTER  |
| <b>High Level Functions</b>     | <code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_SWEEP\_FREQUENCY\_START

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the start frequency of the sweep and the default unit is "Hz".   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_SWEEP_FREQUENCY_START macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1uHZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1uHZ, 60MHZ]<br/>SDG1000XPlus range is [1uHZ, 60MHZ]<br/>SDG2000X range is [1uHZ,120MHZ]<br/>SDG6000X range is [1uHZ, 500MHZ]<br/>SDG7000A range is [1uHZ,1GHZ]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_FREQUENCY_SPAN<br>SDG_ATTR_SWEEP_FREQUENCY_STOP<br>SDG_ATTR_SWEEP_FREQUENCY_CENTER  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureSweepFrequencyStart</code><br><code>sdg_InitAttributes</code>   |

## SDG\_ATTR\_SWEEP\_FREQUENCY\_STOP

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the end frequency of the sweep and the default unit is "Hz".  |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_SWEEP_FREQUENCY_STOP macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1uHZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1uHZ, 60MHZ]<br/>SDG1000XPlus range is [1uHZ, 60MHZ]<br/>SDG2000X range is [1uHZ,120MHZ]<br/>SDG6000X range is [1uHZ, 500MHZ]<br/>SDG7000A range is [1uHZ,1GHZ]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_FREQUENCY_SPAN<br>SDG_ATTR_SWEEP_FREQUENCY_START<br>SDG_ATTR_SWEEP_FREQUENCY_CENTER  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureSweepFrequencyStop</code><br><code>sdg_InitAttributes</code>   |

**SDG\_ATTR\_SWEEP\_FREQUENCY\_START\_RECORD**

|                                     |   |
|-------------------------------------|---|
| <b>Description</b>                  | Auxiliary attributes of<br>SDG_ATTR_SWEEP_FREQUENCY_START.  |
| <b>Data Type</b>                    | ViReal64  |
| <b>Access</b>                       | NULL  |
| <b>Common Control<br/>Functions</b> | sdg_SetAttributeViReal64(ViSession vi, ViConstString<br>repCapName, ViAttr attributeld, ViReal64 value)<br><br><b>Notes:</b><br><b>vi</b> is the instrument handle<br><b>repCapName</b> is channelName(one of the following<br>analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is<br>SDG_ATTR_SWEEP_FREQUENCY_START_RECORD<br>macro.<br><b>value</b> is used to store or set the value of function<br>represented by <b>attributeld</b> |
| <b>Value Range</b>                  | NULL  |
| <b>Related Attribute</b>            | SDG_ATTR_SWEEP_FREQUENCY_STOP_RECORD  |
| <b>High Level Functions</b>         | None  |

## SDG\_ATTR\_SWEEP\_FREQUENCY\_STOP\_RECORD

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Auxiliary attributes of<br>SDG_ATTR_SWEEP_FREQUENCY_STOP.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | NULL  |
| <b>Common Control Functions</b> | sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)<br><br><b>Notes:</b><br><b>vi</b> is the instrument handle<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is<br>SDG_ATTR_SWEEP_FREQUENCY_START_RECORD macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> |
| <b>Value Range</b>              | NULL  |
| <b>Related Attribute</b>        | SDG_ATTR_SWEEP_FREQUENCY_START_RECORD   |
| <b>High Level Functions</b>     | None  |

## IQ

The I/Q vector signal group properties are used to set or read I/Q vector signal related parameters. The I/Q vector signal group has the following attributes:

- ◆ **SDG\_ATTR\_IQ\_FREQUENCY\_CENTER**
- ◆ **SDG\_ATTR\_IQ\_SAMPLE\_RATE**
- ◆ **SDG\_ATTR\_IQ\_SYMBOL\_RATE**
- ◆ **SDG\_ATTR\_IQ\_AMPLITUDE**
- ◆ **SDG\_ATTR\_IQ\_TRIGGER\_SOURCE**
- ◆ **SDG\_ATTR\_IQ\_ADJ\_GAIN**
- ◆ **SDG\_ATTR\_IQ\_ADJ\_I\_OFFSET**
- ◆ **SDG\_ATTR\_IQ\_ADJ\_Q\_OFFSET**
- ◆ **SDG\_ATTR\_IQ\_ADJ\_Q\_SKEW**
- ◆ **SDG\_ATTR\_IQ\_WAVE\_BUILT\_IN**
- ◆ **SDG\_ATTR\_IQ\_WAVE\_USER**

## SDG\_ATTR\_IQ\_FREQUENCY\_CENTER

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the center frequency of the IQ and the default unit is "Hz".   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_FREQUENCY_CENTER macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1HZ,1GHZ]</p> <p><b>Notes:</b><br/>SDG6000X range is [1HZ,500MHZ]<br/>SDG7000A range is [1HZ,1GHZ]<br/>Other models are not involved</p>  |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureIqFrequencyCenter</code>  |

## SDG\_ATTR\_IQ\_SAMPLE\_RATE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specifies the rate at which the function generator outputs the points in IQ waveforms.  |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_SAMPLE_RATE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1KHZ,1.25GHZ]</p> <p><b>Notes:</b><br/>SDG6000X range is [1KHZ,300MHZ]<br/>SDG7000A range is [1KHZ,1.25GHZ]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_SYMBOL_RATE   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureIqSampleRate</code>  |

## SDG\_ATTR\_IQ\_SYMBOL\_RATE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specifies the rate at which the function generator outputs oversampled points in the IQ waveform. This parameter is limited by the oversampling factor  |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_SYMBOL_RATE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [250Sa/s,312.5MSa/s]</p> <p><b>Notes:</b><br/>SDG6000X range is [250Sa/s,75MSa/s]<br/>SDG7000A range is [250Sa/s,312.5MSa/s]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_SAMPLE_RATE   |
| <b>High Level Functions</b>     | sdg_ConfigureIqSymbolRate   |

## SDG\_ATTR\_IQ\_AMPLITUDE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute specifies the amplitude of the IQ.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_IQ_AMPLITUDE macro<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [0mVrms,939mVrms]<br><br><b>Notes:</b><br>SDG6000X range is [0mVrms,225.8mVrms]<br>SDG7000A range is [0mVrms, 939mVrms]  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | Nnoe  |

## SDG\_ATTR\_IQ\_TRIGGER\_SOURCE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute specifies the trigger source.  |
| <b>Data Type</b>                | ViInt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</code><br><br><code>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</code> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_IQ\_TRIGGER\_SOURCE macro

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

**Value Range**

| Type | Discrete Value                     | Value |
|------|------------------------------------|-------|
| INT  | SDG_VAL_IQ_TRIGGER_SOURCE_INTERNAL | 0     |
| EXT  | SDG_VAL_IQ_TRIGGER_SOURCE_EXTERNAL | 1     |
| MAN  | SDG_VAL_IQ_TRIGGER_SOURCE_MANUAL   | 2     |

|                             |   |
|-----------------------------|---|
| <b>Related Attribute</b>    | None                                      |
| <b>High Level Functions</b> | <code>sdg_ConfigureIqTriggerSource</code> |

## SDG\_ATTR\_IQ\_ADJ\_GAIN

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used for amplitude gain balance, which adjusts the amplitude difference between the two I/Q channels.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_ADJ_GAIN macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [-4dB,4 dB]   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureIqAdjGain</code>  |

## SDG\_ATTR\_IQ\_ADJ\_I\_OFFSET

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used to adjust the offset of the I channel. Combined with the Q-channel DC bias adjustment to compensate for the bias imbalance of the I/Q channels  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_ADJ_J_OFFSET macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [-0.25V,0.25V]  |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_ADJ_Q_OFFSET<br>SDG_ATTR_IQ_ADJ_Q_SKEW   |
| <b>High Level Functions</b>     | sdg_ConfigureIqAdjIOffset  |

## SDG\_ATTR\_IQ\_ADJ\_Q\_OFFSET

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used to adjust the offset of the I channel. Combined with the I-channel DC bias adjustment to compensate for the bias imbalance of the I/Q channels  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_ADJ_Q_OFFSET macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [-0.25V,0.25V]  |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_ADJ_I_OFFSET<br>SDG_ATTR_IQ_ADJ_Q_SKEW   |
| <b>High Level Functions</b>     | sdg_ConfigureIqAdjQOffset  |

## SDG\_ATTR\_IQ\_ADJ\_Q\_SKEW

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used for the phase angle adjustment of the Q channel to compensate for the phase imbalance between the I/Q channels  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_ADJ_Q_SKEW macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [-10° ,10° ]  |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_ADJ_I_OFFSET<br>SDG_ATTR_IQ_ADJ_Q_OFFSET   |
| <b>High Level Functions</b>     | sdg_ConfigureIqAdjSkew   |

## SDG\_ATTR\_IQ\_WAVE\_BUILT\_IN

**Description** This attribute is used to select the I/Q waveform from the built-in waveform list.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_IQ\_WAVE\_BUILT\_IN macro

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

**Value Range**

| Type   | Discrete Value                  | Value |
|--------|---------------------------------|-------|
| 2ASK   | SDG_VAL_IQ_WAVE_BUILT_IN_2ASK   | 0     |
| 4ASK   | SDG_VAL_IQ_WAVE_BUILT_IN_4ASK   | 1     |
| 8ASK   | SDG_VAL_IQ_WAVE_BUILT_IN_8ASK   | 2     |
| BPSK   | SDG_VAL_IQ_WAVE_BUILT_IN_BPSK   | 3     |
| QPSK   | SDG_VAL_IQ_WAVE_BUILT_IN_QPSK   | 4     |
| 8PSK   | SDG_VAL_IQ_WAVE_BUILT_IN_8PSK   | 5     |
| DBPSK  | SDG_VAL_IQ_WAVE_BUILT_IN_DBPSK  | 6     |
| DQPSK  | SDG_VAL_IQ_WAVE_BUILT_IN_DQPSK  | 7     |
| D8PSK  | SDG_VAL_IQ_WAVE_BUILT_IN_D8PSK  | 8     |
| 8QAM   | SDG_VAL_IQ_WAVE_BUILT_IN_8QAM   | 9     |
| 16QAM  | SDG_VAL_IQ_WAVE_BUILT_IN_16QAM  | 10    |
| 32QAM  | SDG_VAL_IQ_WAVE_BUILT_IN_32QAM  | 11    |
| 64QAM  | SDG_VAL_IQ_WAVE_BUILT_IN_64QAM  | 12    |
| 128QAM | SDG_VAL_IQ_WAVE_BUILT_IN_128QAM | 13    |
| 256QAM | SDG_VAL_IQ_WAVE_BUILT_IN_256QAM | 14    |

**Related Attribute** SDG\_ATTR\_IQ\_WAVE\_USER

**High Level Functions** `sdg_ConfigureIqWaveBuiltIn`

## SDG\_ATTR\_IQ\_WAVE\_USER

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used to select the I/Q waveform among the user-stored waveforms.   |
| <b>Data Type</b>                | ViString   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViString(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViConstString value)</code></p> <p><code>sdg_GetAttributeViString(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 bufSize, ViChar value[])</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_IQ_WAVE_USER macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | NULL   |
| <b>Related Attribute</b>        | SDG_ATTR_IQ_WAVE_BUILT_IN  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureIqWaveUser</code>   |

## PRBS

The PRBS group properties are used to set or read PRBS related parameters. The PRBS group has the following attributes:

- ◆ **SDG\_ATTR\_PRBS\_BITRATE**
- ◆ **SDG\_ATTR\_PRBS\_AMPLITUDE**
- ◆ **SDG\_ATTR\_PRBS\_OFFSET**
- ◆ **SDG\_ATTR\_PRBS\_LENGTH**
- ◆ **SDG\_ATTR\_PRBS\_EDGE**
- ◆ **SDG\_ATTR\_PRBS\_DIFFSTATE**
- ◆ **SDG\_ATTR\_PRBS\_LOGIC\_STATE**

## SDG\_ATTR\_PRBS\_BITRATE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute is used to set the bit rate/symbol period of PRBS.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PRBS_BITRATE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1ubps,312.5Mbps]</p> <p><b>Notes:</b><br/>SDG1000XPlus range is [1ubps, 40Mbps]<br/>SDG6000X range is [1ubps, 300Mbps]<br/>SDG7000A range is [1ubps,312.5Mbps]<br/>Other models are not involved.</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_PRBS_AMPLITUDE<br>SDG_ATTR_PRBS_OFFSET   |
| <b>High Level Functions</b>     | sdg_ConfigurePrbsBitRate  |

## SDG\_ATTR\_PRBS\_AMPLITUDE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute specifies the amplitude of the PRBS.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</pre><br><pre>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PRBS_AMPLITUDE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [2mV,24V]<br><br><b>Notes:</b><br>SDG1000XPlus range is [2mV,20V]<br>SDG6000X range is [2mV,20V]<br>SDG7000A range is [2mV,24V]   |
| <b>Related Attribute</b>        | SDG_ATTR_PRBS_OFFSET<br>SDG_ATTR_PRBS_BITRATE  |
| <b>High Level Functions</b>     | sdg_ConfigurePrbsAmplitude   |

## SDG\_ATTR\_PRBS\_OFFSET

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute specifies the offset of the PRBS  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_PRBS_OFFSET macro<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [-12V,12V]<br><br><b>Notes:</b><br>SDG1000XPlus range is [-9.99V,9.99V]<br>SDG6000X range is [-9.99V,9.99V]<br>SDG7000A range is [-12V,12V]   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigurePrbsOffset</code>   |

## SDG\_ATTR\_PRBS\_LENGTH

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute specifies the length of the PRBS. The settable range is PRBS-3 ~ 32. Its length = $2^{length} - 1$  |
| <b>Data Type</b>                | Vilnt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PRBS_LENGTH macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [3,32]  |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | sdg_ConfigurePrbsLength  |

## SDG\_ATTR\_PRBS\_EDGE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This attribute specifies the edge rise/fall time of PRBS.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</pre><br><pre>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PRBS_EDGE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0.5ns,1us]<br><br><b>Notes:</b><br>SDG6000XPlus range is [10ns,1us]<br>SDG6000X range is [1ns,1us]<br>SDG7000A range is [0.5ns,1us]   |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | <code>sdg_ConfigurePrbsEdge</code>  |

## SDG\_ATTR\_PRBS\_DIFFSTATE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute specifies the differential output switch mode of the PRBS.  |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code></p> <p><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PRBS_DIFFSTATE macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | None   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigurePrbsDifferentialState</code>  |

## SDG\_ATTR\_PRBS\_LOGIC\_STATE

**Description** This attribute is used to set the logic level of PRBS.

**Data Type** ViInt32

**Access** W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_PRBS\_LOGIC\_STATE macro

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

**Value Range**

| Type | Discrete Value   | Value |
|------|------------------|-------|
| TTL  | SDG_VAL_PRBS_TTL | 0     |

**Related Attribute** None

**High Level Functions** `sdg_ConfigurePrbsLogicLevel`

## MOD

The Modulation group properties are used to set or read Modulation related parameters. The Modulation group has the following attributes:

- ◆ **SDG\_ATTR\_MOD\_TYPE**
- ◆ **SDG\_ATTR\_MOD\_STATE**
- ◆ **SDG\_ATTR\_AM\_ENABLED**
- ◆ **SDG\_ATTR\_AM\_SOURCE**
- ◆ **SDG\_ATTR\_AM\_INTERNAL\_DEPTH**
- ◆ **SDG\_ATTR\_AM\_INTERNAL\_WAVEFORM**
- ◆ **SDG\_ATTR\_AM\_INTERNAL\_FREQUENCY**
- ◆ **SDG\_ATTR\_FM\_ENABLED**
- ◆ **SDG\_ATTR\_FM\_SOURCE**
- ◆ **SDG\_ATTR\_FM\_INTERNAL\_DEVIATION**
- ◆ **SDG\_ATTR\_FM\_INTERNAL\_WAVEFORM**
- ◆ **SDG\_ATTR\_FM\_INTERNAL\_FREQUENCY**
- ◆ **SDG\_ATTR\_PM\_ENABLED**
- ◆ **SDG\_ATTR\_PM\_SOURCE**
- ◆ **SDG\_ATTR\_PM\_DEVIATION**
- ◆ **SDG\_ATTR\_PM\_INTERNAL\_WAVEFORM**
- ◆ **SDG\_ATTR\_PM\_INTERNAL\_FREQUENCY**
- ◆ **SDG\_ATTR\_ASK\_ENABLED**
- ◆ **SDG\_ATTR\_ASK\_SOURCE**
- ◆ **SDG\_ATTR\_ASK\_POLARITY**
- ◆ **SDG\_ATTR\_ASK\_INTERNAL\_RATE**
- ◆ **SDG\_ATTR\_ASK\_AMPLITUDE**
- ◆ **SDG\_ATTR\_FSK\_ENABLED**

- ◆ SDG\_ATTR\_FSK\_SOURCE
- ◆ SDG\_ATTR\_FSK\_FREQUENCY
- ◆ SDG\_ATTR\_FSK\_POLARITY
- ◆ SDG\_ATTR\_FSK\_INTERNAL\_RATE
- ◆ SDG\_ATTR\_PSK\_ENABLED
- ◆ SDG\_ATTR\_PSK\_SOURCE
- ◆ SDG\_ATTR\_PSK\_POLARITY
- ◆ SDG\_ATTR\_PSK\_PHASE
- ◆ SDG\_ATTR\_PSK\_INTERNAL\_RATE
- ◆ SDG\_ATTR\_PWM\_ENABLED
- ◆ SDG\_ATTR\_PWM\_SOURCE
- ◆ SDG\_ATTR\_PWM\_DEVIATION\_DCYCLE
- ◆ SDG\_ATTR\_PWM\_DEVIATION\_WIDTH
- ◆ SDG\_ATTR\_PWM\_INTERNAL\_WAVEFORM
- ◆ SDG\_ATTR\_PWM\_INTERNAL\_FREQUENCY
- ◆ SDG\_ATTR\_PULSE\_WIDTH
- ◆ SDG\_ATTR\_PULSE\_PERIOD

## SDG\_ATTR\_MOD\_TYPE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Select the modulation mode.  |
| <b>Data Type</b>                | Vilnt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value) |

sdg\_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 \*value)

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_MOD\_TYPE macro

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

**Value Range**

| Type   | Discrete Value         | Value |
|--------|------------------------|-------|
| AM     | SDG_VAL_MOD_TYPE_AM    | 0     |
| FM     | SDG_VAL_MOD_TYPE_FM    | 1     |
| PM     | SDG_VAL_MOD_TYPE_PM    | 2     |
| ASK    | SDG_VAL_MOD_TYPE_ASK   | 3     |
| FSK    | SDG_VAL_MOD_TYPE_FSK   | 4     |
| PSK    | SDG_VAL_MOD_TYPE_PSK   | 5     |
| PWM    | SDG_VAL_MOD_TYPE_PWM   | 6     |
| BPSK   | SDG_VAL_MOD_TYPE_BPSK  | 7     |
| QPSK   | SDG_VAL_MOD_TYPE_QPSK  | 8     |
| 3PSK   | SDG_VAL_MOD_TYPE_3FSK  | 9     |
| 4PSK   | SDG_VAL_MOD_TYPE_4FSK  | 10    |
| OSK    | SDG_VAL_MOD_TYPE_OSK   | 11    |
| DSB-SC | SDG_VAL_MOD_TYPE_DSBSC | 12    |

**Notes:**

Different models support different modulation types

|                             |      |
|-----------------------------|------|
| <b>Related Attribute</b>    | None |
| <b>High Level Functions</b> | None |

## SDG\_ATTR\_MOD\_STATE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Enable or disable the modulation function.  |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_MOD_STATE macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means MOD is on<br>VI_FALSE means MOD is off  |
| <b>Related Attribute</b>        | SDG_ATTR_MOD_TYPE   |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_AM\_ENABLED

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator applies amplitude modulation to the output signal.   |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_AM_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means AM is on<br>VI_FALSE means AM is off   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureAMEnabled</code>  |

## SDG\_ATTR\_AM\_SOURCE

**Description** This channel-based attribute specifies the signal that the function generator uses to modulate the output signal.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_AM\_SOURCE macro

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

**Value Range**

| Type | Discrete Value      | Value |
|------|---------------------|-------|
| INT  | SDG_VAL_AM_INTERNAL | 0     |
| EXT  | SDG_VAL_AM_EXTERNAL | 1     |
| CH1  | SDG_VAL_AM_CH1      | 2     |
| CH2  | SDG_VAL_AM_CH2      | 3     |

**Related Attribute** None

**High Level Functions** `sdg_ConfigureAMSource`

## SDG\_ATTR\_AM\_INTERNAL\_DEPTH

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the extent of modulation the function generator applies to the carrier signal.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_AM_INTERNAL_DEPTH macro<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [0%,120%]   |
| <b>Related Attribute</b>        | SDG_ATTR_AM_INTERNAL_FREQUENCY<br>SDG_ATTR_AM_INTERNAL_WAVEFORM  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureAMInternal</code>   |

## SDG\_ATTR\_AM\_INTERNAL\_WAVEFORM

**Description** This attribute turns on or off the measurement gate.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_AM\_INTERNAL\_WAVEFORM macro

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

**Value Range**

| Type     | Discrete Value                | Value |
|----------|-------------------------------|-------|
| SINE     | SDG_VAL_AM_INTERNAL_SINE      | 0     |
| SQUARE   | SDG_VAL_AM_INTERNAL_SQUARE    | 1     |
| TRIANGLE | SDG_VAL_AM_INTERNAL_TRIANGLE  | 2     |
| UPRAMP   | SDG_VAL_AM_INTERNAL_RAMP_UP   | 3     |
| DNRAMP   | SDG_VAL_AM_INTERNAL_RAMP_DOWN | 4     |
| NOISE    | SDG_VAL_AM_INTERNAL_NOISE     | 5     |
| ARB      | SDG_VAL_AM_INTERNAL_ARB       | 6     |

**Related Attribute** SDG\_ATTR\_AM\_INTERNAL\_DEPTH  
SDG\_ATTR\_AM\_INTERNAL\_FREQUENCY

**High Level Functions** `sdg_ConfigureAMInternal`

## SDG\_ATTR\_AM\_INTERNAL\_FREQUENCY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the frequency of the standard waveform that the function generator uses to modulate the output signal.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_AM_INTERNAL_FREQUENCY macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,20KHZ]<br/>SDG1000XPlus range is [1mHZ,1MHZ]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_AM_INTERNAL_DEPTH<br>SDG_ATTR_AM_INTERNAL_WAVEFORM  |
| <b>High Level Functions</b>     | sdg_ConfigureAMInternal  |

## SDG\_ATTR\_FM\_ENABLED

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies whether the function generator applies frequency modulation to the output signal.  |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FM_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means FM is on<br>VI_FALSE means FM is off   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | <code>sdg_ConfigureFMEnabled</code>  |

## SDG\_ATTR\_FM\_SOURCE

**Description** Specifies the signal the function generator uses to modulate the output signal.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_FM\_SOURCE macro

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

**Value Range**

| Type | Discrete Value      | Value |
|------|---------------------|-------|
| INT  | SDG_VAL_FM_INTERNAL | 0     |
| EXT  | SDG_VAL_FM_EXTERNAL | 1     |
| CH1  | SDG_VAL_FM_CH1      | 2     |
| CH2  | SDG_VAL_FM_CH2      | 3     |

**Related Attribute** None

**High Level Functions** `sdg_ConfigureFMSource`

## SDG\_ATTR\_FM\_INTERNAL\_DEVIATION

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the maximum frequency deviation the modulating waveform applies to the carrier waveform.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_FM_INTERNAL_DEVIATION macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1uHZ,500MHZ]</p> <p><b>Notes:</b><br/>The max deviation is 0.5*BW(BW stands for maximum output frequency)</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_FM_INTERNAL_WAVEFORM<br>SDG_ATTR_FM_INTERNAL_FREQUENCY  |
| <b>High Level Functions</b>     | <code>sdg_ConfigureFMInternal</code>   |

## SDG\_ATTR\_FM\_INTERNAL\_WAVEFORM

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the standard waveform type that the function generator uses to modulate the output signal.   |
| <b>Data Type</b>                | Vilnt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</p> <p>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</p> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_FM\_INTERNAL\_WAVEFORM macro

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

**Value Range**

| Type     | Discrete Value                | Value |
|----------|-------------------------------|-------|
| SINE     | SDG_VAL_FM_INTERNAL_SINE      | 0     |
| SQUARE   | SDG_VAL_FM_INTERNAL_SQUARE    | 1     |
| TRIANGLE | SDG_VAL_FM_INTERNAL_TRIANGLE  | 2     |
| UPRAMP   | SDG_VAL_FM_INTERNAL_RAMP_UP   | 3     |
| DNRAMP   | SDG_VAL_FM_INTERNAL_RAMP_DOWN | 4     |
| NOISE    | SDG_VAL_FM_INTERNAL_NOISE     | 5     |
| ARB      | SDG_VAL_FM_INTERNAL_ARB       | 6     |

**Notes:**

Only SDG7000A can modulate noise

**Related Attribute**

SDG\_ATTR\_FM\_INTERNAL\_DEVIATION  
SDG\_ATTR\_FM\_INTERNAL\_FREQUENCY

**High Level Functions**

sdg\_ConfigureFMInterna

## SDG\_ATTR\_FM\_INTERNAL\_FREQUENCY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the frequency of the standard waveform that the function generator uses to modulate the output signal.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_FM_INTERNAL_FREQUENCY macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,20KHZ]<br/>SDG1000XPlus range is [1mHZ,1MHZ]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_FM_INTERNAL_DEVIATION<br>SDG_ATTR_FM_INTERNAL_WAVEFORM  |
| <b>High Level Functions</b>     | sdg_ConfigureFMInternal  |

## SDG\_ATTR\_PM\_ENABLED

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies whether the function generator applies phase modulation to the output signal.  |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</p> <p>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PM_ENABLED macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | VI_TRUE means PM is on<br>VI_FALSE means PM is off   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PM\_SOURCE

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specifies the signal the function generator uses to modulate the output signal.   |
| <b>Data Type</b>                | Vilnt32   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)</code><br><br><code>sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)</code> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_PM\_SOURCE macro

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

**Value Range**

| Type | Discrete Value      | Value |
|------|---------------------|-------|
| INT  | SDG_VAL_PM_INTERNAL | 0     |
| EXT  | SDG_VAL_PM_EXTERNAL | 1     |

**Related Attribute** None

**High Level Functions** None

## SDG\_ATTR\_PM\_DEVIATION

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Specifies the maximum frequency deviation the modulating waveform applies to the carrier waveform.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</pre><br><pre>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</pre> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PM_DEVIATION macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0° ,360° ]   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PM\_INTERNAL\_WAVEFORM

**Description** Specifies the standard waveform type that the function generator uses to modulate the output signal.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_PM\_INTERNAL\_WAVEFORM macro

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

**Value Range**

| Type  | Discrete Value               | Value |
|-------|------------------------------|-------|
| SIN   | SDG_VAL_PM_INTERNAL_SINE     | 0     |
| SQU   | SDG_VAL_PM_INTERNAL_SQUARE   | 1     |
| TRI   | SDG_VAL_PM_INTERNAL_TRIANGLE | 2     |
| RAMP  | SDG_VAL_PM_INTERNAL_RAMP     | 3     |
| DRAM  | SDG_VAL_PM_INTERNAL_NRAMP    | 4     |
| NOISE | SDG_VAL_PM_INTERNAL_NOISE    | 5     |
| USER  | SDG_VAL_PM_INTERNAL_USER     | 6     |

**Related Attribute** None

**High Level Functions** None

## SDG\_ATTR\_PM\_INTERNAL\_FREQUENCY

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specifies the frequency of the standard waveform that the function generator uses to modulate the output signal   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_PM_INTERNAL_FREQUENCY macro</p> <p><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,20KHZ]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_ASK\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Specifies whether the function generator applies amplitude shift keying modulation to the output signal.  |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</p> <p>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ASK_ENABLED macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | VI_TRUE means ASK is on<br>VI_FALSE means ASK is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_ASK\_SOURCE

**Description** Specifies the signal the function generator uses to modulate the output signal.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_ASK\_SOURCE macro

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

**Value Range**

| Type | Discrete Value       | Value |
|------|----------------------|-------|
| INT  | SDG_VAL_ASK_INTERNAL | 0     |
| EXT  | SDG_VAL_ASK_EXTERNAL | 1     |

**Related Attribute** SDG\_ATTR\_ASK\_AMPLITUDE  
SDG\_ATTR\_ASK\_INTERNAL\_RATE

**High Level Functions** None

## SDG\_ATTR\_ASK\_INTERNAL\_RATE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the ASK(amplitude shift keying) rate.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code><br><br><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_ASK_INTERNAL_RATE macro<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | The value range is [1mHZ,2MHZ]<br><br><b>Notes:</b><br>SDG1000X range is [1mHZ,50KHZ]<br>SDG2000X range is [1mHZ,1MHZ]<br>SDG6000X range is [1mHZ,1MHZ]<br>SDG7000A range is [1mHZ, 2MHZ]  |
| <b>Related Attribute</b>        | SDG_ATTR_ASK_AMPLITUDE<br>SDG_ATTR_ASK_SOURCE  |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_ASK\_AMPLITUDE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the amplitude of the ASK(amplitude shift keying) modulating waveform.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_ASK_AMPLITUDE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [0V,10V]  |
| <b>Related Attribute</b>        | SDG_ATTR_ASK_INTERNAL_RATE<br>SDG_ATTR_ASK_SOURCE  |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_FSK\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator applies frequency shift keying modulation to the output signal.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_FSK_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means FSK is on<br>VI_FALSE means FSK is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_FSK\_SOURCE

**Description** This channel-based attribute specifies whether the function generator applies frequency shift keying modulation to the output signal.

**Data Type** ViInt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)`

`sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_FSK\_SOURCE macro

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

**Value Range**

| Type | Discrete Value       | Value |
|------|----------------------|-------|
| INT  | SDG_VAL_FSK_INTERNAL | 0     |
| EXT  | SDG_VAL_FSK_EXTERNAL | 1     |

**Related Attribute** SDG\_ATTR\_FSK\_FREQUENCY  
SDG\_ATTR\_FSK\_INTERNAL\_RATE

**High Level Functions** None

## SDG\_ATTR\_FSK\_FREQUENCY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the FSK(Frequency Shift Keying) hop frequency.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_FSK_FREQUENCY macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1uHZ,1GHZ]</p> <p><b>Notes:</b><br/>The max deciation is BW(BW stands for maximum output frequency)</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_FSK_SOURCE<br>SDG_ATTR_FSK_INTERNAL_RATE  |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_FSK\_INTERNAL\_RATE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the FSK(Frequency Shift Keying) rate.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_FSK_INTERNAL_RATE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,50KHZ]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_FSK_FREQUENCY<br>SDG_ATTR_FSK_SOURCE  |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PSK\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator applies phase shift keying modulation to the output signal.   |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</p> <p>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PSK_ENABLED macro.<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | VI_TRUE means PSK is on<br>VI_FALSE means PSK is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_PSK\_SOURCE

**Description** This channel-based attribute set the PSK modulation source to INTERNAL or EXTERNAL.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_PSK\_SOURCE macro

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

**Value Range**

| Type | Discrete Value       | Value |
|------|----------------------|-------|
| INT  | SDG_VAL_PSK_INTERNAL | 0     |
| EXT  | SDG_VAL_PSK_EXTERNAL | 1     |

**Related Attribute** SDG\_ATTR\_PSK\_INTERNAL\_RATE  
SDG\_ATTR\_PSK\_PHASE  
SDG\_ATTR\_PSK\_POLARITY

**High Level Functions** None

## SDG\_ATTR\_PSK\_POLARITY

**Description** Select the positive or negative polarity of the modulating waveform to control the phase output.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_PSK\_POLARITY macro

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

**Value Range**

| Type | Discrete Value                | Value |
|------|-------------------------------|-------|
| POS  | SDG_VAL_PSK_POLARITY_POSITIVE | 0     |
| NEG  | SDG_VAL_PSK_POLARITY_NEGATIVE | 1     |

**Related Attribute** SDG\_ATTR\_PSK\_INTERNAL\_RATE  
SDG\_ATTR\_PSK\_PHASE  
SDG\_ATTR\_PSK\_SOURCE

**High Level Functions** None

## SDG\_ATTR\_PSK\_PHASE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the phase of the PSK modulating waveform.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PSK_PHASE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | The value range is [-360° ,360° ]  |
| <b>Related Attribute</b>        | SDG_ATTR_PSK_INTERNAL_RATE<br>SDG_ATTR_PSK_POLARITY<br>SDG_ATTR_PSK_SOURCE   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PSK\_INTERNAL\_RATE

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the PSK(Phase Shift Keying) rate.  |
| <b>Data Type</b>                | ViBoolean  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PSK_INTERNAL_RATE macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,20KHZ]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_PSK_PHASE<br>SDG_ATTR_PSK_POLARITY<br>SDG_ATTR_PSK_SOURCE   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PWM\_ENABLED

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | This channel-based attribute specifies whether the function generator applies pulse width modulation to the output signal.  |
| <b>Data Type</b>                | ViBoolean   |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <code>sdg_SetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean value)</code><br><br><code>sdg_GetAttributeViBoolean(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViBoolean *value)</code><br><br><b>Notes:</b><br><b>vi</b> is the instrument handle.<br><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br><b>attributeld</b> is SDG_ATTR_PWM_ENABLED macro.<br><b>value</b> is used to store or set the value of function represented by <b>attributeld</b> . |
| <b>Value Range</b>              | VI_TRUE means PWM is on<br>VI_FALSE means PWM is off  |
| <b>Related Attribute</b>        | None  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_PWM\_SOURCE

**Description** This channel-based attribute set the PWM modulation source to internal or external.

**Data Type** Vilnt32

**Access** R/W

**Common Control Functions** `sdg_SetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 value)`

`sdg_GetAttributeVilnt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, Vilnt32 *value)`

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is SDG\_ATTR\_PWM\_SOURCE macro

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

**Value Range**

| Type | Discrete Value       | Value |
|------|----------------------|-------|
| INT  | SDG_VAL_PWM_INTERNAL | 0     |
| EXT  | SDG_VAL_PWM_EXTERNAL | 1     |

**Related Attribute** None

**High Level Functions** None

## SDG\_ATTR\_PWM\_DEVIATION\_WIDTH

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Set the pulse width deviation of PWM.  |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PWM_DEVIATION_WIDTH macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [0s,500s]</p> <p><b>Notes:</b><br/>SDG1000X range is [0s,17.6ns]<br/>SDG2000X range is [0s,500s]<br/>SDG6000X range is [0s,500s]<br/>SDG7000A range is [0s,500s]</p>   |
| <b>Related Attribute</b>        | None   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PWM\_INTERNAL\_WAVEFORM

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | Select the PWM modulating waveform.  |
| <b>Data Type</b>                | ViInt32  |
| <b>Access</b>                   | R/W  |
| <b>Common Control Functions</b> | <pre>sdg_SetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 value)</pre> <pre>sdg_GetAttributeViInt32(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViInt32 *value)</pre> |

**Notes:**

**vi** is the instrument handle.

**repCapName** is channelName(one of the following analog inputs: CHAN1, CHAN2).

**attributeld** is

SDG\_ATTR\_PWM\_INTERNAL\_WAVEFORM macro

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

**Value Range**

| Type | Discrete Value                | Value |
|------|-------------------------------|-------|
| SIN  | SDG_VAL_PWM_INTERNAL_SINE     | 0     |
| SQU  | SDG_VAL_PWM_INTERNAL_SQUARE   | 1     |
| TRI  | SDG_VAL_PWM_INTERNAL_TRIANGLE | 2     |
| RAMP | SDG_VAL_PWM_INTERNAL_RAMP     | 3     |
| DRAM | SDG_VAL_PWM_INTERNAL_NRAMP    | 4     |
| NOIS | SDG_VAL_PWM_INTERNAL_NOISE    | 5     |
| USER | SDG_VAL_PWM_INTERNAL_USER     | 6     |

|                             |                                 |
|-----------------------------|---------------------------------|
| <b>Related Attribute</b>    | SDG_ATTR_PWM_INTERNAL_FREQUENCY |
| <b>High Level Functions</b> | None                            |

## SDG\_ATTR\_PWM\_INTERNAL\_FREQUENCY

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Set the frequency of the PWM modulating waveform.   |
| <b>Data Type</b>                | ViReal64  |
| <b>Access</b>                   | R/W   |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is<br/>SDG_ATTR_PWM_INTERNAL_FREQUENCY macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1mHZ,2MHZ]</p> <p><b>Notes:</b><br/>SDG1000X range is [1mHZ,20kHz]<br/>SDG2000X range is [1mHZ,1MHZ]<br/>SDG6000X range is [1mHZ,1MHZ]<br/>SDG7000A range is [1mHZ, 2MHZ]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_PWM_INTERNAL_WAVEFORM  |
| <b>High Level Functions</b>     | None  |

## SDG\_ATTR\_PULSE\_WIDTH

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used to set the width of the pulse wave.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R  |
| <b>Common Control Functions</b> | <p><code>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</code></p> <p><code>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</code></p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PULSE_WIDTH macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [1ns,1Ms]</p> <p><b>Notes:</b><br/>SDG1000X range is [32.6ns,47.4ns]<br/>SDG2000X range is [16.3ns,1Ms]<br/>SDG6000X range is [3.3ns,1Ms]<br/>SDG7000A range is [1ns,1Ms]</p> <p>The maximum pulse width is limited by the frequency setting, and the minimum pulse width has nothing to do with the frequency</p>   |
| <b>Related Attribute</b>        | SDG_ATTR_PULSE_DELAY   |
| <b>High Level Functions</b>     | None   |

## SDG\_ATTR\_PULSE\_DELAY

|                                 |  |
|---------------------------------|--|
| <b>Description</b>              | This attribute is used to set the delay of the pulse wave.   |
| <b>Data Type</b>                | ViReal64   |
| <b>Access</b>                   | R  |
| <b>Common Control Functions</b> | <p>sdg_SetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 value)</p> <p>sdg_GetAttributeViReal64(ViSession vi, ViConstString repCapName, ViAttr attributeld, ViReal64 *value)</p> <p><b>Notes:</b><br/><b>vi</b> is the instrument handle.<br/><b>repCapName</b> is channelName(one of the following analog inputs: CHAN1, CHAN2).<br/><b>attributeld</b> is SDG_ATTR_PULSE_DELAY macro<br/><b>value</b> is used to store or set the value of function represented by <b>attributeld</b>.</p> |
| <b>Value Range</b>              | <p>The value range is [-1Ms,1Ms]</p> <p><b>Notes:</b><br/>SDG1000X range is [-80ns,80ns]<br/>SDG2000X range is [-999999.999s,999999.999s]<br/>SDG6000X range is [-999999.999s,999999.999s]<br/>SDG7000A range is [-1Ms,1Ms]</p>  |
| <b>Related Attribute</b>        | SDG_ATTR_PULSE_WIDTH   |
| <b>High Level Functions</b>     | None   |

## High Level Functions

Some high level functions are available to set multiple attributes.

- **sdg\_InitWithOptions(ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, ViConstString optionString, 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 sets to the InitWithOptions function of the instrument driver. It includes settings for Simulate, RangeCheck, QueryInstrStatus and Cache |
| *newVi   | Instrument handle  |
| <b>Example:</b><br><pre>sdg_InitWithOptions("USB0::0xF4EC::0x1013::0123456789::INSTR", VI_TRUE, VI_FALSE, "Simulate=0,RangeCheck=1,QueryInstrStatus=0,Cache=0", &amp;session);</pre> |  |

**Notes:** Siglent's driver 1.0 does not support simulation, cache, range check and querying instrument status. Therefore, it is only useful when connecting to an actual instrument. You should initiate the instrument by calling `sdg_InitWithOptions()` before using it, and input the parameters we suggested except argument `resourceName`.

- **sdg\_close (ViSession vi)**

This function closes the instrument.

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

|                                   |                   |
|-----------------------------------|-------------------|
| vi                                | Instrument handle |
| <b>Example:</b><br>sdg_close(vi); |                   |

**Notes:** This function must unlock the session before calling Ivi\_Dispose.

- **sdg\_ConfigureOperationMode (ViSession vi, ViConstString channelName, ViInt32 outputMode)**

This function configures the operation mode of the function generator. The operation mode determines how the function generator produces waveforms.

| Parameter   | Description                              |
|---|--|
| vi  | Instrument handle                        |
| channelName   | ChannelName: 1 or 2                      |
| outputMode  | operation mode of the function generator |
| <b>Example:</b><br>ConfigureOperationMode(vi,"1",0) |  |

- **sdg\_ConfigureOutputEnabled (ViSession vi, ViConstString channelName, ViBoolean enabled)**

This function configures whether the signal the function generator produces appears at the channel's output connector.

| Parameter   | Description                   |
|-------------|-------------------------------|
| vi          | Instrument handle             |
| channelName | ChannelName: 1 or 2           |
| enabled     | Turn output function on / off |

**Example:**

```
sdg_ConfigureOutputEnabled(vi,"1",0)
```

- **sdg\_ConfigureOutputImpedance (ViSession vi, ViConstString channelName, ViReal64 impedance)**

This function configures the output impedance for the channel you specify.

| Parameter                                 | Description         |
|---|---------------------|
| vi  | Instrument handle   |
| channelName                               | ChannelName: 1 or 2 |
| impedance                                 | Output impedance    |
| <b>Example:</b>                           |                     |
| sdg_ConfigureOutputImpedance (vi,"1",50); |                     |

- **sdg\_ConfigureOutputMode (ViSession vi, ViInt32 outputMode)**

This function configures the output mode of the function generator. The output mode determines the kind of waveform the function generator produces.

| Parameter                       | Description                           |
|---------------------------------|---------------------------------------|
| vi                              | Instrument handle                     |
| outputMode                      | output mode of the function generator |
| <b>Example:</b>                 |                                       |
| sdg_ConfigureOutputMode (vi,0); |                                       |

- **sdg\_ConfigureRefClockSource (ViSession vi, ViInt32 refClockSource)**

This function configures the function generator's reference clock source. The function generator uses the reference clock to derive frequencies and sample rates for signal generation.

| Parameter  | Description       |
|--|-------------------|
| vi   | Instrument handle |
| refClockSource   | Ref Clock Source  |
| <b>Example :</b><br>sdg_ConfigureRefClockSource (vi, 2); |                   |

- **sdg\_ConfigureStandardWaveform (ViSession vi, ViConstString channelName, ViInt32 waveform, ViReal64 amplitude, ViReal64 dcOffset, ViReal64 frequency, ViReal64 startPhase)**

This function sets the Standard Waveform.

| Parameter   | Description   |
|---|---|
| vi  | Instrument handle   |
| channelName   | ChannelName: 1 or 2   |
| waveform  | Output waveform type  |
| amplitude   | Output waveform amplitude   |
| dcOffset  | Output waveform dcOffset  |
| frequency   | Output waveform frequency   |
| startPhase  | the horizontal offset of the standard waveform the function generator produces. |
| <b>Example:</b><br>sdg_ConfigureStandardWaveform(vi, "1", SDG_VAL_WFM_SINE, 20, 0, 1000, 0) |   |

- **sdg\_ConfigureSquareWaveformSpec(ViSession vi, ViConstString channelName, ViReal64 dutyCycle)**

This function configures the Square Duty Cycle and Square Period of waveform. Period is paired with Frequency in ConfigureStandardWaveform and whichever one is executed last overrides the other.

| Parameter  | Description   |
|--|---|
| vi   | Instrument handle   |
| channelName  | ChannelName: 1 or 2   |
| dutyCycle  | The length of time the output voltage level remains high in a square waveform |
| <b>Example:</b><br>sdg_ConfigureSquareWaveformSpec (vi," 1",50); |   |

- **sdg\_ConfigureRampWaveformSpec(ViSession vi, ViConstString channelName, ViReal64 symmetry)**

This function configures the Ramp Symmetry of the function generator that affect ramp waveform generation.

| Parameter   | Description         |
|---|---------------------|
| vi  | Instrument handle   |
| channelName   | ChannelName: 1 or 2 |
| symmetry  | the Ramp Symmetry   |
| <b>Example:</b><br>sdg_ConfigureChanCharacteristics (vi,"1", 50); |                     |

- **sdg\_ConfigureArbWaveform (ViSession vi, ViConstString channelName, ViInt32 handle, ViReal64 gain, ViReal64 offset)**

This function configures the attributes of the function generator that affect arbitrary waveform generation. These attributes are the arbitrary waveform handle, gain, and offset.

| Description  | Description   |
|--|---|
| vi   | Instrument handle   |
| channelName  | ChannelName: 1 or 2   |
| handle   | attribute identifies which arbitrary waveform the function generator produces |
| gain   | similar to amplitude  |
| offset   | the value the function generator adds to the arbitrary waveform data          |
| <b>Example:</b><br><code>sdg_ConfigureArbWaveform (vi, "2", 10100, 3, 4);</code> |   |

➤ **sdg\_CreateArbWaveform(ViSession vi, Vilnt32 size, ViReal64 data[], Vilnt32 \*handle)**

This function creates an arbitrary waveform and returns a handle that identifies that waveform. You use the handles this function returns to specify what waveform to generate with the `sdg_ConfigureArbWaveform` function. You also use the handles this function returns to specify a sequence of arbitrary waveforms with the `sdg_CreateArbSequence` function. Use `sdg_ConfigChannel` function to specify a target channel for 2-channel instrument.

| Description | Description       |
|-------------|-------------------|
| vi          | Instrument handle |
| size        | Size of waveform  |
| data[]      | Data of waveform  |

|  |  |
|--|--|
| *handle  | a handle that identifies that waveform |
| <b>Example:</b>  |  |
| sdg_CreateArbWaveform(vi, 1e+3, 1e+3, SDG_VAL_WFM_SINE); |  |

- **sdg\_QueryArbWfmCapabilities (ViSession vi, ViInt32 \*MaxNumWfms, ViInt32 \*WfmQuantum, ViInt32 \*MinWfmSize, ViInt32 \*MaxWfmSize)**

This function returns the attributes of the function generator that are related to creating arbitrary waveforms. These attributes are the maximum number of waveforms, waveform quantum, minimum waveform size, and maximum waveform size

| Description  | Description                 |
|--|-----------------------------|
| vi   | Instrument handle           |
| *MaxNumWfms  | maximum number of waveforms |
| *WfmQuantum  | waveform quantum            |
| *MinWfmSize  | minimum waveform size       |
| *MaxWfmSize)   | maximum waveform size       |
| <b>Example:</b>  |                             |
| sdg_QueryArbWfmCapabilities (vi, 3, 4, 10010, 10015 ); |                             |

- **sdg\_ConfigureArbMode(ViSession vi, ViConstString channelName, ViInt32 arbMode)**

This function configures the arbitrary waveform output mode.

| Description | Description         |
|-------------|---------------------|
| vi          | Instrument handle   |
| channelName | ChannelName: 1 or 2 |
| arbMode     | Arb mode            |

**Example:**

```
sdg_ConfigureArbMode (vi, "1", 0);
```

- **sdg\_ConfigureArbFrequency (ViSession Vi, ViConstString ChannelName, ViReal64 Frequency)**

This function configures the arbitrary waveform frequency, which is the rate at which the function generator produces one cycle of an arbitrary waveform.

| Description                               | Description         |
|---|---------------------|
| vi  | Instrument handle   |
| ChannelName                               | ChannelName: 1 or 2 |
| Frequency                                 | set frequency value |
| <b>Example:</b>                           |                     |
| sdg_ConfigureArbFrequency (vi, "1", 1e+3) |                     |

- **sdg\_SendSoftwareTrigger (ViSession vi)**

This function can make the user send a software trigger to cause signal output to occur

| Description                   | Description       |
|-------------------------------|-------------------|
| vi                            | Instrument handle |
| <b>Example:</b>               |                   |
| sdg_SendSoftwareTrigger (vi); |                   |

- **sdg\_ConfigureBurstState (ViSession vi, ViConstString channelName, ViBoolean enabled)**
- This function configures the burst switch. The driver uses this value to set the AM Enabled attribute

| Description                       | Description         |
|-----------------------------------|---------------------|
| vi                                | Instrument handle   |
| channelName                       | ChannelName: 1 or 2 |
| <b>enabled</b>                    | Turn on the Burst   |
| <b>Example:</b>                   |                     |
| sdg_ConfigureBurstState(vi,"1",1) |                     |

➤ **sdg\_ConfigureBurstTrigSrc (ViSession vi, ViConstString channelName, Vilnt32 src)**

This function configures the burst trigger source. The driver uses this value to set the Burst Trigger Source attribute.

| Description                          | Description              |
|--------------------------------------|--------------------------|
| vi                                   | Instrument handle        |
| channelName                          | ChannelName: 1 or 2      |
| src                                  | The burst trigger source |
| <b>Example:</b>                      |                          |
| sdg_ConfigureBurstTrigSrc (vi,"1",3) |                          |

➤ **sdg\_ConfigureBurstCount (ViSession vi, ViConstString channelName, Vilnt32 count)**

This function configures the burst count. The driver uses this value to set the Burst Count attribute.

| Description     | Description          |
|-----------------|----------------------|
| vi              | Instrument handle    |
| channelName     | ChannelName: 1 or 2  |
| count           | the number of cycles |
| <b>Example:</b> |                      |

```
sdg_ConfigureBurstCount(vi,"1",3)
```

- **sdg\_ConfigureBurstGateEnabled (ViSession vi, ViConstString channelName, ViBoolean enabled)**

This function configures the Burst-GATE switch. The driver uses this value to set the Burst-GATE Enabled attribute

| Description                               | Description            |
|---|------------------------|
| vi  | Instrument handle      |
| channelName                               | ChannelName: 1 or 2    |
| enabled                                   | Turn on the Burst-GATE |
| <b>Example:</b>                           |                        |
| sdg_ConfigureBurstGateEnabled (vi,"1",0); |                        |

- **sdg\_ConfigureBurstNcycEnabled (ViSession vi, ViConstString channelName, ViBoolean enabled)**

This function configures the Burst-NCYC switch. The driver uses this value to set the Burst-NCYC Enabled attribute

| Description                               | Description            |
|---|------------------------|
| vi  | Instrument handle      |
| channelName                               | ChannelName: 1 or 2    |
| enabled                                   | Turn on the Burst-NCYC |
| <b>Example:</b>                           |                        |
| sdg_ConfigureBurstNcycEnabled (vi,"1",0); |                        |

- **sdg\_ConfigureBurstStartPhase (ViSession vi, ViConstString channelName, ViReal64 phase)**

This function configures the burst start phase. The driver uses this value to set the Burst Start phase attribute.

| Description                               | Description           |
|---|-----------------------|
| vi  | Instrument handle     |
| channelName                               | ChannelName: 1 or 2   |
| phase                                     | the burst start phase |
| <b>Example:</b>                           |                       |
| sdg_ConfigureBurstStartPhase (vi,"1",45); |                       |

- **sdg\_ConfigureBurstTrigOut (ViSession vi, ViConstString channelName, ViInt32 trigoutmod)**

This function configures mod of the burst trigger out. The driver uses this value to set the Burst Trigger Out attribute.

| Description                               | Description                  |
|---|------------------------------|
| vi  | Instrument handle            |
| channelName                               | ChannelName: 1 or 2          |
| trigoutmod                                | Mod of the burst trigger out |
| <b>Example:</b>                           |                              |
| sdg_ConfigureBurstGateEnabled (vi,"1",1); |                              |

- **sdg\_ConfigureAMEnabled (ViSession vi, ViConstString channelName, ViBoolean enabled);**

This function configures the am switch. The driver uses this value to set the AM Enabled attribute

| Description | Description       |
|-------------|-------------------|
| vi          | Instrument handle |

|   |                     |
|---|---------------------|
| channelName   | ChannelName: 1 or 2 |
| enabled   | Turn on the AM      |
| <b>Example:</b><br>sdg_ConfigureAMEnabled (vi,"1",0); |                     |

➤ **sdg\_ConfigureAMInternal (ViSession vi, ViReal64 Depth, ViInt32 Waveform, ViReal64 Frequency)**

This function configures the am internal. The driver uses this value to configure the AM Internal attribute.

| Description   | Description                        |
|---|------------------------------------|
| vi  | Instrument handle                  |
| Depth   | the extent of modulation           |
| Waveform  | standard waveform type             |
| Frequency   | frequency of the standard waveform |
| <b>Example:</b><br>sdg_ConfigureAMInternal(vi, 90, 2, 10) |                                    |

➤ **sdg\_ConfigureAMSource (ViSession vi, ViConstString channelName, ViInt32 Source)**

This function configures trigger source of the Am . The driver uses this value to set the Am Trigger Source attribute.

| Description  | Description         |
|--|---------------------|
| vi   | Instrument handle   |
| channelName  | ChannelName: 1 or 2 |
| Source   | trigger source      |
| <b>Example:</b><br>sdg_ConfigureAMSource (vi,"1",0); |                     |

- **sdg\_ConfigureFMEnabled(ViSession vi,ViConstString channelName,ViBoolean enabled)**

This function configures the FM switch.The driver uses this value to set the FM Enabled attribute

| Description   | Description         |
|---|---------------------|
| vi  | Instrument handle   |
| channelName   | ChannelName: 1 or 2 |
| enabled   | Turn on the FM      |
| <b>Example:</b><br>sdg_ConfigureFMEnabled (vi,"1",0); |                     |

- **sdg\_ConfigureFMInternal (ViSession vi, ViReal64 Dev, ViInt32 Waveform, ViReal64 Frequency)**

This function configures the fm internal.The driver uses this value to configure the FM Internal attribute

| Description  | Description   |
|--|---|
| vi   | Instrument handle                                   |
| Dev  | maximum frequency deviation the modulating waveform |
| Waveform   | standard waveform type                              |
| Frequency  | frequency of the standard waveform                  |
| <b>Example:</b><br>sdg_ConfigureFMInternal(vi, 10, 2, 10); |   |

- **sdg\_ConfigureFMSource (ViSession vi, ViConstString channelName, ViInt32 Source)**

This function configures the Burst-GATE switch. The driver uses this value to set the Burst-GATE Enabled attribute

| Description                      | Description         |
|----------------------------------|---------------------|
| vi                               | Instrument handle   |
| channelName                      | ChannelName: 1 or 2 |
| Source                           | Trigger source      |
| <b>Example:</b>                  |                     |
| sdg_ConfigureFMSource(vi,"2",0); |                     |

- **sdg\_ConfigureSweepState** (ViSession vi, ViConstString channelName, ViBoolean state)

This function configures state of sweep. The driver uses this value to set the Sweep State attribute

| Description                        | Description         |
|------------------------------------|---------------------|
| vi                                 | Instrument handle   |
| channelName                        | ChannelName: 1 or 2 |
| state                              | The state of Sweep  |
| <b>Example:</b>                    |                     |
| sdg_ConfigureSweepState(vi,"1",1); |                     |

- **sdg\_ConfigureSweepTime** (ViSession vi, ViConstString channelName, ViReal64 time)

This function configures time of sweep. The driver uses this value to set the Sweep Time attribute

| Description | Description       |
|-------------|-------------------|
| vi          | Instrument handle |

|  |                     |
|--|---------------------|
| channelName  | ChannelName: 1 or 2 |
| time   | time of sweep       |
| <b>Example:</b><br>sdg_ConfigureSweepTime(vi,"1",3); |                     |

- **sdg\_ConfigureSweepFrequencyStart** (ViSession vi, ViConstString channelName, ViReal64 freqstart)

This function configures start frequency of sweep. The driver uses this value to set the Sweep Start Frequency attribute

| Description  | Description           |
|--|-----------------------|
| vi   | Instrument handle     |
| channelName  | ChannelName: 1 or 2   |
| Freqstart  | Sweep start frequency |
| <b>Example:</b><br>sdg_ConfigureSweepFrequencyStart(vi,"1",600); |                       |

- **sdg\_ConfigureSweepFrequencyStop** (ViSession vi, ViConstString channelName, ViReal64 freqstop)

This function configures stop frequency of sweep. The driver uses this value to set the Sweep Stop Frequency attribute

| Description  | Description          |
|--|----------------------|
| vi   | Instrument handle    |
| channelName  | ChannelName: 1 or 2  |
| Freqstop   | Sweep stop frequency |
| <b>Example:</b><br>sdg_ConfigureSweepFrequencyStop(vi,"1",2000); |                      |

- **sdg\_ConfigureSweepTriggerSource (ViSession vi, ViConstString channelName, ViInt32 trigsrc)**

This function configures trigger source of sweep. The driver uses this value to set the Sweep Trigger Source attribute.

| Description                                | Description         |
|--|---------------------|
| vi   | Instrument handle   |
| channelName                                | ChannelName: 1 or 2 |
| trigsrc                                    | Trigger Source      |
| <b>Example:</b>                            |                     |
| sdg_ConfigureSweepTriggerSource(vi,"1",0); |                     |

- **sdg\_ConfigureSweepTriggerOut (ViSession vi, ViConstString channelName, ViBoolean state)**

This function configures trigger out of sweep. The driver uses this value to set the Sweep Trigger Out attribute.

| Description                             | Description                           |
|---|---------------------------------------|
| vi                                      | Instrument handle                     |
| channelName                             | ChannelName: 1 or 2                   |
| state                                   | edge type of the sweep trigger output |
| <b>Example:</b>                         |                                       |
| sdg_ConfigureSweepTriggerOut(vi,"1",1); |                                       |

- **sdg\_ConfigureSweepType (ViSession vi, ViConstString channelName, ViInt32 type)**

This function configures the type of sweep. The driver uses this value to set the Sweep Type attribute.

| Description | Description |
|-------------|-------------|
|-------------|-------------|

|   |                     |
|---|---------------------|
| vi  | Instrument handle   |
| channelName   | ChannelName: 1 or 2 |
| type  | type of the sweep   |
| <b>Example:</b><br><code>sdg_ConfigureSweepType(vi,"1",0);</code> |                     |

- **sdg\_ConfigureSweepDirection (ViSession vi, ViConstString channelName, ViInt32 dir)**

This function configures the direction of sweep. The driver uses this value to set the Sweep Direction attribute.

| Description  | Description                |
|--|----------------------------|
| vi   | Instrument handle          |
| channelName  | ChannelName: 1 or 2        |
| dir  | the direction of the sweep |
| <b>Example:</b><br><code>sdg_ConfigureSweepDirection(vi,"1",2);</code> |                            |

- **sdg\_ConfigureIqFrequencyCenter (ViSession vi, ViReal64 freqcenter)**

This function configures the center frequency of IQ. The driver uses this value to set the IQ Center Frequency attribute.

| Description  | Description                |
|--|----------------------------|
| vi   | Instrument handle          |
| frequency  | the center frequency of IQ |
| <b>Example:</b><br><code>sdg_ConfigureIqFrequencyCenter(vi,34000000);</code> |                            |

- **sdg\_ConfigureIqSampleRate (ViSession vi, ViReal64 samplerate)**

This function configures the sample rate of IQ. The driver uses this value to set the IQ Sample Rate attribute

| Description                           | Description           |
|---------------------------------------|-----------------------|
| vi                                    | Instrument handle     |
| samplerate                            | the sample rate of IQ |
| <b>Example:</b>                       |                       |
| sdg_ConfigureIqSampleRate(vi,600000); |                       |

➤ **sdg\_ConfigureIqSymbolRate (ViSession vi, ViReal64 symbolrate)**

This function configures the symbol rate of IQ. The driver uses this value to set the IQ Symbol Rate attribute.

| Description                           | Description           |
|---------------------------------------|-----------------------|
| vi                                    | Instrument handle     |
| symbolrate                            | the symbol rate of IQ |
| <b>Example:</b>                       |                       |
| sdg_ConfigureIqSymbolRate(vi,400000); |                       |

➤ **sdg\_ConfigureIqAmplitude (ViSession vi, ViReal64 amplitude)**

This function configures the amplitude of IQ. The driver uses this value to set the IQ Amplitude attribute

| Description                       | Description         |
|-----------------------------------|---------------------|
| vi                                | Instrument handle   |
| amplitude                         | the amplitude of IQ |
| <b>Example:</b>                   |                     |
| sdg_ConfigureIqAmplitude(vi,0.15) |                     |

➤ **sdg\_ConfigureIqTriggerSource (ViSession vi, ViInt32 trigsrc)**

This function configures the trigger source of IQ. The driver uses this value to set the IQ Trigger Source attribute.

| Description   | Description       |
|---|-------------------|
| vi  | Instrument handle |
| trigsrc   | trigger source    |
| <b>Example:</b><br><code>sdg_ConfigureIqTriggerSource(vi,0);</code> |                   |

➤ **sdg\_ConfigureIqAdjGain (ViSession vi, ViReal64 gain)**

This function configures the gain of adjustment of IQ. The driver uses this value to set the IQ Gain attribute

| Description   | Description                  |
|---|------------------------------|
| vi  | Instrument handle            |
| gain  | the gain of adjustment of IQ |
| <b>Example:</b><br><code>sdg_ConfigureIqAdjGain(vi,1);</code> |                              |

➤ **sdg\_ConfigureIqAdjIOffset (ViSession vi, ViReal64 offset)**

This function configures the I-offset of adjustment of IQ. The driver uses this value to set the IQ I-offset attribute.

| Description  | Description            |
|--|------------------------|
| vi   | Instrument handle      |
| offset   | I-offset of adjustment |
| <b>Example:</b><br><code>sdg_ConfigureIqAdjIOffset(vi,1);</code> |                        |

➤ **sdg\_ConfigureIqAdjQOffset (ViSession vi, ViReal64 offset)**

This function configures the Q-offest of adjustment of IQ.The driver uses this value to set the IQ Q-offset attribute.

| Description   | Description            |
|---|------------------------|
| vi  | Instrument handle      |
| offset  | Q-offest of adjustment |
| <b>Example:</b><br>sdg_ConfigureIqAdjQOffest(vi,1); |                        |

➤ **sdg\_ConfigureIqAdjSkew (ViSession vi, ViReal64 angle)**

This function configures the skew of adjustment of IQ.The driver uses this value to set the IQ Skew attribute

| Description                                      | Description        |
|--|--------------------|
| vi   | Instrument handle  |
| angle  | skew of adjustment |
| <b>Example:</b><br>sdg_ConfigureIqAdjSkew(vi,1); |                    |

➤ **sdg\_ConfigureIqWaveBuiltIn (ViSession vi, ViInt32 wave)**

This function configures the wave in the system-stored of IQ.The driver uses this value to set the IQ Built-in Wave attribute.

| Description   | Description                         |
|---|-------------------------------------|
| vi  | Instrument handle                   |
| wave  | the wave in the system-stored of IQ |
| <b>Example:</b><br>sdg_ConfigureIqWaveBuiltIn(vi,0) |                                     |

➤ **sdg\_ConfigureIqWaveUser (ViSession vi, ViString wavename)**

This function configures the user's wave of IQ. The driver uses this value to set the IQ

User's Wave attribute

| Description   | Description               |
|---|---------------------------|
| vi  | Instrument handle         |
| wavename  | The name of waveform file |
| <b>Example:</b>                                     |                           |
| sdg_ConfigureIqWaveUser(vi,xxx); //xxx is file name |                           |

- **sdg\_ConfigurePrbsBitRate (ViSession vi, ViConstString channelName , ViReal64 bitrate)**

This function configures the bit rate of PRBS.The driver uses this value to set the

PRBS Bit Rate attribute

| Description                            | Description                    |
|--|--------------------------------|
| vi                                     | Instrument handle              |
| channelName                            | ChannelName: 1 or 2            |
| bitrate                                | the bit rate attribute of PRBS |
| <b>Example:</b>                        |                                |
| sdg_ConfigurePrbsBitRate(vi,"1",1000); |                                |

- **sdg\_ConfigurePrbsAmplitude (ViSession vi, ViConstString channelName , ViReal64 amp)**

This function configures the amplitude of PRBS.The driver uses this value to set the

PRBS Amplitude attribute

| Description | Description         |
|-------------|---------------------|
| vi          | Instrument handle   |
| channelName | ChannelName: 1 or 2 |

|                                       |                                 |
|---------------------------------------|---------------------------------|
| amp                                   | the amplitude attribute of PRBS |
| <b>Example:</b>                       |                                 |
| sdg_ConfigurePrbsAmplitude(vi,"1",5); |                                 |

➤ **sdg\_ConfigurePrbsOffset (ViSession vi, ViConstString channelName , ViReal64 offset)**

This function configures the offset of PRBS. The driver uses this value to set the PRBS Offset attribute.

| Description                        | Description                  |
|------------------------------------|------------------------------|
| vi                                 | Instrument handle            |
| channelName                        | ChannelName: 1 or 2          |
| offset                             | the offset attribute of PRBS |
| <b>Example:</b>                    |                              |
| sdg_ConfigurePrbsOffset(vi,"1",1); |                              |

➤ **sdg\_ConfigurePrbsLength (ViSession vi, ViConstString channelName , ViInt32 length)**

This function configures the length of PRBS. The driver uses this value to set the PRBS Length attribute.

| Description                        | Description                  |
|------------------------------------|------------------------------|
| vi                                 | Instrument handle            |
| channelName                        | ChannelName: 1 or 2          |
| length                             | the length attribute of PRBS |
| <b>Example:</b>                    |                              |
| sdg_ConfigurePrbsLength(vi,"1",3); |                              |

➤ **sdg\_ConfigurePrbsEdge (ViSession vi, ViConstString channelName , ViReal64**

**edge)**

This function configures the edge of PRBS. The driver uses this value to set the PRBS Edge attribute.

| Description  | Description         |
|--|---------------------|
| vi   | Instrument handle   |
| channelName  | ChannelName: 1 or 2 |
| edge   | Edge attribute      |
| <b>Example:</b><br><code>sdg_ConfigurePrbsEdge(vi,"1",0.000000001);</code> |                     |

- **sdg\_ConfigurePrbsDifferentialState (ViSession vi, ViConstString channelName , ViBoolean state)**

This function configures the differential state of PRBS. The driver uses this value to set the PRBS Differential State attribute.

| Description   | Description         |
|---|---------------------|
| vi  | Instrument handle   |
| channelName   | ChannelName: 1 or 2 |
| state   | differential state  |
| <b>Example:</b><br><code>sdg_ConfigurePrbsDifferentialState(vi,"1",0);</code> |                     |

- **sdg\_ConfigurePrbsLogicLevel (ViSession vi, ViConstString channelName , ViInt32 level)**

This function configures the logic level of PRBS. The driver uses this value to set the PRBS Logic Level attribute.

| Description | Description |
|-------------|-------------|
|-------------|-------------|

|  |                         |
|--|-------------------------|
| vi   | Instrument handle       |
| channelName  | ChannelName: 1 or 2     |
| level  | the logic level of PRBS |
| <b>Example:</b><br>sdg_ConfigurePrbsLogicLevel (vi,"1",0); |                         |

## IVI-C Driver Programming Example

The example is running in an environment where NI VISA 5.4, LabWindow/CVI 2017, and IVI Compliance Package 15.0 are installed.

### Using dynamic link library

```
#include <Windows.h>
#include <stdio.h>
#include <userint.h>
#include <utility.h>
#include "sdg.h"

#define SDG_EXAMPLE_INSTR_RES_ADDR
"USB0::0xF4EC::0x1101::0123456789::INSTR"
#define SDG_EXAMPLE_INIT_OPTION
"Simulate=0,RangeCheck=0,QueryInstrStatus=0,Cache=1"

ViSession SDG_vi;

#define BSWV_TYPE_NUM 7
static ViInt16 waveType[] = {SDG_VAL_WFM_SINE,
                             SDG_VAL_WFM_SQUARE,
                             SDG_VAL_WFM_RAMP_UP,
                             SDG_VAL_WFM_RAMP_DOWN,
                             SDG_VAL_WFM_PULSE,
                             SDG_VAL_WFM_NOISE,
                             SDG_VAL_WFM_DC,
                             };

typedef ViStatus _VI_FUNC (*setAttr)(ViSession vi,
                                     ViConstString channelName,
                                     ViAttr attribute, ViReal64 value);
typedef ViStatus _VI_FUNC (*getAttr)(ViSession vi,
                                     ViConstString channelName,
                                     ViAttr attribute, ViReal64 *value);
typedef ViStatus _VI_FUNC (*sdslnit)(ViRsrc resourceName,
```

```
ViBoolean IDQuery,ViBoolean resetDevice,
ViConstString optionString,ViSession *newVi);
```

```
int main (int argc, char *argv[])
{
    HINSTANCE hDLL = LoadLibrary("sdg.dll");    //显式调用 DLL 文件

    if (hDLL != NULL)
    {
        sdgInit sdg_InitWithOptions = (sdgInit)GetProcAddress(hDLL,"sdg_InitWithOptions");
        setAttr sdg_SetAttributeViReal64 =
(setAttr)GetProcAddress(hDLL,"sdg_SetAttributeViReal64");
        getAttr sdg_GetAttributeViReal64 =
(getAttr)GetProcAddress(hDLL,"sdg_GetAttributeViReal64");
        ViStatus error = VI_SUCCESS;
        ViInt32 i = 0;
        ViString strMsg = "The function generator is outputting an standard waveform."
            "\n\nPress <Enter> to continue";

        if (InitCVIRTE (0, argv, 0) == 0)
            return -1;
        sdg_InitWithOptions (SDG_EXAMPLE_INSTR_RES_ADDR, VI_TRUE, VI_FALSE,
SDG_EXAMPLE_INIT_OPTION, &SDG_vi);
        for(i = 0;i < BSWV_TYPE_NUM;i++)
        {
            sdg_ConfigureStandardWaveform(SDG_vi, "1", waveType[i], 20, 0, 1000, 0);
            sdg_ConfigureOutputEnabled (SDG_vi, "1", VI_TRUE);
            Delay(1);
        }
        sdg_ConfigureOutputEnabled (SDG_vi, "1", VI_FALSE);
        for(i = 0;i < BSWV_TYPE_NUM;i++)
        {
            sdg_ConfigureStandardWaveform(SDG_vi, "2", waveType[i], 20, 0, 1000, 0);
            sdg_ConfigureOutputEnabled (SDG_vi, "2", VI_TRUE);
            Delay(1);
        }
        sdg_ConfigureOutputEnabled (SDG_vi, "2", VI_FALSE);
        /* Display a message as the finish */
        MessagePopup ("Message", strMsg);
    }
    Error:
        if(VI_SUCCESS != error)
```

```
{
    ViChar szErrMsg[2048];
    /* Retrieve the error and display it */
    sdg_GetError(SDG_vi, &error, 2048, szErrMsg);
    MessagePopup("Error!", szErrMsg);
}
if(SDG_vi)
    sdg_close(SDG_vi);
return 0;
//FreeLibrary(hDLL)
}
}
```



## SDG Series

## Digital Oscilloscope

### 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](mailto:sales@siglent.com)  
Website: [int.siglent.com](http://int.siglent.com)

#### USA:

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@siglent.com](mailto:info@siglent.com)  
Website: [www.siglentna.com](http://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

Follow us on  
Facebook: [SiglentTech](#)

