### **NI-FGEN LabVIEW Reference**

July 2007, 370774E-01

This section describes the VIs and properties included with NI-FGEN that you can use to configure and operate your signal generator.

To comment on National Instruments documentation, refer to the <u>National</u> <u>Instruments Web site</u>.

© 2004—2007 National Instruments Corporation. All rights reserved.

### **NI-FGEN VI Reference**

Use the VIs and functions on the NI-FGEN palette to build the block diagram.

Click the icons for VI and function descriptions.



## niFgen Initialize

Performs the following initialization actions:

- Creates a new IVI instrument driver session.

- Opens a session to the specified device using the interface and address that you specify for **Resource Name**.

- If **ID Query** is set to TRUE, this VI queries the device ID and checks that the ID is valid for NI-FGEN.

- If **Reset Device** is set to TRUE, this VI resets the device to a known state.

- Sends initialization commands to set the device to the state necessary for the operation of NI-FGEN.

- Returns a session handle that you can use to identify the device in all subsequent NI-FGEN VI calls.



Resource Name specifies the instrument name, for example "PXI1Slot3" where "PXI1Slot3" is an instrument name assigned by Measurement & Automation Explorer. Syntax:

Configured in MAX Under	Valid Syntax
NI-DAQmx Devices	DAQmxDeviceName
NI-DAQmx Devices (alternate syntax)	DAQ::DAQmxDeviceName
Traditional NI-DAQ Devices	DAQ::DeviceNumber

**Caution** The alternate syntax allowed for NI-DAQmx devices is intended to allow application that may have originally been designed for a Traditional NI-DAQ device use an NI-DAQmx device. For example, if the application expects "DAQ::1", you can rename the device to"1" in MAX and pass in "DAQ::1" for the resource name.

If you use the "DAQ::n" syntax and an NI-DAQmx device name already exists with the same name, the NI-DAQmx device will be matched first.

You can also pass in the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical name identifies a particular virtual instrument.

Example resource names:

Resource Name	Description
DAQ::1	Traditional NI-DAQ device, device number = 1
DAQ::2	NI-DAQmx device, device name ="2"
myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
DAQ::myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
myLogicalName	IVI logical name or virtual instrument, name = "myLogicalName"

Traditional NI-DAQ and NI-DAQmx device names are not casesensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

Id Query specifies whether or not to verify that NI-FGEN supports the device you initialize.

Circumstances can arise where sending an ID query to the device is undesirable. When you set this parameter to FALSE, the VI initializes the device without performing an ID query.

- **Reset Device** specifies whether you want to reset the device during the initialization procedure.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## Configuration

Use the VIs located on the **NI-FGEN**»Configuration palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.

## **Configure Output**

Use the VIs located on the **NI-FGEN**»Configuration»Configure Output palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.

#### niFgen Output Enable

Configures the signal generator to generate a signal at the channel's output connector.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Channel Name specifies the channel this VI uses.
- **Output Enable** specifies the state of the output enable relay.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Configure Output Impedance

Configures the output impedance for the channel you specify.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Impedance specifies the impedance value that you want the signal generator to use. NI-FGEN sets Output Impedance to this value.

Units: ohms

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Configure Analog Filter

Configures the analog filter for the device. The analog filter setting can be applied in Arbitrary Waveform, Arbitrary Sequence, or Script output mode, or in Frequency List or Standard Function mode when used with user-defined functions.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Analog Filter Enable** specifies whether to enable the analog filter.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- Filter Correction Frequency specifies the filter correction frequency of the analog filter. On the NI 5411 and NI 5431, NI-FGEN adjusts signal amplitude to compensate for the filter's attenuation at that frequency. To disable amplitude correction, set Filter Correction Frequency to 0. For Standard Waveform Output, Filter Correction Frequency should typically be set to be the same as the frequency of the standard waveform.

Units: hertz (Hz)

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Configure Digital Filter

Configures the digital filter for the device. The digital filter setting can be applied in Arbitrary Waveform, Arbitrary Sequence, or Script output mode, or in Frequency List or Standard Function mode when used with user-defined functions.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Digital Filter Enable** specifies whether to enable the digital filter.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

132

**code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

" ); } } // End hiding from older browsers -->

## niFgen Configure Digital Patterning

Configures the state of the DIGITAL DATA & CONTROL/DIGITAL PATTERN connector.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Digital Patterning Enable** specifies whether or not pattern signals are present on the DIGITAL DATA & CONTROL/DIGITAL PATTERN connector.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

132

**code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Configure Clock**

Use the VIs located on the **NI-FGEN**»Configuration»Configure Clock palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.

#### niFgen Configure Sample Clock Source

Sets the source of the sample clock (update clock) of the signal generator.

Instrument Handle Out Source (Onboard Clock)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Source** specifies the sample clock source that you want the signal generator to use.

For a complete list of the sample clock sources available on your device, refer to the Device Routes tab in MAX.

Refer to the <u>Sample Clock Source</u> property for more information about setting this parameter.

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

TF

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

" ); } } // End hiding from older browsers -->

## niFgen Configure Clock Mode

Selects the clock mode for the signal generator.

Some signal generators allow you to switch the sample clock to High-Resolution or Automatic Sampling mode with this VI.

When in Divide-Down Sampling mode, NI-FGEN rounds the sample rate to a frequency that can be achieved by dividing down the board clock (sample clock timebase). However, in High-Resolution Sampling mode, you can set the sample rate to any value. In Automatic Sampling Mode, NI-FGEN selects the clock mode based on the sample rate, using Divide-Down Sampling when possible.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Waveform** or **Arbitrary Sequence** using <u>niFgen Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Clock Mode** sets the clock mode of the signal generator.

#### **Defined Values**

High-Resolution Sampling	Sample rate is generated by a high-resolution clock source.
Divide-Down Sampling	Sample rate is generated by dividing the source frequency.
Automatic	NI-FGEN selects between the divide-down and high-resolution modes

error in describes error conditions that occur before this VI or function runs.

**status** is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Set Sample Rate

Configures the sample rate, which determines the rate at which the signal generator produces arbitrary waveforms. When you configure the signal generator to produce an arbitrary sequence, this is the sample rate for all arbitrary waveforms in the sequence. If the update clock (sample clock) source is external, setting the sample rate informs NI-FGEN of the external rate.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Waveform** or **Arbitrary Sequence** using <u>niFgen Configure Output Mode VI</u> before calling this VI.

Instrument Handle	MI-FGEN
Sample Rate (Samp/s)	
error in (no error)	error out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Sample Rate specifies the sample rate at which you want the signal generator to output arbitrary waveforms. If you are using an external update clock (sample clock), this is the frequency of the external update clock.

Units: Samples per second (S/s)

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Configure Reference Clock

Configures the signal generator reference clock source and frequency. The signal generator uses the reference clock to tune the sample clock timebase of the signal generator so that the frequency stability and accuracy of the sample clock timebase matches that of the reference clock.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Source** specifies the reference clock source that you want the signal generator to use.

For example, when you set this parameter to "ClkIn," the signal generator uses the signal it receives at its Clk In front panel connector as the reference clock. For a complete list of the reference clock sources available on your device, refer to the Device Routes tab in MAX.

The reference clock will phase-lock with the signal generator's sample clock timebase to allow the frequency stability and accuracy of the sample clock timebase to match that of the reference clock.

Refer to the <u>Reference Clock Source</u> property for more information on setting this parameter.

**Reference Clock Frequency** specifies the reference clock frequency.

Units: Hertz (Hz)

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Adjust Sample Clock Relative Delay

Delays (or phase shifts) the sample clock, which delays the output of the module. Delaying the sample clock can be useful when lining up the output of multiple modules or when intentionally phase shifting the output relative to a fixed reference such as the PLL reference clock.

Adjustment time can be positive or negative, but it must be less than or equal to the sample clock period. The delay takes effect immediately after this VI is called. To delay an external sample clock, set the <u>Sample</u> <u>Clock Absolute Delay</u> property.



**Note** Calling this VI after calling NI-TClk Synchronize breaks synchronization.

Instrument Handle Out Adjustment Time

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Adjustment Time specifies the amount of time to adjust the sample clock delay.

Units: Seconds (s)

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

" ); } } // End hiding from older browsers -->

## niFgen Export Signal

Routes signals (clocks, triggers, and events) to the output terminal you specify.

Any routes created within a session persist after the session closes to prevent signal glitching. To unconfigure signal routes created in previous sessions, set **Reset Device** in the <u>niFgen Initialize VI</u> to TRUE or use <u>niFgen Reset Device</u>.

If you export a signal with this VI and commit the session, the signal is routed to the output terminal you specify.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Signal Identifier** specifies which instance of the selected signal to export.

Defined	Values
---------	--------

	Default (for non instance-based signals)
"ScriptTrigger0"	Script trigger 0
" ScriptTrigger1"	Script trigger 1
"ScriptTrigger2 "	Script trigger 2
" ScriptTrigger3"	Script trigger 3
"Marker0"	Marker 0
"Marker1"	Marker 1
"Marker2"	Marker 2
"Marker3"	Marker 3
" DataMarker0"	Data Marker 0
"DataMarker1"	Data Marker 1
"DataMarker2"	Data Marker 2
"DataMarker3"	Data Marker 3

**Signal** specifies the signal to export.

#### **Defined Values**

NIFGEN_VAL_ONBOARD_REFERENCE_CLOCK	The onboard 10MHz synchronization clock (PCI only)
NIFGEN_VAL_SYNC_OUT	The SYNC signal that normally appears on the SYNC_OUT front panel connector.
NIFGEN_VAL_START_TRIGGER	Start trigger
NIFGEN_VAL_MARKER_EVENT	Marker event
NIFGEN_VAL_SAMPLE_CLOCK_TIMEBASE	The clock from which the sample clock is derived
NIFGEN_VAL_SYNCHRONIZATION	A synchronization strobe used to guarantee absolute synchronization between two or more signal generators. (5404, 5411, 5431 only)

**Output Terminal** specifies the terminal to export the signal (clock, trigger, or event).

**Note:**The following defined values are examples of possible output terminals. For a complete list of the output terminals available on

your device, refer to the Device Routes tab in MAX.

#### **Defined Values**

	Do not export signal
"PFI0"	PFI line 0
"PFI1"	PFI line 1
"PFI4"	PFI line 4
"PFI5"	PFI line 5
"PXI_TRIG_0"	PXI or RTSI line 0
"PXI_TRIG_1"	PXI or RTSI line 1
"PXI_TRIG_2"	PXI or RTSI line 2
"PXI_TRIG_3"	PXI or RTSI line 3
"PXI_TRIG_4"	PXI or RTSI line 4
"PXI_TRIG_5"	PXI or RTSI line 5
"PXI_TRIG_6"	PXI or RTSI line 6
"PXI_TRIG_7"	PXI or RTSI line 7
"DDC_ClkOut"	Clock out from DDC connector
"PXI_STAR"	PXI Star line

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.

**error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# Configuration

Use the VI located on the **NI-FGEN»5404 Routing** palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.


## niFgen Route Signal Out

Routes various signals in the signal generator to the RTSI lines or front panel terminals.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You can clear a previously routed signal by routing NONE to the destination terminal.

Not all signal routes established by calling niFgen Route Signal Out VI are released when the NI-FGEN session is closed by <u>niFgen Close VI</u>.

Routes To	NI-5401/5411/5431	Other
Front Panel	Remains connected.	Remains connected.
RTSI/PXI Backplane	Remains connected.	Disconnected.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Route Signal From** specifies the source of the signal to route.
- **Route Signal To** specifies the destination of the signal to route.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

**source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Configure Trigger & Synchronization**

Use the VIs located on the **NI-FGEN**»**Configuration**»**Configure Trigger & Synchronization** palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.

## niFgen Configure Trigger Mode

Sets the trigger mode.

Supported trigger modes are Single Trigger, Continuous Trigger, Stepped Trigger, and Burst Trigger. Refer to **Devices**»<**your** 

**device>»Triggering»Triggering Mode** in the *NI Signal Generators Help* for descriptions of the signal generator's specific behavior for supported trigger modes.



**Notes** The signal generator must be in the Generating state when you call this VI.

In Frequency List output mode, Stepped Trigger mode is the same as Burst Trigger mode.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Trigger Mode** specifies the trigger mode.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument

session to the next VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Configure Trigger (poly)

Configures the trigger source. The signal generator responds to a trigger according to the current trigger mode.



**Note** The signal generator must not be in the Generating state when you call this VI.

•

Use the pull-down menu to select an instance of this VI.

Select an instance

#### niFgen Configure Digital Edge Start Trigger

Configures the Start trigger for edge triggering.

Edge (Rising)
Instrument Handle
Source
error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Source** specifies which trigger source the signal generator uses.
- edge specifies the edge to detect. You can choose **Rising Edge** or **Falling Edge**.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Configure Software Edge Start Trigger

Configures the Start trigger for software triggering. Use the niFgen Send Software Edge Triggering VI to assert the trigger condition.

Instrument Handle And Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

#### niFgen Disable Start Trigger

Disables the Start trigger.

Instrument Handle And Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the

error.

#### niFgen Configure Digital Edge Script Trigger

Configures a Script trigger for edge triggering.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Source** specifies which trigger source the signal generator uses.
- edge specifies the edge to detect. You can choose **Rising Edge** or **Falling Edge**.
- **Trigger ID (Script Trigger 0)** specifies the Script Trigger used for triggering.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Configure Digital Level Script Trigger

Configures a Script trigger for level triggering.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Source** specifies which trigger source the signal generator uses.
- **Trigger When (High)** specifies the active level for the desired trigger.
- **Trigger ID (Script Trigger 0)** specifies the Script Trigger used for triggering.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Configure Software Edge Script Trigger

Configures a Script trigger for software triggering. Use the niFgen Send Software Edge Triggering VI to assert the trigger condition.

Instrument Handle Out Trigger ID (Script Trigger 0)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Trigger ID (Script Trigger 0) specifies the Script Trigger used for triggering.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Disable Script Trigger

Disables the Script trigger.

Instrument Handle And HIFGEN Trigger ID (Script Trigger 0)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Trigger ID (Script Trigger 0) specifies the Script Trigger used for triggering.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Configure Synchronization

Sets the signal generator to receive a synchronization signal to synchronize two or more NI 5401/5411/5431 signal generators. One signal generator should route a SYNC signal to a RTSI line by calling niFgen Export Signal VI (niFgen Route Signal Out VI for the NI 5404), and other signal generators should receive the signal by calling the niFgen Configure Synchronization VI.



**Notes** The signal generator must not be in the Generating state when you call this VI.

Only the NI 5401/5411/5431 signal generators require this VI to be called for proper synchronization.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Channel Name specifies the channel this VI uses.
- Synchronization Source specifies the source of the synchronization signal you want to use.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

**Instrument Handle Out** passes a reference to your instrument session to the next VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Send Software Edge Trigger

Sends a command to trigger the signal generator. This VI can act as an override for an external edge trigger.



**Note** This VI does not override external digital edge triggers of the NI 5401/5411/5431.

Instrument Handle Andrew Instrument Handle Out Trigger (Start Trigger) Trigger ID (None)		
1701	<b>Instrument Handle</b> identifies your instrument session. <b>Instrument Handle</b> was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u> .	
132	Trigger specifies the type of software trigger to send.	
abc	<b>Trigger ID</b> specifies the Script Trigger to use for triggering.	
(200) (200)	error in describes error conditions that occur before this VI or function runs.	
	<b>status</b> is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.	
	<b>code</b> is the error or warning code. The default is 0. If <b>status</b> is TRUE, <b>code</b> is a negative error code. If <b>status</b> is FALSE, <b>code</b> is 0 or a warning code.	
	<b>source</b> identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.	
▶I/0	<b>Instrument Handle Out</b> passes a reference to your instrument session to the next VI.	
	error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.	

(checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Configure Onboard Signal Processing**

Use the VIs located on the **NI-FGEN**»**Configuration**»**Configure Onboard Signal Processing** palette to configure custom FIR filter coefficients and get FIR filter coefficients.

Click the icons for VI and function descriptions.

### niFgen Configure Custom FIR Filter Coefficients

Sets the FIR filter coefficients used by the onboard signal processing block. The values are coerced to the closest settings achievable by the signal generator.

Refer to **Devices»NI 5441»Theory of Operation»Onboard Signal Processing»Components»FIR Filter** in the *NI Signal Generators Help* for more information about FIR filter coefficients.



**Note** The signal generator must not be in the Generating state when you call this VI.

Instrument Handle Out Channel Name FIR Filter Coefficients

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **FIR Filter Coefficients** specifies the array of data the onboard signal processor uses for the FIR filter coefficients. For the NI 5441, you must wire a symmetric array of 95 coefficients to this parameter.

The coefficients should range between -1.00 and +1.00.

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Get FIR Filter Coefficients

Returns the FIR filter coefficients used by the onboard signal processing block. These coefficients are determined by NI-FGEN and based on the FIR filter type and corresponding property (Alpha, Passband, BT) unless you are using the custom filter. If you are using a custom filter, the coefficients returned are those set with the niFgen Configure Custom FIR Filter Coefficients VI coerced to the quantized values used by the device.

Refer to **Devices»NI 5441»Theory of Operation»Onboard Signal Processing»Components»FIR Filter** in the *NI Signal Generators Help* for more information about FIR filter coefficients.

Instrument Handle Channel Name FIR Filter Coefficients

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **FIR Filter Coefficients** returns the FIR filter coefficients that the onboard signal processor is using as an array of data. For the NI 5441, the array contains 95 coefficients coerced to the quantized

values used by the device.

The coefficients range between -1.00 and +1.00.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Waveform Control**

Use the VIs located on the **NI-FGEN**»Waveform Control palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



## niFgen Configure Output Mode

Configures the output mode of the signal generator. The output mode determines how the signal generator produces waveforms.

For example, you can select to output a standard waveform, an arbitrary waveform, or a sequence of arbitrary waveforms.



**Note** The signal generator must not be in the Generating state when you call this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Output Mode** specifies the output mode that you want the signal generator to use. The value you specify determines which VIs and properties you can use to configure the waveform the signal generator produces.

Refer to the <u>Output Mode</u> property for more information about setting this parameter.

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Standard Waveform**

Use the VIs located on the **NI-FGEN**»Waveform Control»Standard Waveform palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



### niFgen Configure Standard Waveform

Configures the properties of the signal generator that affect standard waveform generation. These properties are the waveform, amplitude, DC offset, frequency, and start phase.

**Notes**: You must set Output Mode to Standard Function using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Waveform specifies the standard waveform that you want the signal generator to produce. NI-FGEN sets Waveform to this value.
- **Frequency** specifies the frequency of the standard waveform that you want the signal generator to produce. NI-FGEN sets Frequency to this value (in hertz).
  - Note This parameter does not affect signal generator behavior when you set **Waveform** to DC.

Frequency ranges vary from device to device.

Amplitude specifies the amplitude of the standard waveform that you want the signal generator to produce. This value is the amplitude at the output terminal.

For example, to produce a waveform ranging from -5.00 V to +5.00 V, set **Amplitude** to 10.00 V. Units:  $V_{pk-pk}$ 



**Note** This parameter does not affect signal generator behavior when you set **Waveform Type** to DC.

- **Channel Name** specifies the channel this VI uses.
- **Start Phase** specifies the horizontal offset of the standard

waveform that you want the signal generator to produce. Specify this parameter in degrees of one waveform cycle. NI-FGEN sets Start Phase to this value (measured in degrees of one cycle). A start phase of 180 degrees means output generation begins halfway through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle and is therefore identical to a start phase of 0 degrees.



**Note** This parameter does not affect signal generator behavior when you set the waveform parameter to DC.

**DC Offset** specifies the DC offset of the standard waveform that you want the signal generator to produce. This value is the offset at the output terminal. The value is the offset from ground to the center of the waveform you specify with the Waveform parameter.

For example, to configure a waveform with an amplitude of 10.00 V to range from 0.00 V to +10.00 V, set the **DC Offset** to 5.00 V. Units: volts

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

TF

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
## niFgen Define User Standard Waveform

Defines a user waveform for either standard function output or frequency list output.

To select the waveform, set **Waveform** of <u>niFgen Configure Standard</u> <u>Waveform VI</u> or <u>niFgen Create Frequency List VI</u> to User.

The waveform data must be scaled between -1.0 and 1.0. Use the **Amplitude** parameter to generate different output voltages.

Note You must set Output Mode to Standard Function or Frequency List using <u>niFgen Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Size** specifies the size of the waveform in samples.
- **Waveform Data Array** specifies the array of data you want to load into the new waveform.

You must normalize the data points in the array to be between -1.00 and +1.00.

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Clear User Standard Waveform

Clears the user-defined waveform created by <u>niFgen Define User</u> <u>Standard Waveform VI</u>.

Instrument Handle Out Channel Name error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# **Frequency List**

Use the VIs located on the **NI-FGEN**»Waveform Control»Frequency List palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.

🔄 Frequency List 💦 📃 🔲 🔀		
🗘 🔍 Search		

# niFgen Query Freq List Capabilities

Returns the properties of the signal generator that are related to creating frequency lists. These properties are the maximum number of frequency lists, minimum frequency list length, maximum frequency list length, minimum frequency list duration, maximum frequency list duration, and frequency list duration quantum.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Maximum Number of Freq Lists returns the maximum number of frequency lists that the signal generator allows.
- **Frequency List Duration Quantum** returns the quantum of which all durations must be a multiple in a frequency list.
- Minimum Frequency List Duration returns the minimum duration that the signal generator allows in a step of a frequency list.

- Maximum Frequency List Duration returns the maximum duration that the signal generator allows in a step of a frequency list.
- Minimum Frequency List Length returns the minimum number of steps that the signal generator allows in a frequency list.
- Maximum Frequency List Length returns the maximum number of steps that the signal generator allows in a frequency list.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Create Frequency List

Creates a frequency list from an array of frequencies and an array of durations. The two arrays should have the same number of elements. The VI returns a handle that identifies the frequency list. You can pass this handle to <u>niFgen Configure Frequency List VI</u> to specify what frequency list you want the signal generator to produce.



**Note** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to Frequency List using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.

A frequency list consists of a list of frequencies and durations. The signal generator generates each frequency for the given amount of time and then moves on to the next frequency. When the end of the list is reached, the signal generator starts over at the beginning of the list.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Frequency List Length if specified, controls the number of steps in the frequency list. The value must be between the minimum and maximum frequency list lengths that the signal generator allows. You can obtain the minimum and maximum frequency list lengths from the Minimum Frequency List Length and Maximum Frequency List Length parameters in <u>niFgen Query Freq List</u> <u>Capabilities VI</u>.

The **Frequency Array** and the **Duration Array** parameters must each be at least as long as the frequency list length if one is specified.

**Waveform** specifies the standard waveform that you want the signal generator to produce. NI-FGEN sets Waveform to this

value.

Frequency Array specifies the array of frequencies to form the frequency list. The array must have at least as many elements as the value you specify in the Frequency List Length parameter. Each Frequency Array element has a corresponding Duration Array element that indicates how long that frequency is repeated.

Units: Hz

Duration Array specifies the array of durations to form the frequency list. The array must have at least as many elements as the value that you specify in the Frequency List Length parameter. Each Duration Array element has a corresponding Frequency Array element and indicates how long in seconds to generate the corresponding frequency.

Units: Seconds

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Frequency List Handle** specifies the handle of the frequency list you want the signal generator to produce. You can create multiple frequency lists using <u>niFgen Create Frequency List VI</u>. niFgen Create Frequency List returns a handle that you use to identify each list.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Configure Frequency List

Configures the properties of the signal generator that affect frequency list generation. Selects the frequency list to produce and sets the amplitude, DC offset, and start phase.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Frequency List** using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **DC Offset** specifies the DC offset of the standard waveform that you want the signal generator to produce. This value is the offset at the output terminal. The value is the offset from ground to the center of the waveform you specify with the Waveform parameter.

For example, to configure a waveform with an amplitude of 10.00 V to range from 0.00 V to +10.00 V, set the **DC Offset** to 5.00 V. Units: volts

- **Channel Name** specifies the channel this VI uses.
- **Frequency List Handle** specifies the handle of the frequency list you want the signal generator to produce. You can create multiple frequency lists using <u>niFgen Create Frequency List VI</u>. niFgen Create Frequency List returns a handle that you use to identify each list.
- Amplitude specifies the amplitude of the standard waveform that you want the signal generator to produce. This value is the amplitude at the output terminal.

For example, to produce a waveform ranging from -5.00 V to +5.00

V, set Amplitude to 10.00 V. Units: V<sub>pk-pk</sub>



**Note** This parameter does not affect signal generator behavior when you set **Waveform Type** to DC.

Start Phase specifies the horizontal offset of the standard waveform that you want the signal generator to produce. Specify this parameter in degrees of one waveform cycle. NI-FGEN sets Start Phase to this value (measured in degrees of one cycle). A start phase of 180 degrees means output generation begins halfway through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle and is therefore identical to a start phase of 0 degrees.

Note This parameter does not affect signal generator behavior when you set the waveform parameter to DC.

error in describes error conditions that occur before this VI or function runs.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code**

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Clear Frequency List

Removes a previously created frequency list from the signal generator memory and invalidates the **Frequency List Handle**.



**Note** The signal generator must not be in the Generating state when you call this VI.

Instrument Handle	MI-FGEN	<sup>•</sup> Instrument Handle Out
Frequency List Handle	- 12	
error in (no error)	3W	error out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Frequency List Handle** specifies the handle of the frequency list you want the signal generator to remove. You can create multiple frequency lists using <u>niFgen Create Frequency List VI</u>. niFgen Create Frequency List returns a handle that you use to identify each list.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Commit

Causes a transition to the Committed state. This VI verifies property values, reserves the device, and commits the property values to the device. If the property values are all valid, NI-FGEN sets the device hardware configuration to match the session configuration. This VI does not support the NI 5401/5404/5411/5431 signal generators.

In the Committed state, waveforms, scripts, and sequences can be loaded into memory. If any properties are changed, NI-FGEN implicitly transitions back to the Idle state. This VI has no effect if the device is already in the Committed or Generating states, and returns a successful status value.

Calling this VI before <u>niFgen Initiate Generation VI</u> is optional but has several benefits:

- Routes are committed, so signals are exported or imported.
- Any reference clock and external clock circuits are phase-locked.
- A subsequent <u>niFgen Initiate Generation VI</u> can run faster because the device is already configured.

Instrument Handle Andrew Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Initiate Generation

Initiates signal generation. If you want to abort signal output, call <u>niFgen</u> <u>Abort Generation VI</u>. Call this VI to cause the signal generator to produce a signal again.

Instrument Handle And HI-FGEN Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Abort Generation

Aborts any previously initiated signal generation. Call <u>niFgen Initiate</u> <u>Generation</u> VI to initiate signal generation.

Instrument Handle ------- Instrument Handle Out error in (no error) ------- error out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

# **Arbitrary Waveform**

Use the VIs located on the **NI-FGEN**»Waveform Control»Arbitrary Waveform palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



## niFgen Query Arb Waveform Capabilities

Returns the attributes of the signal 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.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Minimum Waveform Size returns the minimum number of points that the signal generator allows in a waveform. For some signal generators, you may need to supply a larger waveform than the value specified by this parameter. Refer to the *NI Signal Generators Help* for a table of minimum waveform sizes.
- Maximum Waveform Size returns the maximum number of points that the signal generator allows in a waveform. On some signal generators this may vary with remaining onboard memory.

- Waveform Quantum returns the size (number of points) of each waveform must be a multiple of a constant quantum value. This parameter obtains the quantum value that the signal generator uses. For example, if this output returns a value of 8, all waveform sizes must be a multiple of 8.
- Maximum Number of Waveforms returns the maximum number of arbitrary waveforms that the signal generator allows. On some signal generators this may vary with remaining onboard memory.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Create Waveform (poly)

Takes the data provided and creates an onboard waveform of the same size for use in Arbitrary Waveform or Arbitrary Sequence output mode.

The **Waveform Handle** returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.



**Notes** The signal generator must not be in the Generating state when you call any instance of this polymorphic VI.

You must set **Output Mode** to **Arbitrary Waveform** or **Arbitrary Sequence** using <u>niFgen Configure Output Mode VI</u> before calling this VI.

•

Use the pull-down menu to select an instance of this VI.

Select an instance

### niFgen Create Waveform (I16)

Creates an onboard waveform from I16 data for use in Arbitrary Waveform mode or Arbitrary Sequence mode. The waveform handle returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

This VI can also create an onboard waveform from complex data for use on devices with the <u>OSP Enabled</u> property set to **True** and the <u>Data</u> <u>Processing Mode</u> property set to **Complex**. To write complex data, you must interleave the IQ pairs.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Data Array** specifies the array of data you want to use for the new arbitrary waveform.

You must normalize the data points in the array to be between -32768 and 32767 inclusive.

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument

session to the next VI.

- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Create Waveform (WDT)

Creates an onboard waveform from WDT data, for use in Arbitrary Waveform or Arbitrary Sequence output mode. The waveform handle returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

Notes The data must be between -1.0 and +1.0. Use the Gain parameter of the <u>niFgen Configure Arbitrary Waveform VI</u> or <u>niFgen Configure Arbitrary Sequence VI</u> to generate different voltage outputs.

You can set **Use Waveform dT for Rate** to **True** to obtain rate information from the waveform, using the inverse of the dT as the rate. If **Use Waveform dT for Rate** is **False**, the sample rate you set by calling <u>niFgen Set Sample Rate VI</u> or by setting the <u>Sample</u> <u>Rate</u> property is used as the sample rate. If no sample rate is set, the default sample rate is used.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform** specifies the data you want to use for the arbitrary waveform.
- **Use Waveform dT for Rate** specifies whether the sample rate should be set to match the sampling information contained in the **Waveform** parameter.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Create Waveform (DBL)

Creates an onboard waveform from double-precision floating-point data for use in Arbitrary Waveform or Arbitrary Sequence output mode. The waveform handle returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

Ins <sup>i</sup> Wavef	trument H Channe orm Data error in (ne	Handle Out el Name Waveform Handle Array error out o error)	
170	<b>Instrument Handle</b> identifies your instrument session. <b>Instrument Handle</b> was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u> .		
abc	Channel Name specifies the channel this VI uses.		
[DBL]	<b>Waveform Data Array</b> specifies the array of data you want to load into the new waveform.		
	You r -1.00	nust normalize the data points in the array to be between and +1.00.	
[255]	<b>error</b> functi	<b>in</b> describes error conditions that occur before this VI or on runs.	
	TFI	<b>status</b> is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.	
	132	<b>code</b> is the error or warning code. The default is 0. If <b>status</b> is TRUE, <b>code</b> is a negative error code. If <b>status</b> is FALSE, <b>code</b> is 0 or a warning code.	
	abc	<b>source</b> identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.	
170	<b>Instrument Handle Out</b> passes a reference to your instrument session to the next VI.		
132	Waveform Handle returns a handle that identifies the waveform.		
	error out contains error information. If error in indicates that an		

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Create Waveform From File (I16)

Takes binary data from a specified file of 16-bit integers and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **Waveform Handle** returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed. This VI can also create an onboard waveform from complex data for use on devices with the <u>OSP Enabled</u> property set to **True** and the <u>Data Processing Mode</u> property set to **Complex**. To write complex data, you must interleave the IQ pairs.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **File Name** specifies the full path and name of the file where the waveform data resides.
- **Byte Order** specifies the byte order of the data in the file: Big Endian or Little Endian.
  - Notes Data written by most applications in Windows (including LabWindows/CVI) is in Little Endian format. Data written to a file from LabVIEW is in Big Endian format by default on all platforms. (Big Endian and Little Endian refer to the way data is stored in memory, which can differ on different processors.)

The 16-bit integer, or I16, data (values between -32768 and +32767) is assumed to represent -1 to +1 volts. Use the Gain parameter of the <u>niFgen Configure Arbitrary Waveform</u> VI or <u>niFgen Configure Arbitrary Sequence VI</u> to generate different voltage outputs.



error in describes error conditions that occur before this VI or

function runs.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Create Waveform From File (HWS)

Takes waveform data from a specified HWS (Hierarchical Waveform Storage) file and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **Waveform Handle** returned can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

When the Analog Waveform Editor saves data in an HWS file, it also stores the sample rate, gain, and offset with the data. If the **Use Rate from Waveform** and **Use Gain and Offset from Waveform** inputs are set to **True**, this VI also sets those properties on NI-FGEN.

 $\overline{\mathbb{N}}$ 

**Notes** If you choose to have this VI set the gain and offset properties for you, you should *not* use <u>niFgen Configure Arbitrary</u> <u>Waveform VI</u> or <u>niFgen Configure Arbitrary Sequence VI</u>, as they also set the gain and offset, thereby overriding the values set by this VI. Instead, use the niFgen Property Node to set the Arbitrary Waveform Handle or Arbitrary Sequence Handle properties.

If you set the **Use Rate from Waveform** input to **True** and if Onboard Signal Processing (OSP) is enabled, the rate from the waveform is interpreted as the IQ rate, and NI-FGEN sets the <u>IQ</u> <u>Rate</u> property for you. In all other cases, it is interpreted as the sample rate, and FGEN sets the sample rate attribute for you.

HWS files may contain 16-bit integer (I16) or double-precision floating-point (DBL) data (depending on how it was saved). This VI automatically adapts to either data type. If the file contains floatingpoint data, it must be between -1.0 and +1.0. Check the Scale Data to +/- 1V option if writing data from the Analog Waveform Editor to ensure your data is between -1.0 and +1.0.

Use the **Gain** parameter of the <u>niFgen Configure Arbitrary</u> <u>Waveform VI</u> or <u>niFgen Configure Arbitrary Sequence VI</u> to generate different voltage outputs.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **File Name** specifies the full path and name of the file where the waveform data resides.
- **Use Rate from Waveform** specifies whether or not NI-FGEN retrieves the sample rate or IQ rate from the specified HWS file and applies it to the NI-FGEN driver.
- **Use Gain and Offset from Waveform** specifies whether or not NI-FGEN retrieves the gain and offset values from the specified HWS file and applies them to the NI-FGEN driver.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status

that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
#### niFgen Create Waveform From File (DBL)

Takes waveform data from a specified file of double-precision floatingpoint data and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **Waveform Handle** returned can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

Notes The floating-point data must be between -1.0 and +1.0. Use the **Gain** parameter of the <u>niFgen Configure Arbitrary</u> <u>Waveform VI</u> or <u>niFgen Configure Arbitrary Sequence VI</u> to generate different voltage outputs.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **File Name** specifies the full path and name of the file where the waveform data resides.
- Byte Order specifies the byte order of the data in the file: Big Endian or Little Endian
  - Notes Data written by most applications in Windows (including LabWindows/CVI) is in Little Endian format. Data written to a file from LabVIEW is in Big Endian format by default on all platforms. (Big Endian and Little Endian refer to the way data is stored in memory, which can differ on different processors.)
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Create Waveform (Complex DBL)

Creates an onboard waveform from complex double-precision floatingpoint data for use in Arbitrary Waveform or Arbitrary Sequence output mode on devices with the <u>OSP Enabled</u> property set to **True** and the <u>Data Processing Mode</u> property set to **Complex**. The **Waveform Handle** returned by the VI can be used later for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Data Array specifies the array of data you want to load into the new waveform. The array must have at least as many elements as the value that you specify in the Waveform Size parameter.

You must normalize the data points in the array to be between -1.00 and +1.00.

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

**Instrument Handle Out** passes a reference to your instrument session to the next VI.

- **Waveform Handle** returns a handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Configure Arbitrary Waveform

Configures the properties of the signal generator that affect arbitrary waveform generation. Selects the arbitrary waveform to produce and sets the gain and offset.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Waveform** using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- •••• Offset specifies the value the signal generator adds to the arbitrary waveform. When you create an arbitrary waveform, you must first normalize the data points to a range of -1.00 to +1.00. You can use this parameter to shift the range of the arbitrary waveform.

For example, to configure the output signal to range from 0.00 V to 2.00 V instead of -1.00 V to 1.00 V, set offset to 1.00.

Units: volts

**Gain** specifies the factor by which the signal generator scales the arbitrary waveforms in the sequence. When you create an arbitrary waveform, you must first normalize the data points to a range of -1.00 to +1.00. You can use this parameter to scale the waveform to other ranges. The gain is applied before the offset is added. The gain is applied before the offset is added.

For example, to configure the output signal to range from -2.00 V to +2.00 V, set the gain to 2.00.

Units: volts

- Waveform Handle selects the arbitrary waveform to produce. You can create multiple arbitrary waveforms using <u>niFgen Create</u> Waveform (poly) VI. niFgen Create Waveform (poly) VI returns a handle that you use to identify each waveform.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Clear Arbitrary Waveform

Removes a previously created arbitrary waveform from the signal generator memory and invalidates the waveform handle.



**Note** The signal generator must not be in the Generating state when you call this VI.

Instrument Handle	Instrument Handle Out
Waveform Handle	
error in (no error)	error out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Waveform Handle selects the arbitrary waveform to clear. You can create multiple arbitrary waveforms using <u>niFgen Create</u> Waveform (poly) VI. niFgen Create Waveform (poly) VI returns a handle that you use to identify each waveform.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Clear Arbitrary Memory

Removes all previously created arbitrary waveforms, sequences, and scripts from the signal generator memory, and invalidates all waveform handles, sequence handles, and waveform names.



**Note** The signal generator must not be in the Generating state when you call this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Allocate Waveform

Specifies the size of a waveform up front so that it can be allocated in onboard memory before loading the associated data. Data can then be loaded in smaller blocks with the <u>niFgen Write Waveform (poly) VI</u>.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Waveform** or **Arbitrary Sequence** using <u>niFgen Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Size** specifies the size of the waveform in samples. This value must be an integer multiple of the waveform quantum.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.

- **Waveform Handle Out** returns the handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Set Waveform Next Write Position

Sets the position in the waveform to which data is written at the next write. This VI allows you to write to arbitrary locations within the waveform. These settings apply only to the next write to the waveform specified by the **Waveform Handle** parameter. Subsequent writes to that waveform begin where the last write left off, unless this VI is called again. The **Waveform Handle** passed in must have been created by a call to niFgen Allocate Waveform VI or niFgen Clear Arbitrary Waveform VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u>.
- **Channel Name** specifies the channel this VI uses.
- Waveform Handle specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- **Relative To** specifies the reference position in the waveform. The position and offset together determine where to start loading data into the waveform.
- **Offset** specifies the offset from the **Relative To** parameter at which to start loading the data into the waveform.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle Out** returns the handle that identifies the waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Write Waveform (poly)

Writes data to the waveform in onboard memory.

The waveform handle passed in must have been created by a call to niFgen Allocate Waveform VI or niFgen Create Waveform VI. The write position and offset can be set by calling niFgen Set Waveform Next Write Position VI.

Use the pull-down menu to select an instance of this VI.

Select an instance

#### niFgen Write Waveform

Writes data to the waveform in the onboard memory.

Waveform Data Instrument Handle Channel Name error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Handle specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- **Waveform Data** specifies the array of data you want to load into the waveform.

You must normalize the data points in the array to be between -1.00 and +1.00.

- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle Out** returns the handle that identifies the waveform.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Binary 16 Waveform

Writes binary data to the waveform in onboard memory. This instance can also write complex data to the waveform in onboard memory on devices with the OSP Enabled property set to True and the Data Processing Mode property set to Complex. To write complex data, you must interleave the IQ pairs or use the niFgen Write Waveform (Complex DBL) VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Handle** specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- **Waveform Data** specifies the array of data you want to load into the waveform.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle Out** returns the handle that identifies the

waveform.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write WDT Waveform

Writes WDT data to the waveform in onboard memory.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Handle specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- Use Waveform dT for Rate specifies whether the sample rate should be set to match the sampling information contained in the Waveform parameter.
- **Waveform** specifies the data you want to use for the arbitrary waveform.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Waveform Handle Out returns the handle that identifies the waveform.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Waveform (Complex DBL)

Writes complex data to the waveform in onboard memory on devices with the OSP Enabled property set to True and the Data Processing Mode property set to Complex.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Handle** specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- **Waveform Data** specifies the array of data you want to load into the waveform.

You must normalize the data points in the array to be between -1.00 and +1.00.

- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Handle Out** returns the handle that identifies the

waveform.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Waveform (Direct DMA I16)

Writes a specified amount of data from a Direct DMA-compatible device to the waveform in onboard memory.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Handle** specifies the handle of the arbitrary waveform previously allocated with <u>niFgen Allocate Waveform VI</u>.
- **Samples to Write** specifies the number of samples to write from the Direct DMA-compatible data source.
- Window Address specifies the windows address of the Direct DMA-compatible data source.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Waveform Handle Out returns the handle that identifies the waveform.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# **Arbitrary Sequence**

Use the VIs located on the **NI-FGEN**»Waveform Control»Arbitrary Sequence palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



## niFgen Query Arb Sequence Capabilities

Returns the attributes of the signal generator that are related to creating arbitrary sequences. These attributes are the maximum number of sequences, minimum sequence length, maximum sequence length, and maximum loop count.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Minimum Sequence Length returns the minimum number of arbitrary waveforms the signal generator allows in a sequence.
- Maximum Number of Sequences returns the maximum number of arbitrary waveform sequences that the signal generator allows. On some signal generators this may vary with remaining onboard memory.
- Maximum Sequence Length returns the maximum number of arbitrary waveforms the signal generator allows in a sequence. On

some signal generators this may vary with remaining onboard memory.

- **Maximum Loop Count** returns the maximum number of times the signal generator can repeat an arbitrary waveform in a sequence.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Create Arbitrary Sequence

Creates an arbitrary sequence from an array of waveform handles and an array of corresponding loop counts. This VI returns a handle that identifies the sequence. You pass this handle to <u>niFgen Configure</u> <u>Arbitrary Sequence VI</u> to specify what arbitrary sequence you want the signal generator to produce.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to Arbitrary Sequence using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.

An arbitrary sequence consists of multiple waveforms. For each waveform, you can specify the number of times that the signal generator produces the waveform before proceeding to the next waveform. The number of times to repeat a specific waveform is called the loop count.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Loop Counts Array specifies the array of loop counts that you want to use to create a new arbitrary sequence. Each Loop Counts Array element corresponds to a Waveform Handles Array element and indicates how many times to repeat that waveform. Each element of the Loop Count Array must be less than or equal to the maximum loop count the signal generator allows. You can obtain the maximum loop count from the Maximum Loop Count parameter of niFgen Query Arb Sequence Capabilities VI.
- Sequence Length specifies the number of waveforms in the new arbitrary sequence that you want to create. The value you pass must be between the minimum and maximum sequence lengths the signal generator allows, which on some signal generators

depend on the remaining unallocated memory. You can obtain the minimum and maximum sequence lengths from the **Minimum Sequence Length** and **Maximum Sequence Length** parameters in <u>niFgen Query Arb Sequence Capabilities VI</u>.

- Waveform Handles Array specifies the array of waveform handles from which you want to create a new arbitrary sequence. Each Waveform Handles Array element has a corresponding Loop Counts Array element that indicates how many times that waveform is repeated. You obtain waveform handles when you create arbitrary waveforms with <u>niFgen Create Waveform (poly) VI</u> or <u>niFgen Allocate Waveform VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Sequence Handle** returns a handle that identifies the new sequence.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Create Advanced Arb Sequence

Creates an arbitrary sequence from an array of waveform handles and an array of corresponding loop counts. This VI returns a handle that identifies the sequence. You pass this handle to <u>niFgen Configure</u> <u>Arbitrary Sequence VI</u> to specify what arbitrary sequence you want the signal generator to produce.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Sequence** using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.

An arbitrary sequence consists of multiple waveforms. For each waveform, you specify the number of times the signal generator produces the waveform before proceeding to the next waveform. The number of times to repeat a specific waveform is called the loop count.

niFgen Create Advanced Arb Sequence VI extends <u>niFgen Create</u> <u>Arbitrary Sequence VI</u> by adding the ability to set the number of samples in each sequence step and to set marker locations.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Marker Location Array specifies the array of marker locations where you want a marker to be generated in the sequence. Each Marker Location Array element corresponds to a Waveform Handles Array element and indicates where in the waveform a marker is to occur. The marker location must be less than the size of the waveform the marker is in.

If you do not want a marker generated for a particular sequence stage, set the marker location to -1 (no marker).

- Loop Counts Array specifies the array of loop counts that you want to use to create a new arbitrary sequence. Each Loop Counts Array element corresponds to a Waveform Handles Array element and indicates how many times to repeat that waveform. Each element of the Loop Count Array must be less than or equal to the maximum loop count the signal generator allows. You can obtain the maximum loop count from the Maximum Loop Count parameter of niFgen Query Arb Sequence Capabilities VI.
- Sequence Length specifies the number of waveforms in the new arbitrary sequence that you want to create. The value you pass must be between the minimum and maximum sequence lengths the signal generator allows, which on some signal generators depend on the remaining unallocated memory. You can obtain the minimum and maximum sequence lengths from the Minimum Sequence Length and Maximum Sequence Length parameters in <u>niFgen Query Arb Sequence Capabilities VI</u>.
- Waveform Handles Array specifies the array of waveform handles from which you want to create a new arbitrary sequence. Each Waveform Handles Array element has a corresponding Loop Counts Array element that indicates how many times that waveform is repeated. You obtain waveform handles when you create arbitrary waveforms with <u>niFgen Create Waveform (poly) VI</u> or <u>niFgen Allocate Waveform VI</u>.
- Sample Counts Array specifies the array of sample counts that you want to use to create a new arbitrary sequence. Each Sample Counts Array element corresponds to a Waveform Handles Array element and indicates the subset, in samples, of the given waveform to generate. Each element of the Sample Count Array must be larger than the minimum waveform size, a multiple of the waveform quantum and no larger than the number of samples in the corresponding waveform. You can obtain these values by calling niFgen Query Arb Waveform Capabilities.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Coerced Markers Array** returns an array of all given markers that are coerced (rounded) to the nearest marker quantum. Not all devices coerce markers.
- **Sequence Handle** returns a handle that identifies the new sequence.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Configure Arbitrary Sequence

Configures the signal generator properties that affect arbitrary sequence generation. Selects the arbitrary sequence to produce and sets the gain and offset.



**Notes** The signal generator must not be in the Generating state when you call this VI.

You must set **Output Mode** to **Arbitrary Sequence** using <u>niFgen</u> <u>Configure Output Mode VI</u> before calling this VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- •••• Offset specifies the value the signal generator adds to the arbitrary waveform. When you create an arbitrary waveform, you must first normalize the data points to a range of -1.00 to +1.00. You can use this parameter to shift the range of the arbitrary waveform.

For example, to configure the output signal to range from 0.00 V to 2.00 V instead of -1.00 V to 1.00 V, set offset to 1.00.

Units: volts

- Sequence Handle specifies the handle of the arbitrary sequence to produce. You can create multiple arbitrary sequences using niFgen Create Arbitrary Sequence VI or niFgen Create Advanced Arb Sequence VI. Those VIs return a handle that you use to identify each sequence.
- Gain specifies the factor by which the signal generator scales the arbitrary waveforms in the sequence. When you create an arbitrary waveform, you must first normalize the data points to a range of

-1.00 to +1.00. You can use this parameter to scale the waveform to other ranges. The gain is applied before the offset is added. The gain is applied before the offset is added.

For example, to configure the output signal to range from -2.00 V to +2.00 V, set the gain to 2.00.

Units: volts

**error in** describes error conditions that occur before this VI or function runs.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
## niFgen Clear Arbitrary Sequence

Removes a previously created arbitrary sequence from the signal generator memory and invalidates the sequence handle.



**Note** The signal generator must not be in the Generating state when you call this VI.

Instrument Handle	Instrument Handle Out
Sequence Handle –	
error in (no error)	error out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Sequence Handle specifies the handle of the arbitrary sequence to produce. You can create multiple arbitrary sequences using niFgen Create Arbitrary Sequence VI or niFgen Create Advanced Arb Sequence VI. Those VIs return a handle that you use to identify each sequence.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# Script

Use the VIs located on the **NI-FGEN**»**Waveform Control**»**Script** palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



### niFgen Allocate Named Waveform

Specifies the size of a named waveform up front so that it can be allocated in onboard memory before loading the associated data. Data can then be loaded in smaller blocks with the <u>niFgen Write Waveform</u> (poly) VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Name specifies the name of the waveform.
- **Waveform Size** specifies the size of the waveform in samples.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.

### niFgen Set Named Waveform Next Write Position

Sets the position in the named waveform to which data is written at the next write. This VI allows you to write to arbitrary locations within the waveform. These settings apply only to the next write to the waveform specified by the **Waveform Handle** parameter. Subsequent writes to that waveform begin where the last write left off, unless this VI is called again. The **Waveform Handle** passed in must have been created by a call to niFgen Allocate Waveform VI or niFgen Clear Arbitrary Waveform VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u>.
- Channel Name specifies the channel this VI uses.
- Waveform Name specifies the name of the waveform.
- **Relative To** specifies the reference position in the waveform. The position and offset together determine where to start loading data into the waveform.
- **Offset** specifies the offset from the **Relative To** parameter at which to start loading the data into the waveform.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Write Named Waveform

The waveform handle passed in must have been created by a call to <u>niFgen Allocate Named Waveform VI</u> or <u>niFgen Create Waveform VI</u>. The write position and offset can be set by calling <u>niFgen Set Named</u> <u>Waveform Next Write Position VI</u>

•

Use the pull-down menu to select an instance of this VI.

Select an instance

#### niFgen Write Named Waveform (DBL)

Writes floating-point data to the named waveform in onboard memory.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Name specifies the name of the waveform.
- **Waveform Data** returns the waveform output.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Named Waveform (I16)

Writes binary data to the named waveform in onboard memory. This instance can also write complex data to the named waveform in onboard memory on devices with the OSP Enabled property set to True and the Data Processing Mode property set to Complex. To write complex data, you must interleave the IQ pairs or use the niFgen Write Named Waveform (Complex DBL) VI.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Name** specifies the name of the waveform.
- **Waveform Data** specifies the array of data you want to load into the waveform.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Named Waveform (WDT)

Writes WDT data to the named waveform in onboard memory.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Name specifies the name of the waveform.
- **Waveform** specifies the data you want to use for the arbitrary waveform.
- **Use Waveform dT for Rate** specifies whether the sample rate should be set to match the sampling information contained in the **Waveform** parameter.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Named Waveform (Complex DBL)

Writes complex data to the named waveform in onboard memory on devices with the OSP Enabled property set to True and the Data Processing Mode property set to Complex.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Name** specifies the name of the waveform.
- **Waveform Data** returns the waveform output.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Write Named Waveform Direct DMA (I16)

Writes a specified amount of data from a Direct DMA-compatible device to the named waveform in onboard memory.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- Waveform Name specifies the name of the waveform.
- Window Address specifies the windows address of the Direct DMA-compatible data source.
- **Samples to Write** specifies the number of samples to write from the Direct DMA-compatible data source.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Waveform Name Out** passes the name of the named waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Delete Named Waveform

Deletes the specified named waveform from onboard memory.

Instrument Handle Out Channel Name error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Waveform Name** specifies the name of the waveform.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Write Script

Writes a string containing one or more scripts that govern the generation of waveforms.

Instrument Handle Courter Instrument Handle Out Channel Name Script

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Channel Name specifies the channel this VI uses.
- **Script** contains the text of the script you want to use for your generation operation.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code**

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Delete Script

Deletes the specified script from onboard memory.

Instrument Handle Out Channel Name

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Channel Name** specifies the channel this VI uses.
- **Script Name** specifies the name of the script you want to delete.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Is Done

Determines whether the current generation has completed. This VI sets the **Done** parameter to be true if the session is in the Idle or Committed states.



**Note** Devices can only become done in single or stepped trigger modes.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Done** indicates True if the generation has completed, False otherwise.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Wait Until Done

function runs.

Waits until the device is done generating or until the maximum time has expired.



**Note** Devices can only become done in single or stepped trigger modes.

Instru N en	ment Handle And Hinter Instrument Handle Out MaxTime (msec) Merce error out For in (no error)
1/0	<b>Instrument Handle</b> identifies your instrument session. <b>Instrument Handle</b> was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> <u>Initialize With Options VI</u> .
132	MaxTime specifies the timeout value in milliseconds.
[***)	error in describes error conditions that occur before this VI or

TF status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- 132 code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
- abc **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran. error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - TF status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - 132 code is the error or warning code. If status is TRUE, code

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Close

Performs the following operations:

- Closes the instrument I/O session.
- Destroys the NI-FGEN session and all of its properties.
- Deallocates any memory resources NI-FGEN uses.



Not all signal routes established by calling <u>niFgen Export Signal VI</u> and <u>niFgen Route Signal Out VI</u> are released when the NI-FGEN session is closed.

Routes To	NI-5401/5411/5431	Other
Front Panel	Remains connected.	Remains connected.
RTSI/PXI Backplane	Remains connected.	Disconnected

error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Initialize With Options

Performs the following initialization actions:

- Creates a new IVI instrument session and optionally sets the initial state of the following session properties: RangeCheck, Cache, Simulate, and RecordCoersions.

- Opens a session to the specified device using the interface and address that you specify for **Resource Name**.

- If **Reset Device** is set to TRUE, this VI resets the device to a known state.

- Sends initialization commands to set the instrument to the state necessary for NI-FGEN operation.

- Returns a session handle that you can use to identify the device in all subsequent NI-FGEN VI calls.



abc

**Resource Name** specifies the instrument name, for example "PXI1Slot3" where "PXI1Slot3" is an instrument name assigned by Measurement & Automation Explorer. Syntax:

Configured in MAX Under	Valid Syntax
NI-DAQmx Devices	DAQmxDeviceName
NI-DAQmx Devices (alternate syntax)	DAQ::DAQmxDeviceName
Traditional NI- DAQ Devices	DAQ::DeviceNumber



originally been designed for a Traditional NI-DAQ device use an NI-DAQmx device. For example, if the application expects "DAQ::1", you can rename the device to"1" in MAX and pass in "DAQ::1" for the resource name.

If you use the "DAQ::n" syntax and an NI-DAQmx device name already exists with the same name, the NI-DAQmx device will be matched first.

You can also pass in the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical name identifies a particular virtual instrument.

Example	resource	names	S:

.

\_

Resource Name	Description
DAQ::1	Traditional NI-DAQ device, device number = 1
DAQ::2	NI-DAQmx device, device name ="2"
myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
DAQ::myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
myLogicalName	IVI logical name or virtual instrument, name = "myLogicalName"

	Traditional NI-DAQ and NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case- sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.		
TFI	<b>Id Query</b> specifies whether or not to verify that NI-FGEN supports the device you initialize.		
	Circumstances can arise where sending an ID query to the device is undesirable. When you set this parameter to FALSE, the VI initializes the device without performing an ID query.		
TF	<b>Reset Device</b> specifies whether you want to reset the device during the initialization procedure.		
abci	<b>Option String</b> specifies the initial values of certain session properties.		
	Default Values: "Simulate=0,RangeCheck=1,Cache=1"		
	The syntax for <b>Option String</b> is		
	attributeName=value		
	where		
	<i>attributeName</i> = the name of the attribute		
	<i>value</i> = the value to which the attribute is set.	Droporty	1
	To set multiple properties, separate them with a comma.	Name	(
	If you do not wire this input or pass in an empty string, the session uses the default values for these properties. You can override the default values by assigning a value explicitly in a string that you pass for this input.		

	You do not have to specify all of the properties and may leave any of them out. If you do not specify one of the properties, its default value is used.	
RangeCheck	niFgen»Inherent IVI Settings»User Options»Range Check	1 (True)
Cache	niFgen»Inherent IVI Settings»User Options»Cache	1 (True)
Simulate	niFgen»Inherent IVI Settings»User Options»Simulate	0 (False)

Valid Values

True: 1

False: 0

If simulation is enabled (Simulate=1), you may specify the device that you want to simulate. To specify a device, enter the following syntax in **Option String**.

DriverSetup=Model:<driver model number>;BoardType:<module type>;MemorySize:<size of onboard memory in bytes>

**Option String Examples:** 

NI PXI-5404 - Simulate=1,DriverSetup=Model:5404;BoardType:PXI

NI PCI-5421 -

Simulate=1,DriverSetup=Model:5421;BoardType:PCI;MemorySize:268435456 error in describes error conditions that occur before this VI or function runs.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Instrument Handle Out passes a reference to your instrument session to the next VI. E: error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# Utility

Use the VIs located on the **NI-FGEN»Utility** palette to use additional features of the NI-FGEN driver.

Click the icons for VI and function descriptions.



## niFgen Reset

Resets the instrument to a known state. This VI aborts the generation, clears all routes, and resets session properties to the default values. This VI does not, however, commit the session properties or configure the device hardware to its default state.



**Note** For the NI 5401/5404/5411/5431, this VI exhibits the same behavior as <u>niFgen Reset Device VI</u>.

Instrument Handle And HI-FGEN Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code**

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Reset Device

Performs a hard reset on the device. Generation is stopped, all routes are released, external bidirectional terminals are tri-stated, FPGAs are reset, hardware is configured to its default state, and all session properties are reset to their default states.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Disable

Places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected. The analog output and all exported signals are disabled.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
## niFgen Revision Query

Returns the revision numbers of NI-FGEN and the instrument firmware.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **Size of Instrument Driver Revision** specifies the maximum size of the instrument driver revision string and must be set to a minimum of 2048.
- **Size of Firmware Revision** specifies the maximum size of the firmware revision string and must be set to a minimum of 2048.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Instrument Driver Revision** returns the NI-FGEN software revision numbers in the form of a string.
- **Firmware Revision** returns the instrument firmware revision numbers in the form of a string. Not all devices have firmware information available.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Get Session Reference

Extracts a session that can be passed to NI-TClk VIs. Session References are of generic type, which means that the corresponding wires are blue-green, unlike the wires for regular instrument driver sessions.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Session Reference references the device session that can be passed to NI-TClk VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code**

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Util Create Waveform Data

Creates a array of doubles filled with the specified waveform.



միլին

Waveform Data



**Waveform** specifies waveform information that is used to generate the waveform data.

**Waveform Type** selects the type of waveform.

Amplitude specifies the amplitude of the waveform that you want the signal generator to produce. This value is the amplitude at the output terminal.

For example, to produce a waveform ranging from -5.00 V to +5.00 V, set **Amplitude** to 5.00 V. Units: peak voltage

Note This parameter does not affect signal generator behavior when you set **Waveform Type** to DC.

# of Points specifies the number of points in the waveform.

**Waveform Data** returns the waveform output.

#### niFgen Util Create Frequency Sweep Data

Creates an array of doubles filled with the frequencies specified. The frequencies are created in a logarithmic progression from **Start Frequency** to **Stop Frequency**.



- **Start Frequency** specifies the first frequency of the sweep to be generated.
- **Stop Frequency** specifies the last frequency of the sweep to be generated.
- # Frequency Steps specifies the number of frequencies in the frequency sweep.
- **Frequency Sweep Array** returns the array of double-precision floating-point data filled with the frequencies specified.

#### niFgen Util Create Bin16 Waveform Data

Creates an array of 16-bit integers (I16) corresponding to the selected waveform. 1.0 is scaled to full range of 32,767.

Waveform Data

- **Waveform** specifies waveform information that is used to generate the waveform data.
  - **Waveform Type** selects the type of waveform.
  - Amplitude specifies the amplitude of the waveform that you want the signal generator to produce. This value is the amplitude at the output terminal.

For example, to produce a waveform ranging from -5.00 V to +5.00 V, set **Amplitude** to 5.00 V. Units: peak voltage



**Note** This parameter does not affect signal generator behavior when you set **Waveform Type** to DC.

# of Points specifies the number of points in the waveform.

**Waveform Data** returns the waveform output.

#### niFgen Error Message

Converts a status code returned by an NI-FGEN VI into a user-readable string.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Message Box specifies whether or not a Dialog Message is shown.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Error Code** specifies the **code** parameter that is returned from any of the NI-FGEN VIs.
- **Error Message** returns the user-readable message string that corresponds to the status code you specify.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

#### niFgen Self-Test

Runs the instrument self-test routine and returns the test results.

Note When used on some signal generators, the device will be reset after the niFgen Self-Test VI runs. If you use the niFgen Self-Test VI, your device may not be in its previously configured state after the VI has run.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- Size of Self-Test Message specifies the maximum size of the Self-Test message string and must be set to a minimum of 2048.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Self-Test message returns the self-test response string from the instrument. Refer to the Error and Status Information topic in the *NI Signal Generators Help* for an explanation of the string contents.
- **Self-Test result** returns the value returned from the instrument

self-test. Zero means success. For any other code, refer to the Error and Status Information topic in the *NI Signal Generators Help*.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Property Node

The niFgen Property Node is used to set, get, or check properties.

<u>+</u> ©

Some NI-FGEN properties are channel based. When a property is channel based, you can specify an active channel before setting, getting, or checking properties.

## Calibration

Use the VIs located on the **NI-FGEN**»**Calibration** palette to calibrate your signal generator. Different signal generators use different calibration VIs. Refer to your signal generator calibration procedure for more information.

Click the icons for VI and function descriptions.



## niFgen Self Cal

Performs a full internal (self-) calibration on the device. If the calibration is successful, new calibration data and constants are stored in the onboard EEPROM. Refer to the calibration procedure of the signal generator for more information.



**Note** This VI does not apply to the NI 5401/5404/5411/5431 which have their own calibration VIs.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code**

is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Init Ext Cal

Creates and initializes a special NI-FGEN external calibration session. The ViSession returned is an FGEN session that can be used to configure the device using normal attributes and functions. However, flags have been set that allow the user to program an external calibration procedure using the special calibration properties and VIs. The NI 5401/5404/5411/5431 have different calibration VIs. Refer to the calibration procedure of the signal generator for more information.

Resource Name Anti-FGEN Instrument Handle Out Password

Resource Name specifies the instrument name, for example "PXI1Slot3" where "PXI1Slot3" is an instrument name assigned by Measurement & Automation Explorer. Syntax:

Configured in MAX Under	Valid Syntax
NI-DAQmx Devices	DAQmxDeviceName
NI-DAQmx Devices (alternate syntax)	DAQ::DAQmxDeviceName

Traditional NI-DAQ Devices DAQ::DeviceNumber

**Caution** The alternate syntax allowed for NI-DAQmx devices is intended to allow application that may have originally been designed for a Traditional NI-DAQ device use an NI-DAQmx device. For example, if the application expects "DAQ::1", you can rename the device to"1" in MAX and pass in "DAQ::1" for the resource name.

If you use the "DAQ::n" syntax and an NI-DAQmx device name already exists with the same name, the NI-DAQmx device will be matched first.

You can also pass in the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical name identifies a particular virtual instrument.

Example resource names:

<b>Resource Name</b>	Description
DAQ::1	Traditional NI-DAQ device, device number = 1
DAQ::2	NI-DAQmx device, device name ="2"
myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
DAQ::myDAQmxDevice	NI-DAQmx device, device name ="myDAQmxDevice"
myLogicalName	IVI logical name or virtual instrument, name = "myLogicalName"

Traditional NI-DAQ and NI-DAQmx device names are not casesensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

- **Password** specifies calibration password required to open an external calibration session to the device.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Cal Control**

Use the VIs located on the NI-FGEN»Calibration»Cal Control palette only when following the calibration procedure of the signal generator. Navigate to Start»All Programs»National Instruments»NI-FGEN»Documentation»Calibration to find Calibration Procedures for your signal generator.

Click the icons for VI and function descriptions.



## **ADC Control**

Use the VIs located on the **NI-FGEN**»Calibration»Cal Control»ADC Control palette only when following the calibration procedure of the signal generator.

Click the icons for VI and function descriptions.



#### niFgen Initialize Cal ADC Calibration

Initializes an external calibration session for ADC calibration. For the NI 5421/5422/5441, ADC calibration involves characterizing the gain and offset of the onboard ADC

Instrument Handle And HI-FGEN Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

## niFgen Write Binary 16 Analog Static Value

Writes a 16-bit value to the DAC, which could be output as a DC voltage. This VI writes to the DAC only when used in an external calibration session.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **Value** specifies the value to write.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Read CAL ADC

Takes one or more voltage measurements from the onboard calibration ADC and returns their average value. The signal that the ADC actually measures can be specified using the <u>Cal ADC Input</u> property. The ADC has some inherent gain and offset. These values can be determined during an external calibration session and stored in the calibration EEPROM.

If the **Return Calibrated Value?** is True, NI-FGEN adjusts the value that is returned to account for the gain and offset of the ADC. Otherwise, the raw voltage value reported by the ADC is returned.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **Number of Reads to Average** specifies the number of measurements to be taken and averaged to determine the return value.
- **Return Calibrated Value?** specifies whether the voltage returned from the ADC should be adjusted to account for the gain and offset of the ADC.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Calibration ADC Value** returns the average of the voltage measurements taken from the onboard calibration ADC.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Cal Adjust Cal ADC

Calculates calibration constants pertaining to the gain and offset of the onboard calibration ADC. During external calibration, you can generate voltages and measure them both externally and with the calibration ADC. The measured voltages are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Voltages Measured Externally** is an array of analog output voltages measured with an external instrument.
- **Voltages Measured with Cal ADC** is an array of analog output voltages measured with the onboard calibration ADC.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Oscillator Control**

Use the VIs located on the **NI-FGEN**»Calibration»Cal Control»Oscillator Control palette only when following the calibration procedure of the signal generator.

Click the icons for VI and function descriptions.



#### niFgen Initialize Oscillator Frequency Calibration

Sets up the device to start the oscillator calibration.

The session handle should be the handle returned by <u>niFgen Init Ext Cal</u>  $\underline{VI}$ .

Instrument Handle Out Instrument Handle ~ NI-FGEN Aoso error in (no error) 🚥 error out Instrument Handle identifies your instrument session. Instrument I/0 | Handle was obtained from the <u>niFgen Init Ext Cal VI</u>. error in describes error conditions that occur before this VI or function runs. TFI status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. 132 code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, **code** is 0 or a warning code. abc **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error. **Instrument Handle Out** passes a reference to your instrument session to the next VI. error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. TF status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

#### niFgen Cal Adjust Oscillator Frequency

Calculates calibration constants pertaining to the oscillator. During external calibration, you can generate sine waves and take measurements of the resulting output frequencies. The desired and measured frequencies are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Desired Frequency In Hz** specifies the expected frequency of the output waveform.
- Measured Frequency in Hz specifies the actual (measured) frequency of the output waveform.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## **Analog Output Control**

Use the VIs located on the **NI-FGEN**»Calibration»Cal Control»Analog Output Control palette only when following the calibration procedure of the signal generator.

Click the icons for VI and function descriptions.



## niFgen Initialize Analog Output Calibration

Sets up the device to start the analog output calibration.

The session handle should be the handle returned by <u>niFgen Init Ext Cal</u>  $\underline{VI}$ .

Instrument Handle out error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.
## niFgen Cal Adjust Main Path Output Impedance

Calculates calibration constants pertaining to main analog path output impedance. During external calibration, you can put the device in different configurations and take measurements of the resulting output voltage across different loads. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- Measured Source Voltage specifies the analog output voltage measured across a very high-impedance load.
- Load Impedance specifies the impedance of the load across which the measurement passed in as Measured Voltage Across Load is taken.
- **Configuration** specifies the configuration of the device for this stage of calibration.
- Measured Voltage Across Load specifies the analog output voltage measured across the load impedance specified in Load Impedance.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Cal Adjust Main Path Pre Amp Gain

Calculates calibration constants pertaining to the pre-amplifier gain of the main analog path. During external calibration, you can put the device in different configurations; program different gain, offset, and main DAC values; and take measurements of the resulting output voltages. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **GainDACValues** is an array of the values programmed to the gain calibration DAC during this calibration stage.
- MainDACValues is an array of the values programmed to the main output DAC during this calibration stage.
- **Configuration** specifies the configuration of the device for this stage of calibration.
- **MeasuredOutputs** is an array of the analog output voltages measured during this calibration stage.
- **OffsetDACValues** is an array of the values programmed to the offset calibration DAC during this calibration stage.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

132

**code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Cal Adjust Main Path Post Amp Gain and Offset

Calculates calibration constants pertaining to the post-amplifier gain and offset of the main analog path. During external calibration, you can put the device in different configurations; program different gain, offset, and main DAC values; and take measurements of the resulting output voltage. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **GainDACValues** is an array of the values programmed to the gain calibration DAC during this calibration stage.
- MainDACValues is an array of the values programmed to the main output DAC during this calibration stage.
- **Configuration** specifies the configuration of the device for this stage of calibration.
- **MeasuredOutputs** is an array of the analog output voltages measured during this calibration stage.
- **OffsetDACValues** is an array of the values programmed to the offset calibration DAC during this calibration stage.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Cal Adjust Main Path Pre Amp Offset

Calculates calibration constants pertaining to the pre-amplifier offset of the main analog path. During external calibration, you can put the device in different configurations; program different gain, offset, and main DAC values; and take measurements of the resulting output voltages. The configuration data as well as the measurements, are passed to this VI so that, NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **GainDACValues** is an array of the values programmed to the gain calibration DAC during this calibration stage.
- **Configuration** specifies the configuration of the device for this stage of calibration.
- MeasuredOutputs is an array of the analog output voltages measured during this calibration stage.
- **OffsetDACValues** is an array of the values programmed to the offset calibration DAC during this calibration stage.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Cal Adjust Direct Path Output Impedance

Calculates calibration constants pertaining to direct analog path output impedance. During external calibration, you can put the device in different configurations and take measurements of the resulting output voltage across different loads. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- Measured Source Voltage specifies the analog output voltage measured across a very high-impedance load.
- Load Impedance specifies the impedance of the load across which the measurement passed in as Measured Voltage Across Load is taken.
- **Configuration** specifies the configuration of the device for this stage of calibration.
- Measured Voltage Across Load specifies the analog output voltage measured across the load impedance specified in Load Impedance.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Cal Adjust Direct Path Gain

Calculates calibration constants pertaining to the gain of the direct analog path. During external calibration, you can put the device in different configurations; program different gain and main DAC values; and take measurements of the resulting output voltages. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **GainDACValues** is an array of the values programmed to the gain calibration DAC during this calibration stage.
- MainDACValues is an array of the values programmed to the main output DAC during this calibration stage.
- MeasuredOutputs is an array of the analog output voltages measured during this calibration stage.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

**Instrument Handle Out** passes a reference to your instrument session to the next VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# **Flatness Control**

Use the VIs located on the **NI-FGEN**»Calibration»Cal Control»Flatness Control palette only when following the calibration procedure of the signal generator.

Click the icons for VI and function descriptions.

🔄 Flatness Control 📃 🗖 🔀		
🗘 🔍 Search	0	

## niFgen Initialize Flatness Calibration

Initializes an external calibration session to calibrate flatness.

Instrument Handle Andrew Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Cal Adjust Flatness

During external calibration, the device is configured with the different analog settings. Take measurements of the resulting output voltage across different frequencies. The configuration data, as well as the measurements, are passed to this VI so that NI-FGEN can calculate the appropriate calibration constants and, when the calibration session is committed, store them in the onboard EEPROM.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- **Channel Name** specifies the channel this VI uses.
- **Requested Amplitude** specifies the Amplitude (in Volts) that was used to configure NI-FGEN in order to generate the sine tones at different frequencies.
- **Frequencies Array** specifies the frequencies at which different sine tones were generated (Hertz).
- **Configuration** specifies the configuration of the device for this stage of calibration.
- **Measured Amplitudes Array** specifies the amplitudes measured for each corresponding frequency in the Frequencies Array (Volts).
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Close Ext Cal

Closes an NI-FGEN external calibration session and, if specified, stores the new calibration constants and calibration data, such as time and temperature, in the onboard EEPROM.

Action Action error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- Action specifies the action to perform upon closing. Valid choices are Abort, which disregards the new calibration constants, and Commit, which stores the new calibration constants on the device.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# Utility

Use the VIs located on the **NI-FGEN**»**Calibration**»**Utility** palette to use additional NI-FGEN calibration features.

Click the icons for VI and function descriptions.



## niFgen Get Self Cal Supported

Returns whether or not the device supports self-calibration.

error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or <u>niFgen</u> Initialize With Options VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **SelfCalSupported** returns TRUE if self-calibration is supported, FALSE otherwise.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Restore Last Ext Cal Constants

Overwrites the current calibration constants with those from the last successful external calibration. This action effectively undoes any selfcalibrations performed since the last time an external calibration was performed.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Init Ext Cal VI</u>.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

# niFgen Get Ext Cal Recommended Interval

Returns the recommended interval between external calibrations in months.

Instrument Handle Andle Out error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- Months specifies the recommended maximum interval between external calibrations in months.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Get Self Cal Last Date and Time

Returns the date and time of the last successful self-calibration. The time returned is 24-hour (military) local time; for example, if the device was calibrated at 2:30 PM, this VI returns 14 for the hours parameter and 30 for the minutes parameter.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Year** is expressed as an integer.
- Month is expressed as an integer. For example, December is represented as 12.
- **Day** is expressed as an integer.
- Minute is expressed as an integer.
- Hour is expressed as an integer.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Get Self Cal Last Temp

Returns the temperature at the last successful self-calibration. The temperature is returned in degrees Celsius.

error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Temperature** is expressed in degrees Celsius.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Read Current Temperature

Reads the current onboard temperature of the device. The temperature is returned in degrees Celsius.

Instrument Handle over Instrument Handle Out

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Temperature** is expressed in degrees Celsius.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

### niFgen Get Ext Cal Last Date and Time

Returns the date and time of the last successful external calibration. The time returned is 24-hour (military) local time; for example, if the device was calibrated at 2:30 PM, this function will return 14 for the hours parameter and 30 for the minutes parameter.



- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Year** is expressed as an integer.
- Month is expressed as an integer. For example, December is represented as 12.
- **Day** is expressed as an integer.
- Minute is expressed as an integer.
- Hour is expressed as an integer.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Get Ext Cal Last Temp

Returns the temperature at the last successful external calibration. The temperature is returned in degrees Celsius.

error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **error in** describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **Temperature** is expressed in degrees Celsius.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Get Cal User Defined Info

Retrieves user-defined information from the onboard EEPROM.

error in (no error)

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- **UserDefinedInfo** is a string containing the user information.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the
error, what inputs are in error, and how to eliminate the error.

## niFgen Set Cal User Defined Info

Stores user-defined information in the onboard EEPROM.

Instrument Handle Out UserDefinedInfo

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **UserDefinedInfo** specifies the information string to be stored in the EEPROM.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## niFgen Change Ext Cal Password

Changes the password that is required to initialize an external calibration session. The password may be up to four characters long.

Instrument Handle Out Old Password A Provide A

- Instrument Handle identifies your instrument session. Instrument Handle was obtained from the <u>niFgen Initialize VI</u> or the <u>niFgen</u> Init Ext Cal VI.
- **Old Password** is the old (current) external calibration password.
- **New Password** is the new (desired) external calibration password.
- error in describes error conditions that occur before this VI or function runs.
  - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **Instrument Handle Out** passes a reference to your instrument session to the next VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

**source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

## NI-FGEN Standard Function Generator Express VI

Configures and runs National Instruments signal generators using NI-FGEN in Standard Function output mode.

The NI 5404 frequency generator is not supported.

You must place at least two NI-FGEN Express (Standard Functions) Vis on the block diagram for successful function generation. Select **Start generation** for the Generation Mode in the first Express VI to start the function generation. Select **Stop generation** for the Generation Mode in the final Express VI to abort function generation and close the NI-FGEN device handle.

Dialog Box Options Block Diagram Inputs Block Diagram Outputs

### **Dialog Box Options**

Parameter	Description
Configuration	Contains the following options:
	<ul> <li>Device—Specifies the signal generator used. This ring control lists all NI signal generators installed on this computer that can be used by this Express VI. If you re-open the VI and the selected signal generator is dimmed, it is no longer available.</li> <li>Disable output on close—Specifies the behavior of the output enable relay when the device resources to the selected signal generator are released.</li> </ul>
	Disabling the output grounds the output connector; otherwise, a constant DC voltage representing the last data point generated before stopping the waveform may be present.
	<ul> <li>Generation Mode—Specifies the generation mode of the signal generator. Contains the following options:</li> </ul>
	<ul> <li>Start generation—Configures the signal generator to generate the waveform continuously and starts the generation. In this mode, the Express VI returns immediately after starting the generation. To stop the generation, drop another instance of the Express VI on the block diagram, select the same signal generator, and configure it to Stop generation.</li> </ul>
	<ul> <li>Stop generation—Use this mode to stop a continuous generation that was started by another instance of this Express VI that was configured to Start generation.</li> </ul>
	<ul> <li>Function Setup—Configures the characteristics of the function to generate. Contains the following options:</li> </ul>

	<ul> <li>Type—Specifies the standard waveform for the function generator to produce.</li> <li>Frequency (Hz)—Specifies the frequency</li> </ul>
	of the standard waveform. - <b>Duty cycle (%)</b> —Specifies the duty cycle of the square waveform. This value is the percentage of time the square waveform is high in a cycle.
	<ul> <li>Amplitude (Vpk)—Specifies the amplitude of the standard waveform. This value is the amplitude at the output terminal. For example, to produce a waveform ranging from -5.00 to +5.00 V, set Amplitude (Vpk) to 5.00.</li> </ul>
	<ul> <li>DC offset (V)—Specifies the DC offset of the standard waveform. This value is the offset from ground to the center of the waveform. For example, to configure a waveform with an amplitude of 10.00 V to range from 0.00 V to +10.00 V, set DC offset (V) to 5.00.</li> </ul>
	<ul> <li>Start phase (deg)—Specifies the horizontal offset of the standard waveform.</li> </ul>
	<ul> <li>Output impedance—Specifies the output impedance of the signal generator.</li> </ul>
	<ul> <li>Load impedance—Specifies the load impedance. Using this control allows the signal generator to produce the correct output signal for the most common applications: high (&gt;1 MΩ) load impedance; or matched output and load impedance.</li> </ul>
Trigger and	Contains the following options:
Timing	• <b>Trigger type</b> —Specifies the type of trigger to use for starting waveform generation. Selecting an Immediate trigger starts waveform generation as soon as all configuration settings are applied.
	• Irigger source—Specifies the trigger source. The

sig	gnal generator waits for the specified trigger to
st	art waveform generation.
• E0	<b>dge</b> —Specifies whether to trigger on a rising
e0	lge or a falling edge of the trigger signal.
• R0	<b>eference clock source</b> —Specifies the source of
a	10 MHz reference clock signal. The function
ge	enerator uses this reference clock source to
de	erive output frequencies.

## **Block Diagram Inputs**

Parameter	Description
error in	Describes error conditions that occurred prior to running this Express VI.

### **Block Diagram Outputs**

Parameter	Description	
error out	Contains error information. If <b>error in</b> indicates that an error	
	occurred before this Express VI, error out contains the	
	same error information. Otherwise, it describes the error	
	status that this Express VI produces.	

## NI-FGEN Arbitrary Waveform Generator Express VI

Configures and runs National Instruments signal generators using NI-FGEN in Arbitrary Waveform mode.

For continuous generation, you must place at least two NI-FGEN Express (Arb) Vis on the block diagram for successful arbitrary waveform generation. Select Start continuous for the Generation Mode in the first Express VI to start the continuous waveform generation. Select Stop continuous for the Generation Mode in the final Express VI to abort waveform generation and close the NI-FGEN device handle.

When placed in a loop, the Express VI downloads new data in every loop iteration.

Dialog Box Options Block Diagram Inputs Block Diagram Outputs

### **Dialog Box Options**

Parameter	Description
Configuration	Contains the following options:
	<ul> <li>Device—Specifies the signal generator used. This ring control lists all NI signal generators installed on this computer that can be used by this Express VI. If you relaunch the VI and the selected signal generator is dimmed, it is no longer available.</li> <li>Disable output on close—Specifies the behavior of the output enable relay when the device resources to the selected signal generator are released.</li> </ul>
	Disabling the output grounds the output connector; otherwise, a constant DC voltage representing the last data point generated before stopping the waveform may be present.
	<ul> <li>Generation Mode—Specifies the generation mode of the function generator. Contains the following options:</li> </ul>
	<ul> <li>Finite—Configures the signal generator to generate the waveform one time. In this generation mode, the Express VI waits until waveform generation is complete or until the time specified by Max time (s) has elapsed, whichever occurs first.</li> </ul>
	- <b>Start continuous</b> —Configures the signal generator to generate the waveform continuously and starts the generation. In this mode, the Express VI returns immediately after starting the generation. To stop the generation, drop another instance of the Express VI on the block diagram, select the same signal generator, and configure it to <b>Stop continuous</b> .
	<ul> <li>Stop continuous—Specifies to stop a</li> </ul>

continuous waveform generation.

- **Data Source**—Configures the data source of the signal generator. Contains the following options:
  - Data at VI Terminal—Sets the Express VI to receive its waveform data from the diagram input connector on the VI.

The VI that calls this Express VI must run before opening the Express VI for data to be present on the Express VI input connector. If no data is present on this input connector or the Express VI has not been configured previously, the VI uses Demo Data to validate the configuration.

 Waveform from file—Select this control to generate a waveform stored in a Hierarchical Waveform Storage (HWS) file.
 HWS files can be created by the NI-HWS API or the NI Analog Waveform Editor.

Use the **Waveform file selector** to specify the file to load.

- Waveform file selector—Select the file that contains the waveform to generate. Use the New waveform and Edit waveform options to easily create and edit waveforms using the NI Analog Waveform Editor.
- **Output**—Contains the following options:
  - Extract from waveform—Specifies whether settings for Sample rate (S/s),
     Gain, and Offset (V) should be extracted from the waveform or specified manually.

When **Extract from waveform** is not selected, you must first normalize the data points to a range of -1.00 to +1.00.

- Sample rate (S/s)—Specifies the sample

rate with which the signal generator produces arbitrary waveforms.

This control is disabled if the option **Extract from waveform** is selected. In this case, the control displays the extracted value to apply to the signal generator.

- **Gain**—Specifies the factor by which the signal generator scales the data.

When **Extract from waveform** is not selected, you must first normalize the data points to a range of -1.00 to +1.00. Use this control to scale the arbitrary waveform to other ranges. For example, to configure the output signal to range from -2.00 to +2.00 V, set **Gain** to 2.00.

This control is disabled if the option **Extract from waveform** is selected. In this case, the control displays the extracted value to apply to the signal generator.

- **Offset (V)**—Specifies the value the signal generator adds to the data.

When **Extract from waveform** is not selected, you must first normalize the data points to a range of -1.00 to +1.00. Use this control to shift the arbitrary waveform range. For example, to configure the output signal to range from 0.00 to 2.00 V instead of -1.00 to 1.00 V, set **Offset (V)** to 1.00.

This control is disabled if the option **Extract from waveform** is selected. In this case, the control displays the extracted value to apply to the signal generator.

- Output impedance—Specifies the output

	<ul> <li>impedance of the signal generator.</li> <li>Load impedance—Specifies the load impedance. Using this control allows the signal generator to produce the correct output signal for the most common applications: high (&gt;1 MΩ) load impedance or matched output and load impedance.</li> <li>Filters enabled—Specifies whether the onboard analog and/or digital filters are enabled.</li> </ul>
Trigger and	Contains the following options:
Timing	<ul> <li>Trigger type—Specifies the type of trigger to use for starting waveform generation. Selecting an Immediate trigger starts waveform generation as soon as all configuration settings are applied.</li> <li>Trigger source—Specifies the trigger source. The signal generator waits for the specified trigger to start waveform generation.</li> </ul>
	<ul> <li>Edge—Specifies whether to trigger on a rising</li> </ul>
	edge or a falling edge of the trigger signal.
	<ul> <li>Update clock mode—Specifies the update clock to use. In Divide-Down sampling mode, the sample rate can only be set to certain frequencies, based on dividing down the update clock. However, in High-Resolution mode, the sample rate can be set to any value. In Automatic mode, NI-FGEN determines the correct update clock depending on the selected sample rate.</li> </ul>
	<ul> <li>Reference clock source—Specifies the source of a 10 MHz reference clock signal. The function generator uses this reference clock source to derive output sample rates.</li> </ul>
	• <b>Sample clock output terminal</b> —Specifies which front panel terminal, if any, the signal generator should use to export the Sample clock signal.
	Refer to the NI Signal Generators Help to

<ul> <li>determine which terminals are valid for each signal generator.</li> <li>Max time (s)—Specifies the maximum time to wait for the finite waveform generation to complete.</li> </ul>
---

## **Block Diagram Inputs**

Parameter	Description
close	<ul> <li>Determines whether the driver session remains open when the VI finishes execution. Use this parameter for loop optimization by setting it to FALSE on all iterations other than the last iteration.</li> <li>Note This input is not intended to be used to share the session between Express Vis. If you have a loop containing multiple Express Vis that use the same device, you must wire in TRUE for this input.</li> </ul>
data	Specifies the arbitrary waveform to generate. This input is not available when using the <b>Waveform from file</b> option in the NI-FGEN Express (Arb) VI configuration page. If the <b>Extract from waveform</b> option is not checked in the configuration page, the data values must be in the range -1.0 to +1.0.
max time (sec)	Specifies the maximum time to wait for the finite waveform generation to complete. If you set this value to -1, the VI waits indefinitely.
error in	Describes error conditions that occurred prior to running this Express VI.

### **Block Diagram Outputs**

Parameter	Description
error out	Contains error information. If <b>error in</b> indicates that an error occurred before this Express VI ran, <b>error out</b> contains the same error information. Otherwise, it describes the error status that this Express VI produces.

## Arbitrary Sequence Output:Arbitrary Sequence Handle

Short Name: Arbitrary Sequence Handle

This channel-based property selects which sequence the signal generator produces. You can create multiple sequences using the <u>niFgen</u> <u>Create Arbitrary Sequence VI</u>. The <u>niFgen Create Arbitrary Sequence VI</u> returns a handle that you use to identify the particular sequence. To configure the signal generator to produce a particular sequence, you set this property to the sequence's handle. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_SEQ.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Arbitrary Sequence VI

## Arbitrary Sequence Output:Max Loop Count

#### Short Name: Max Loop Count

Returns the maximum number of times the signal generator can repeat a waveform in a sequence. Typically, this value is constant for the signal generator.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Sequence Capabilities VI

## Arbitrary Sequence Output:Max Number of Sequences

Short Name: Max Number of Sequences

Returns the maximum number of arbitrary sequences the signal generator allows.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Sequence Capabilities VI

# Arbitrary Sequence Output:Max Sequence Length

Short Name: Max Sequence Length

Returns the maximum number of arbitrary waveforms the signal generator allows in a sequence.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Sequence Capabilities VI

# Arbitrary Sequence Output:Min Sequence Length

Short Name: Min Sequence Length

Returns the minimum number of arbitrary waveforms the signal generator allows in a sequence. Typically, this value is constant for the signal generator.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Sequence Capabilities VI

## Arbitrary Waveform Output:Arbitrary Waveform Gain

Short Name: Arbitrary Waveform Gain

Specifies the factor by which the signal generator scales the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to the range -1.0 to +1.0. Use this property to scale the arbitrary waveform to other ranges. For example, when you set this property to 2.0, the output signal ranges from -2.0 to +2.0 V.

Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB, NIFGEN\_VAL\_OUTPUT\_SEQ, or NIFGEN\_VAL\_OUTPUT\_SCRIPT.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VIs	niFgen Configure Arbitrary Waveform VI niFgen Configure Arbitrary Sequence VI

## Arbitrary Waveform Output:Arbitrary Waveform Handle

Short Name: Arbitrary Waveform Handle

Selects which arbitrary waveform the signal generator produces. You can create multiple arbitrary waveforms using the <u>niFgen Create Waveform</u> VI. The <u>niFgen Create Waveform VI</u>, <u>niFgen Allocate Waveform VI</u>, and similar VIs return a handle that you use to identify the particular waveform. To configure the signal generator to produce a particular waveform, you set this property to the waveform's handle. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Arbitrary Waveform Marker Position

Short Name: ArbWfm.MarkerPos

Specifies the position for a marker to be asserted in the arbitrary waveform. This property defaults to -1 when no marker position is specified. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB.

Use the <u>niFgen Export Signal VI</u> to export the marker signal.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Arbitrary Waveform Output:Arbitrary Waveform Offset

Short Name: Arbitrary Waveform Offset

Specifies the value the signal generator adds to the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to the range -1.0 to +1.0. Use this property to shift the arbitrary waveform's range. The units are volts (V). For example, when you set this property to 1.0, the output signal ranges from 0.0 volts to 2.0 volts. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB, NIFGEN\_VAL\_OUTPUT\_SCRIPT, or NIFGEN\_VAL\_OUTPUT\_SEQ.
Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VIs	niFgen Configure Arbitrary Waveform VI niFgen Configure Arbitrary Sequence VI

## Arbitrary Waveform Output:Arbitrary Waveform Repeat Count

Short Name: ArbWfm.RepeatCount

Specifies number of times to repeat the arbitrary waveform when the **Trigger Mode** parameter of the <u>nifgen Configure Trigger Mode VI</u> is set to Single or Stepped. This property is ignored if **Trigger Mode** is set to Continuous or Burst. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB.

When used during <u>streaming</u>, this property specifies the number of times to repeat the streaming waveform (the onboard memory allocated for streaming).

Default Value: 1

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Arbitrary Waveform Output:Data Transfer:Data Transfer Block Size

Short Name: Data Transfer Block Size

The number of samples at a time to download to onboard memory. Useful when the total data to be transferred to onboard memory is large.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Arbitrary Waveform Output:Data Transfer:Direct DMA:Direct DMA Enabled

Short Name: Direct DMA Enabled

Enable the device for Direct DMA writes. When enabled, all <u>niFgen</u> <u>Create Waveform VI</u> and <u>niFgen Write Waveform VI</u> calls that are given a data address in the Direct DMA Window download data residing on the Direct DMA device to the instrument's onboard memory.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Arbitrary Waveform Output:Data Transfer:Direct DMA:Window Address

Short Name: Direct DMA Window Address

Specifies the window address (beginning of window) of the waveform data source. This window address is specified by your Direct DMA-compatible data source.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Arbitrary Waveform Output:Data Transfer:Direct DMA:Window Size in Bytes

Short Name: Direct DMA Window Size

Specifies the size of the memory window in bytes (not samples) provided by your Direct DMA-compatible data source.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Data Transfer:File Transfer Block Size

Short Name: File Transfer Block Size

The number of samples at a time to read from the file and download to onboard memory. Used in conjunction with the <u>niFgen Create Waveform</u> <u>From File</u> and <u>niFgen Write Waveform From File</u> VIs.

If the requested value is not evenly divisible by the required increment, this property is coerced up to the next 64-sample increment (32-sample increment for complex samples).

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Data Transfer:Streaming:Streaming Waveform Handle

Short Name: Streaming Waveform Handle

Specifies the waveform handle of the waveform used to continuously stream data during generation. This property defaults to -1 when no streaming waveform is specified.

Used in conjunction with the <u>Space Available in Streaming Waveform</u> property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Arbitrary Waveform Output:Data Transfer:Streaming:Space Available in Streaming Waveform

Short Name: SpaceAvailInStreamingWfm

Indicates the space available (in samples) in the streaming waveform for writing new data.During generation, this available space may be in multiple locations with, for example, part of the available space at the end of the streaming waveform and the rest at the beginning. In this situation, writing a block of waveform data the size of the total space available in the streaming waveform causes NI-FGEN to return an error, as NI-FGEN will not wrap the data from the end of the waveform to the beginning and cannot write data past the end of the waveform buffer.

To avoid writing data past the end of the waveform, write new data to the waveform in a fixed size that is an integer divisor of the total size of the streaming waveform.

Used in conjunction with the <u>Streaming Waveform Handle</u> or <u>Streaming</u> <u>Waveform Name</u> property.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Arbitrary Waveform Output:Max Number of Waveforms

Short Name: Max Number of Waveforms

Returns the maximum number of arbitrary waveforms that the signal generator allows. On some signal generators this may vary with remaining onboard memory.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Waveform Capabilities VI

## **Arbitrary Waveform Output: Max Waveform Size**

#### Short Name: Max Waveform Size

Returns the maximum number of points the signal generator allows in an arbitrary waveform. On some signal generators this may vary with remaining onboard memory.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Waveform Capabilities VI

## **Arbitrary Waveform Output: Min Waveform Size**

#### Short Name: Min Waveform Size

Returns the minimum number of points the signal generator allows in an arbitrary waveform. Typically, this value is constant for the signal generator.



**Note** In some cases you may need to supply a larger waveform than the value specified by this property. Refer to the *NI Signal Generators Help* for a table of minimum waveform sizes.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Waveform Capabilities VI

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:CIC Filter Enabled

Short Name: CIC Filter Enabled

Enables/disables the CIC filter.



**Note** The CIC Filter Enabled and <u>FIR Filter Enabled</u> properties must have the same enable/disable setting.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:CIC Filter Gain

Short Name: CIC Filter Gain

Specifies the gain applied at the final stage of the CIC filter. Commonly used to compensate for attenuation in the FIR filter. For FIR filter types other than Custom, NI-FGEN calculates the CIC gain in order to achieve unity gain between the FIR and CIC filters. Setting this property overrides the value set by NI-FGEN.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:CIC Interpolation Factor

Short Name: CIC Interpolation Factor

Specifies the interpolation factor for the CIC filter. If you do not set this value, NI-FGEN calculates the appropriate value based on the value of the <u>IQ Rate</u> property.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:FIR Filter Enabled

Short Name: FIR Filter Enabled

Enables/disables the FIR filter.



**Note** The <u>CIC Filter Enabled</u> and FIR Filter Enabled properties must have the same enable/disable setting.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:FIR Interpolation Factor

Short Name: FIR Interpolation Factor

Specifies the interpolation factor for the FIR filter. If you do not set this value, NI-FGEN calculates the appropriate value based on the value of the <u>IQ Rate</u> property.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:OSP Overflow Error Reporting

Short Name: OSP Overflow Error Reporting

Configures error reporting when the OSP block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.

You can use the <u>OSP Overflow Status</u> property to query for overflow conditions whether or not the OSP Overflow Error Reporting property is enabled. The device will continue to generate after an overflow whether or not the OSP Overflow Error Reporting property is enabled.

#### Defined Values:

Error	NI-FGEN returns errors whenever an overflow has occurred in the onboard signal processing block.
Disabled	NI-FGEN does not return errors when an overflow occurs in the onboard signal processing block.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:Advanced:OSP Overflow Status

Short Name: OSP Overflow Status

Returns a bit field of the overflow status in any stage of the OSP block. This property is functional regardless of the value for the <u>OSP Overflow</u> <u>Error Reporting</u> property.



**Note** This property can be used only with signal generators that support onboard signal processing. NI-FGEN returns an error if you use this property with a device that does not support onboard signal processing.

Write 0 to this property to clear the current OSP Overflow Status.

#### Defined Values:

NIFGEN_VAL_OSP_OVERFLOW_NONE	0x0
NIFGEN_VAL_OSP_OVERFLOW_PRE_FILTER_GAIN_I	0x1
NIFGEN_VAL_OSP_OVERFLOW_PRE_FILTER_GAIN_Q	0x2
NIFGEN_VAL_OSP_OVERFLOW_PRE_FILTER_OFFSET_I	0x4
NIFGEN_VAL_OSP_OVERFLOW_PRE_FILTER_OFFSET_Q	0x8
NIFGEN_VAL_OSP_OVERFLOW_FIR_FILTER_I	0x10
NIFGEN_VAL_OSP_OVERFLOW_PFIR_FILTER_I	0x10
NIFGEN_VAL_OSP_OVERFLOW_FIR_FILTER_Q	0x20
NIFGEN_VAL_OSP_OVERFLOW_PFIR_FILTER_Q	0x20
NIFGEN_VAL_OSP_OVERFLOW_CIC_FILTER_I	0x40
NIFGEN_VAL_OSP_OVERFLOW_CIC_FILTER_Q	0x80
NIFGEN_VAL_OSP_OVERFLOW_COMPLEX_DATA	0x100
NIFGEN_VAL_OSP_OVERFLOW_CFIR_FILTER_I	0x200
NIFGEN_VAL_OSP_OVERFLOW_CFIR_FILTER_Q	0x400
NIFGEN_VAL_OSP_OVERFLOW_EQUALIZER	0x800
Data Type	Vilnt32
---------------	---------
Permissions	R/W
Channel Based	No
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:Carrier Enabled

Short Name: Carrier Enabled

Enables/disables generation of the carrier.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:Carrier Frequency

Short Name: Carrier Frequency

Specifies the frequency of the generated carrier.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Arbitrary Waveform Output:Onboard Signal Processing:Data Processing Mode

Short Name: Data Processing Mode

Specifies the way in which data is processed by the OSP block.

Defined Values:

Real	The waveform data points are real numbers (I data).
Complex	The waveform data points are complex numbers (IQ data).

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Arbitrary Waveform Output:Onboard Signal Processing:FIR Filter:Filter Type

Short Name: Filter Type

Specifies the pulse-shaping filter type for the FIR filter.

#### Defined Values:

Flat	Applies a flat filter to the data with the passband value specified in the Flat Filter Passband property.
Raised Cosine	Applies a raised cosine filter to the data with the alpha value specified in the <u>Raised Cosine Filter Alpha</u> property.
Root Raised Cosine	Applies a root raised cosine filter to the data with the alpha value specified in the <u>Root Raised Cosine Filter Alpha</u> property.
Gaussian	Applies a Gaussian filter to the data with the BT value specified in the Gaussian Filter BT property.
Custom	Applies a custom filter to the data. If Custom is selected, you must provide a set of FIR filter coefficients with the <u>niFgen</u> <u>Configure Custom FIR Filter Coefficients VI</u> .

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:FIR Filter:Flat:Passband

Short Name: Flat Filter Passband

Specifies the passband value to use when calculating the FIR filter coefficients. The FIR filter is flat to passband  $\times$  IQ rate. This property is only used when the <u>Filter Type</u> property is set to Flat.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:FIR Filter:Gaussian:BT

Short Name: Gaussian Filter BT

Specifies the BT value to use when calculating the pulse-shaping FIR filter coefficients. Only used when the <u>Filter Type</u> property is set to Gaussian.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:FIR Filter:Raised Cosine:Alpha

Short Name: Raised Cosine Filter Alpha

Specifies the alpha value to use when calculating the pulse shaping FIR filter coefficients. Only used when the <u>Filter Type</u> property is set to Raised Cosine.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:FIR Filter:Root Raised Cosine:Alpha

Short Name: Root Raised Cosine Filter Alpha

Specifies the alpha value to use when calculating the pulse-shaping FIR filter coefficients. Only used when the <u>Filter Type</u> property is set to Root Raised Cosine.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:IQ Rate

Short Name: IQ Rate

Specifies the rate at which the user-provided waveform data is generated when the <u>OSP Enabled</u> property is set to True. NI-FGEN sets the <u>Sample Rate</u> property of the signal generator to the product of the IQ Rate, <u>FIR Interpolation Factor</u>, and <u>CIC Interpolation Factor</u> properties. When the <u>Data Processing Mode</u> property is set to Real, the IQ Rate is the rate at which the signal generator processes real (I) data. When Data Processing Mode is set to Complex, the IQ Rate is the rate at which the signal generator processes complex (IQ) data.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Carrier Phase:Carrier Phase I

Short Name: Carrier Phase I

Specifies the I Carrier Phase in degrees at the first point of the generation.

Default Value: 0.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Carrier Phase:Carrier Phase Q

Short Name: Carrier Phase Q

Specifies the Q Carrier Phase in degrees at the first point of the generation. This property is only used when the <u>Data Processing Mode</u> property is set to Complex.

Default Value: -90.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Gain:Prefilter Gain I

Short Name: Pre-filter Gain I

Specifies the digital gain to apply to the I data stream before any filtering by the OSP block.

Default Value: 1.0

Values: Range is from -2.0 to 2.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Gain:Prefilter Gain Q

Short Name: Pre-filter Gain Q

Specifies the digital gain to apply to the Q data stream before any filtering by the OSP block. This property is only used when the <u>Data Processing</u> <u>Mode</u> property is set to Complex.

Default Value: 1.0

Values: Range is from -2.0 to 2.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Offset:Prefilter Offset I

Short Name: Pre-filter Offset I

Specifies the digital offset to apply to the I data stream. This offset is applied after the prefilter gain and before any filtering.

Default Value: 0.0

Values: Range is from -1.0 to 1.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:IQ Signal Adjustments:Offset:Prefilter Offset Q

Short Name: Pre-filter Offset Q

Specifies the digital offset to apply to the Q data stream. This offset is applied after the prefilter gain and before any filtering. This property is only used when the <u>Data Processing Mode</u> property is set to Complex.

Default Value: 0.0

Values: Range is from -1.0 to 1.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

### Arbitrary Waveform Output:Onboard Signal Processing:OSP Enabled

Short Name: OSP Enabled

Enables/disables the OSP block of the signal generator. When the OSP block is disabled, all OSP-related properties are disabled and have no effect on the generated signal.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Arbitrary Waveform Output: Sample Rate**

#### Short Name: Sample Rate

Specifies the rate at which the signal generator outputs the points in arbitrary waveforms. The units are samples per second. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_ARB or NIFGEN\_VAL\_OUTPUT\_SEQ.

 $\overline{\mathbb{N}}$ 

**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation</u> VI or wait for the generation to complete.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	<u>niFgen Set Sample Rate VI</u>

# Arbitrary Waveform Output:Waveform Quantum

#### Short Name: Waveform Quantum

The size of each arbitrary waveform must be a multiple of a quantum value. This property returns the quantum value the signal generator allows.

For example, when this property returns a value of 4, all waveform sizes must be a multiple of 4. Typically, this value is constant for the signal generator.
Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Arb Waveform Capabilities VI

## **Basic Operation: Operation Mode**

#### Short Name: Operation Mode

Specifies how the signal generator produces waveforms. NI signal generators currently support only one value: continuous. To control trigger mode, set the <u>Trigger Mode</u> property.

Defined Value: NIFGEN\_VAL\_OPERATE\_CONTINUOUS

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## **Output Attributes:Output Enabled**

#### Short Name: Output Enabled

Specifies whether or not the signal the signal generator produces appears at the channel's output connector.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	<u>niFgen Output Enable VI</u>

# **Output Attributes:Output Impedance**

#### Short Name: Output Impedance

Specifies the impedance value the signal generator will use.

#### Default Value: 50.0

#### Defined Values:

- 50.0
- 75.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Output Impedance VI

## **Output Attributes:Output Mode**

#### Short Name: Output Mode

Sets which output mode the signal generator will use. The value you specify determines which VIs and properties you use to configure the waveform the signal generator produces.

**Default Value**: NIFGEN\_VAL\_OUTPUT\_FUNC

#### **Defined Values:**

Standard Function
Frequency List
Arbitrary Waveform
Arbitrary Sequence
Script

### **Standard Function**

When you set Output Mode to Standard Function (NIFGEN\_VAL\_OUTPUT\_FUNC), you can use the following VIs to configure the waveform:

- <u>niFgen Configure Standard Waveform VI</u>
- niFgen Define User Standard Waveform VI
- niFgen Clear User Standard Waveform VI

- <u>Waveform</u>
- <u>Amplitude</u>
- DC Offset
- Frequency
- Start Phase
- Duty Cycle High

### **Frequency List**

When you set Output Mode to Frequency List (NIFGEN\_VAL\_OUTPUT\_FREQ\_LIST), you can use the following VIs to configure the waveform:

- niFgen Create Frequency List VI
- <u>niFgen Clear Frequency List VI</u>
- <u>niFgen Query Freq List Capabilities VI</u>
- niFgen Configure Frequency List VI

- Frequency List Handle
- DC Offset
- Start Phase
- Amplitude

### **Arbitrary Waveform**

When you set Output Mode to Arbitrary Waveform (NIFGEN\_VAL\_OUTPUT\_ARB), you can use the following VIs to configure the waveform:

- niFgen Query Arb Waveform Capabilities VI
- niFgen Create Waveform (poly) VI
- niFgen Clear Arbitrary Waveform VI
- niFgen Clear Arbitrary Memory VI
- niFgen Configure Arbitrary Waveform VI
- <u>niFgen Configure Clock Mode VI</u>
- <u>niFgen Set Sample Rate VI</u>
- niFgen Allocate Waveform VI
- niFgen Set Waveform Next Write Position VI
- <u>niFgen Write Waveform (poly) VI</u>

- Arbitrary Waveform Handle
- <u>Clock Mode</u>

### **Arbitrary Sequence**

When you set Output Mode to Arbitrary Sequence (NIFGEN\_VAL\_OUTPUT\_SEQ), you can use the following VIs to configure the sequence:

- niFgen Query Arb Waveform Capabilities VI
- niFgen Create Waveform (poly) VI
- niFgen Clear Arbitrary Waveform VI
- niFgen Clear Arbitrary Memory VI
- niFgen Query Arb Sequence Capabilities VI
- niFgen Create Arbitrary Sequence VI
- niFgen Create Advanced Arb Sequence VI
- niFgen Clear Arbitrary Sequence VI
- niFgen Configure Arbitrary Sequence VI
- niFgen Configure Clock Mode VI
- <u>niFgen Set Sample Rate VI</u>
- niFgen Allocate Waveform VI
- <u>niFgen Set Waveform Next Write Position VI</u>
- niFgen Write Waveform (poly) VI

- Arbitrary Sequence Handle
- <u>Clock Mode</u>

### Script

When you set Output Mode to Script (NIFGEN VAL OUTPUT SCRIPT), you can use the following VIs to configure the script:

- niFgen Allocate Named Waveform VI
- niFgen Set Named Waveform Next Write Position VI
- niFgen Write Named Waveform VI
- <u>niFgen Delete Named Waveform VI</u>
- <u>niFgen Write Script VI</u>
- niFgen Delete Script

You can also set the following property:

• Script To Generate



Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the niFgen Abort Generation VI or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Calibration:Advanced:DAQmx Task

Short Name: DAQmx Task

The NI-DAQmx task that NI-FGEN uses for the NI 5421. For internal use only.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

## Calibration:Advanced:External Clock Delay Binary Value

**Short Name:** External Clock Delay Binary Value Specifies the external clock delay binary value.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Calibration:Advanced:Oscillator Phase DAC Value

Short Name: Oscillator Phase DAC Value

Specifies the oscillator phase DAC value.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Calibration:Cal ADC Input**

#### Short Name: Cal ADC Input

Specifies the input of the calibration ADC. The ADC can take a reading from several inputs: the analog output, a 2.5 V reference, and ground.

The latter two inputs are used to calibrate the ADC itself.

#### Default Value: NIFGEN\_VAL\_ANALOG\_OUTPUT

#### **Defined Values**:

NIFGEN_VAL_ANALOG_OUTPUT	Specifies that the ADC will measure the analog output
NIFGEN_VAL_INTERNAL_VOLTAGE_REFERENCE	Specifies that the ADC will measure the internal voltage reference
NIFGEN_VAL_GROUND	Specifies that the ADC will measure the ground voltage

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## **Calibration: Flatness Correction Enabled**

Short Name: Flatness Correction Enabled

When TRUE, the signal generator applies a flatness correction factor to the generated sine wave in order to ensure the same output power level at all frequencies.

This property should be set to FALSE when performing Flatness Calibration.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Calibration:Gain DAC Value**

#### Short Name: Gain DAC Value

Specifies the value programmed to the Gain DAC. The value should be treated as an unsigned, right-justified number.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Calibration:Offset DAC Value**

#### Short Name: Offset DAC Value

Specifies the value programmed to the Offset DAC. The value should be treated as an unsigned, right-justified number.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## **Calibration:Oscillator Freq DAC Value**

Short Name: Oscillator Freq DAC Value

Specifies the value programmed to the Oscillator DAC. The value should be treated as an unsigned, right-justified number.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## **Calibration:Post-Amplifier Attenuation**

Short Name: Post-Amplifier Attenuation

Specifies the amount of postamplifier attenuation that should be applied to the signal (in dB).



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Calibration:Pre-Amplifier Attenuation**

Short Name: Pre-Amplifier Attenuation

Specifies the amount of preamplifier attenuation that should be applied to the signal.

Units: decibels



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

## **Clock Attributes: Actual Arb Sample Rate**

#### Short Name: Actual Arb Sample Rate

This property is the actual sample rate value of the signal generator after any coercion or rounding.

Data Type	ViReal64
Permissions	RO
Channel Based	No
High-Level VI	None
## Clock Attributes:Advanced:Sample Clock Absolute Delay

Short Name: Sample Clock Absolute Delay

The delay in seconds to apply to an external sample clock. This property is useful when trying to align the output of two devices.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Clock Attributes:Clock Mode**

#### Short Name: Clock Mode

Specifies the clock mode for the signal generator.

For signal generators that support it, this allows switching the sample clock to a high resolution clocking mode. When in divide down sampling mode, the sample rate can only be set to certain frequencies, based on dividing down the update clock. However, in high resolution mode, the sample rate may be set to any value.

#### **Default Value**: NIFGEN\_VAL\_DIVIDE\_DOWN

#### Defined Values:

NIFGEN_VAL_DIVIDE_DOWN	<b>Divide down sampling</b> —Sample rate is generated by dividing the source frequency.
NIFGEN_VAL_HIGH_RESOLUTION	<b>High resolution sampling</b> — Sample rate is generated by a high resolution clock source.
NIFGEN_VAL_AUTOMATIC	Automatic Selection—NI-FGEN selects between the divide-down and high-resolution modes.

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Clock Mode VI

## Clock Attributes:Onboard Reference Clock:Export Output Terminal

**Short Name:** Exported Onboard Reference Clock Output Terminal Specifies the terminal to export the Onboard Reference Clock.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Clock Attributes: Reference Clock Frequency**

Short Name: Reference Clock Frequency

Specifies the frequency of the reference clock that is used in the phaselocked loop to tune the sample clock timebase.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation</u> VI or wait for the generation to complete.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Reference Clock VI

# Clock Attributes:Reference Clock:Export Output Terminal

Short Name: Exported Reference Clock Output Terminal

Specifies the terminal to export the Reference Clock.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Clock Attributes:Reference Clock Source**

#### Short Name: Reference Clock Source

Specifies the source for the reference clock. For example, when you set this attribute to "ClkIn," the signal generator uses the signal it receives at its Clk In front panel connector as the reference clock.

The reference clock at the specified source phase-locks with the signal generator's sample clock timebase to allow the frequency stability and accuracy of the sample clock timebase to match that of the reference clock.

Notes The signal generator must not be in the Generating state when you change this property. To change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

The following defined values are examples of possible reference clock sources. For a complete list of the reference clock sources available on your device, refer to the **Device Routes** tab in MAX.

#### **Defined Values**

"None"	No Reference clock source
"PXI_Clk10"	10 MHz reference clock signal provided by the PXI bus
"ClkIn"	External signal on the Clk In front panel connector
"OnboardRefClk"	Dedicated 10 MHz clock for PCI modules
"RTSI7"	RTSI line 7
"Refln"	External signal on the Ref In front panel connector

#### Default Value: "None"

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Reference Clock VI

## Clock Attributes:Sample Clock Timebase:Export Output Terminal

Short Name: Exported Sample Clock Timebase Output Terminal

Specifies the terminal to export the Sample Clock Timebase.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.



**Note** The signal generator must not be in the Generating state when you change this property. To change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# Clock Attributes:Sample Clock:Export Output Terminal

Short Name: Exported Sample Clock Output Terminal

Specifies the terminal to export the Sample Clock.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Clock Attributes:Sample Clock:Source**

Short Name: Sample Clock Source

Specifies the Sample Clock source.



**Notes** The signal generator must not be in the Generating state when you change this property. To change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

The following defined values are examples of possible sample clock sources. For a complete list of the sample clock sources available on your device, refer to the **Device Routes** tab in MAX.

#### **Defined Values**

"OnboardClock"	Onboard clock
"ClkIn"	Clk In front panel connector
"PXI_Star"	PXI Star line
"PXI_Trig0"	PXI or RTSI line 0
"PXI_Trig1"	PXI or RTSI line 1
"PXI_Trig2"	PXI or RTSI line 2
"PXI_Trig3"	PXI or RTSI line 3
"PXI_Trig4"	PXI or RTSI line 4
"PXI_Trig5"	PXI or RTSI line 5
"PXI_Trig6"	PXI or RTSI line 6
"PXI_Trig7"	PXI or RTSI line 7
"DDC_ClkIn"	Sample clock from DDC connector

Default Value: "OnboardClock"

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# Frequency List Output:Frequency List Duration Quantum

Short Name: Frequency List Duration Quantum

Returns the quantum of which all durations must be a multiple in a frequency list.

Data Type	ViReal64
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Freq List Capabilities VI

# **Frequency List Output: Frequency List Handle**

#### Short Name: Frequency List Handle

Specifies which frequency list the signal generator will produce. You create a frequency list using the <u>niFgen Create Frequency List VI</u>. The <u>niFgen Create Frequency List VI</u> returns a handle that you use to identify the list.

#### Default Value: None

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Frequency List VI

# Frequency List Output:Maximum Frequency List Duration

Short Name: Maximum Frequency List Duration

This property is the maximum duration in seconds of any one step in the frequency list.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViReal64
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Freq List Capabilities VI

# Frequency List Output:Maximum Frequency List Length

Short Name: Maximum Frequency List Length

This property is the maximum number of steps that can be in a frequency list.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Freq List Capabilities VI

## Frequency List Output:Maximum Number Of Frequency Lists

Short Name: Maximum Number Of Frequency Lists

This property is the maximum number of frequency lists the signal generator allows.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Freq List Capabilities VI

# Frequency List Output:Minimum Frequency List Duration

Short Name: Minimum Frequency List Duration

Returns the minimum duration in seconds of any one step in a frequency list.

Data Type	ViReal64
Permissions	RO
Channel Based	No
High-Level VI	niFgen Query Freq List Capabilities VI

# Frequency List Output:Minimum Frequency List Length

Short Name: Minimum Frequency List Length

Returns the minimum number of frequency lists the signal generator allows.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Advanced Session Information:Logical Name

#### Short Name: Logical Name

A string containing the logical name you specified when opening the current IVI session. You may pass a logical name to the <u>niFgen Initialize</u> or <u>niFgen Initialize With Options VIs</u>. The IVI Configuration utility must contain an entry for the logical name. The logical name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Advanced Session Information:Resource Descriptor

Short Name: Resource Descriptor

Indicates the resource descriptor NI-FGEN uses to identify the physical device. If you initialize NI-FGEN with a logical name, this property contains the resource descriptor that corresponds to the entry in the IVI Configuration utility. If you initialize NI-FGEN with the resource descriptor, this property contains that value.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None
# Inherent IVI Attributes:Driver Capabilities:Channel Count

Short Name: Channel Count

Indicates the number of channels that NI-FGEN supports. For each property for which the IVI\_VAL\_MULTI\_CHANNEL flag property is set, the IVI engine maintains a separate cache value for each channel.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Capabilities:Class Group Capabilities

Short Name: Class Group Capabilities

A string that contains a comma-separated list of class-extension groups that NI-FGEN implements.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Capabilities:Supported Instrument Models

Short Name: Supported Instrument Models

Contains a model code of the instrument. For drivers that support more than one device, this property contains a comma-separated list of supported instrument models.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Identification:Class Specification Major Version

Short Name: Class Specification Major Version

The major version number of the class specification with which NI-FGEN is compliant.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Identification:Class Specification Minor Version

Short Name: Class Specification Minor Version

The minor version number of the class specification with which NI-FGEN is compliant.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Inherent IVI Attributes:Driver Identification:Description

Short Name: Description

A string that contains a brief description of the specific driver.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Identification:Driver Prefix

Short Name: Driver Prefix

A string that contains the prefix for NI-FGEN. The name of each usercallable VI in NI-FGEN starts with this prefix.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

# Inherent IVI Attributes:Driver Identification:Driver Vendor

Short Name: Driver Vendor

A string that contains the name of the vendor that supplies NI-FGEN.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Driver Identification:Revision

Short Name: Revision

A string that contains additional version information about NI-FGEN.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Instrument Identification:Firmware Revision

Short Name: Firmware Revision

A string that contains the firmware revision information for the instrument you are currently using.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	niFgen Revision Query VI

## Inherent IVI Attributes:Instrument Identification:Manufacturer

Short Name: Manufacturer

A string that contains the name of the instrument manufacturer you are currently using.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

## Inherent IVI Attributes:Instrument Identification:Model

Short Name: Model

A string that contains the model number or name of the instrument that you are currently using.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

# Inherent IVI Attributes:User Options:Cache

#### Short Name: Cache

Specifies whether to cache the value of properties. When caching is enabled, NI-FGEN keeps track of the current instrument settings and avoids sending redundant commands to the instrument. Thus, you can significantly increase execution speed. NI-FGEN can choose always to cache or never to cache particular properties regardless of the setting of this property. The Default Value is TRUE. Use the <u>niFgen Initialize With Options VI</u> to override this value.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Initialize With Options VI

# Inherent IVI Attributes:User Options:Interchange Check

#### Short Name: Interchange Check

Specifies whether to perform interchangeability checking and log interchangeability warnings when you call VIs. The Default Value is VI\_FALSE. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. Interchangeability checking examines the attributes in a capability group only if you specify a value for at least one attribute within that group. Interchangeability warnings can occur when an attribute affects the behavior of the instrument and you have not set that attribute, or the attribute has been invalidated since you set it.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Initialize With Options VI

## Inherent IVI Attributes:User Options:Query Instrument Status

Short Name: Query Instrument Status

Specifies whether NI-FGEN queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this property to FALSE to disable status checking and maximize performance. However, the effect on NI-FGEN is minor. NI-FGEN can choose to ignore status checking for particular properties regardless of the setting of this property. The Default Value is TRUE. Use the <u>niFgen Initialize With Options VI</u> to override this value.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	None

# Inherent IVI Attributes:User Options:Range Check

#### Short Name: Range Check

Specifies whether to validate property values and VI parameters. If enabled, in some cases, NI-FGEN does extra validation of parameter values that you pass to NI-FGEN VIs. Range checking parameters is useful for debugging. After you validate your program, you can set this property to FALSE to disable range checking and maximize performance. For some parameters, NI-DCPower always validates the selected range.

The Default Value is TRUE. Use the <u>niFgen Initialize With Options VI</u> to override this value.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Initialize With Options VI

# Inherent IVI Attributes:User Options:Record Value Coercions

Short Name: Record Value Coercions

Specifies whether the IVI engine keeps a list of the value coercions it makes for ViInt32 and ViReal64 properties. The Default Value is FALSE. Use the <u>niFgen Initialize With Options VI</u> to override this value.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	None

# Inherent IVI Attributes:User Options:Simulate

#### Short Name: Simulate

Specifies whether or not to simulate NI-FGEN I/O operations. If simulation is enabled, NI-FGEN VIs perform range checking and call Ivi\_GetAttribute and Ivi\_SetAttribute VIs, but they do not perform instrument I/O. For output parameters that represent instrument data, NI-FGEN VIs return calculated values. The Default Value is FALSE. Use the niFgen Initialize With Options VI to override this value.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Initialize With Options VI
# Instrument Specific Attributes:Bus Type

#### Short Name: Bus Type

This is the bus type of the signal generator.

#### **Defined Values:**

NIFGEN\_VAL\_BUS\_INVALID NIFGEN\_VAL\_BUS\_AT NIFGEN\_VAL\_BUS\_PCI NIFGEN\_VAL\_BUS\_PXI NIFGEN\_VAL\_BUS\_PXIE NIFGEN\_VAL\_BUS\_VXI NIFGEN\_VAL\_BUS\_PCMCIA

Data Type	Vilnt32
Permissions	RO
Channel Based	Yes
High-Level VI	None

# Instrument Specific Attributes:Memory Size

Short Name: Memory Size

This is the amount of memory in bytes on the signal generator.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Instrument Specific Attributes:Serial Number

Short Name: Serial Number

This is the serial number of the signal generator.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

# Instrument Specific Attributes:Video Waveform Type

#### Short Name: Video Waveform Type

This selects the waveform type for the NI 5431 to generate. This ensures the crystal is set to the proper frequency.

#### **Defined Values**:

NIFGEN_VAL_PAL_B	PAL B Video Type
NIFGEN_VAL_PAL_D	PAL D Video Type
NIFGEN_VAL_PAL_G	PAL G Video Type
NIFGEN_VAL_PAL_H	PAL H Video Type
NIFGEN_VAL_PAL_I	PAL I Video Type
NIFGEN_VAL_PAL_M	PAL M Video Type
NIFGEN_VAL_PAL_N	PAL N Video Type
NIFGEN_VAL_NTSC_M	NTSC M Video Type

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Obsolete:Inherent IVI Attributes:Driver** Identification:Major Version

Short Name: Major Version

The major version number of NI-FGEN.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Revision Query VI

# **Obsolete:Inherent IVI Attributes:Driver** Identification:Minor Version

Short Name: Minor Version

The minor version number of NI-FGEN.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	niFgen Revision Query VI

# Obsolete:Inherent IVI Attributes:Error Info:Error Elaboration

Short Name: Error Elaboration

An optional string that contains additional information concerning the primary error condition.

Data Type	ViString
Permissions	RO
Channel Based	No
High-Level VI	None

# Obsolete:Inherent IVI Attributes:Error Info:Primary Error

#### Short Name: Primary Error

A code that describes the first error that occurred since the last call to niFgen\_GetErrorInfo on the session. The value follows the VXI*plug&play* completion code conventions. A negative value (0x80000000 or higher in hex) describes an error condition. A positive value describes a warning condition and indicates that no error occurred. A zero indicates that no error or warning occurred. The error and warning values can be status codes defined by IVI, VISA, class drivers, or specific drivers.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Obsolete:Inherent IVI Attributes:Error Info:Secondary Error

#### Short Name: Secondary Error

An optional code that provides additional information concerning the primary error condition. The error and warning values can be status codes defined by IVI, VISA, class drivers, or specific drivers. Zero indicates no additional information.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# **Clock Attributes: Update Clock Source**

Short Name: Update Clock Source

This property controls the update clock source.

Default Value: NIFGEN\_VAL\_INTERNAL

#### Defined Values:

NIFGEN_VAL_INTERNAL	Internal update clock
NIFGEN_VAL_EXTERNAL	External update clock given on the IO connector
NIFGEN_VAL_CLK_IN	Coaxial CLK IN connector on the board front panel
NIFGEN_VAL_DDC_CLK_IN	DDC CLK IN line of the Digital Data & Control connector
NIFGEN_VAL_PXI_STAR	(PXI only) PXI STAR trigger line. This choice is valid only in PXI chassis slots 3 through 15.
NIFGEN_VAL_RTSI_0	RTSI line 0
NIFGEN_VAL_RTSI_1	RTSI line 1
NIFGEN_VAL_RTSI_2	RTSI line 2
NIFGEN_VAL_RTSI_3	RTSI line 3
NIFGEN_VAL_RTSI_4	RTSI line 4
NIFGEN_VAL_RTSI_5	RTSI line 5
NIFGEN_VAL_RTSI_6	RTSI line 6
NIFGEN_VAL_RTSI_7	RTSI line 7 (PCI only)



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Update Clock Source VI

# **Output Attributes: Analog Filter Enabled**

#### Short Name: Analog Filter Enabled

This property controls whether or not the signal generator will apply an analog filter to the output signal. This property is valid in Arbitrary Waveform, Arbitrary Sequence, and Script output modes. This property can also be used in Standard Function and Frequency List output modes for user-defined waveforms.

#### Default Value: False

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Analog Filter VI

# **Output Attributes: Analog Path**

#### Short Name: Analog Path

Specifies the analog signal path that should be used. The main path allows the user to configure gain, offset, analog filter status, output impedance, and output enable. The direct path presents a much smaller gain range, and the user cannot adjust offset or the filter status. The direct path provides a smaller output range but lower distortion. The main path has two amplifier options, high- and low-gain. Setting this value to NIFGEN\_VAL\_MAIN\_ANALOG\_PATH allows NI-FGEN to choose the amplifier based on the user-specified gain.

Default Value: NIFGEN\_VAL\_MAIN\_ANALOG\_PATH

#### Defined Values:

NIFGEN_VAL_MAIN_ANALOG_PATH	Main Path
NIFGEN_VAL_DIRECT_ANALOG_PATH	Direct Path
NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	Main Path with Low Gain Amplifier
NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	Main Path with High Gain Amplifier

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Data Marker Events Count**

Short Name: Data Markers Count

Returns the number of Data Marker Events supported by the device.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# **Output Attributes:Data Mask:Analog Data Mask**

#### Short Name: Analog Data Mask

Specifies the mask to apply to the analog output. The masked data is replaced with the data in the <u>Analog Static Value</u> property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Data Mask:Analog Static Value**

#### Short Name: Analog Static Value

Specifies the static value that replaces data masked by the <u>Analog Data</u> <u>Mask</u> property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Data Mask:Digital Data Mask**

#### Short Name: Digital Data Mask

Specifies the mask to apply to the output on the digital connector. The masked data is replaced with the data in the <u>Digital Static Value</u> property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Data Mask:Digital Static Value**

### Short Name: Digital Static Value

Specifies the static value that replaces data masked by the <u>Digital Data</u> <u>Mask</u> property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Digital Filter Enabled**

#### Short Name: Digital Filter Enabled

This property controls whether or not the signal generator will apply a digital filter to the output signal. This property is valid in Arbitrary Waveform, Arbitrary Sequence, and Script output modes. This property can also be used in Standard Function and Frequency List output modes for user-defined waveforms.

#### Default Value: False

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	<u>niFgen Configure Digital Filter VI</u>
# Output Attributes:Digital Filter Interpolation Factor

Short Name: Digital Filter Interpolation Factor

This property specifies the interpolation factor when the digital filter is enabled. This property only affects the hardware when <u>Digital Filter</u> <u>Enabled</u> is true. Legal values are 2, 4, and 8.



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Digital Gain**

#### Short Name: Digital Gain

Specifies a factor by which the signal generator should digitally multiply generated data before converting it to an analog signal in the DAC. For a digital gain higher than 1.0, the product of digital gain times the generated data must be inside the range  $\pm 1.0$  (assuming floating point data). If the product exceeds these limits, the signal generator clips the output and an error results.

Some signal generators support both digital gain and an analog gain specified with the <u>Amplitude</u> property or <u>Arbitrary Waveform Gain</u> property. Digital gain has the advantage that it can be changed during generation without glitches that may occur when changing analog gains, due to relay switching. However, the output resolution of the DAC is a function of analog gain, so only analog gain makes full use of the resolution of the DAC.

#### Default Value: 1.0

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Digital Pattern Enabled**

#### Short Name: Digital Pattern Enabled

This property controls whether or not the signal generator will generate a digital pattern corresponding to the output signal.

Data Type	ViBoolean
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Digital Patterning VI

# **Output Attributes: Filter Correction Frequency**

Short Name: Filter Correction Frequency

This property controls the filter correction frequency of the analog filter. This will correct for the ripples in the analog filter frequency response at the frequency specified. For Standard Waveform Output, the filter correction frequency should be set to be the same as the frequency of the standard waveform. To have no filter correction, set this to 0 Hz.

Units: Hertz (Hz)

Default Value: 0

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	niFgen Configure Analog Filter VI

# **Output Attributes:Load Impedance**

#### Short Name: Load Impedance

Specifies the load impedance connected to the analog output of the channel. If the load impedance is set to

NIFGEN\_VAL\_MATCHED\_LOAD\_IMPEDANCE (-1.0), NI-FGEN matches the load impedance to the output impedance. NI-FGEN compensates to give the desired peak-to-peak voltage amplitude or arbitrary gain (relative to 1 V).

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Output Attributes:Sync Duty Cycle High**

#### Short Name: Sync Duty Cycle High

This property controls the duty cycle of the square wave the signal generator produces on the sync out line. You specify this property as a percentage of the time the square wave is high in each cycle.

Units: Percentage of time the waveform is high

Default Value: 50%

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## Routing and Event Configuration:Data Marker:Data Bit Number

Short Name: Data Marker Event Bit Number

Specifies the bit number to assign to the Data Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Data Marker:Level:Active Level

**Short Name:** Data Marker Event Level Polarity Specifies the output polarity of the Data Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Data Marker:Output Terminal

Short Name: Data Marker Event Output Terminal

Specifies the destination terminal for the Data Marker Event.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Advanced:Latched Status

Short Name: Done Event Latched Status

Returns the latched status of the specified Done Event.

Data Type	ViBoolean
Permissions	RO
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Advanced:Delay Value

Short Name: Done Event Delay

Specifies the amount of delay applied to a Done event with respect to the analog data from the NI signal generator. The default value is zero, which aligns the Done event with the analog output signal. A negative delay value indicates that the Done event precedes the analog data, while a positive delay value indicates that the Done event follows the analog data.

You can specify the units of the delay value by setting the <u>Done Event</u> <u>Delay Units</u> property.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Level:Active Level

Short Name: Done Event Active Level

Specifies the output polarity of the Done Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Done:Output Behavior

Short Name: Done Event Output Behavior

Specifies the output behavior for the Done Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Output Terminal

Short Name: Done Event Output Terminal

Specifies the destination terminal for the Done Event.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Pulse:Polarity

**Short Name:** Done Event Pulse Polarity Specifies the output polarity of the Done Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Pulse:Width Units

**Short Name:** Done Event Pulse Width Units Specifies the pulse width units for the Done Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Done:Pulse:Width Value

**Short Name:** Done Event Pulse Width Value Specifies the pulse width for the Done Event.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Exported Sample Clock Divisor

#### Short Name: Exported Sample Clock Divisor

Specifies the factor by which to divide the update (sample) clock before it is exported. To export the sample clock, use the <u>niFgen Export Signal VI</u>.

#### Default Value: 1

Values: Range is from 1 to 4,096



**Note** You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None
## Routing and Event Configuration:Exported Sample Clock Timebase Divisor

Short Name: Exported Sample Clock Timebase Divisor

Specifies the factor by which to divide the board clock (sample clock timebase) before it is exported. To export the sample clock timebase, use the <u>niFgen Export Signal VI</u>.

#### Default Value: 2

Values: Range is from 2 to 4,194,304.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Advanced:All Marker Events Latched Status

Short Name: All Marker Events Latched Status

Returns a bit field of the latched status of all Marker Events.

Write 0 to this property to clear the latched status of all Marker Events.

#### Defined Values:

Marker 0	0x0
Marker 1	0x1
Marker 2	0x2
Marker 3	0x4

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Advanced:All Marker Events Live Status

Short Name: All Marker Events Live Status

Returns a bit field of the live status of all Marker Events.

#### Defined Values:

Marker 0	0x0
Marker 1	0x1
Marker 2	0x2
Marker 3	0x4

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Advanced:Latched Status

Short Name: Marker Event Latched Status

Specifies the latched status of the specified Marker Event.

Write TRUE to this property to clear the latched status of the Marker Event.

Data Type	ViBoolean
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Advanced:Live Status

Short Name: Marker Event Live Status

Returns the live status of the specified Marker Event.

Data Type	ViBoolean
Permissions	RO
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Advanced:Delay Value

Short Name: Marker Event Delay

Specifies the amount of delay applied to a Marker event with respect to the analog data from the NI signal generator. The default value is zero, which aligns the Marker event with the analog output signal. A negative delay value indicates that the Marker event precedes the analog data, while a positive delay value indicates that the Marker event follows the analog data.

You can specify the units of the delay value by setting the <u>Marker Event</u> <u>Delay Units</u> property.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Marker:Advanced:Delay Units

Short Name: Marker Event Delay Units

Specifies the units applied to the value of the <u>Marker Event Delay</u> property.

#### **Defined Values**

Seconds	Delay is specified in seconds and then coerced up by NI- FGEN to the nearest Sample clock period.
Sample Clock Periods	Delay is specified in Sample clock periods and then coerced up by NI-FGEN to the nearest Sample clock period.

#### **Default Value**

Seconds

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Marker:Output Behavior

Short Name: Marker Event Output Behavior

Specifies the output behavior for the Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Output Terminal

Short Name: Marker Event Output Terminal

Specifies the destination terminal for the Marker Event.

Refer to MAX to find the possible strings to use with the attribute. For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Pulse:Polarity

**Short Name:** Marker Event Pulse Polarity Specifies the output polarity of the Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Pulse:Width Units

**Short Name:** Marker Event Pulse Width Units Specifies the pulse width units of the Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Pulse:Width Value

**Short Name:** Marker Event Pulse Width Value Specifies the pulse width value of the Marker Event.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Marker:Toggle:Initial State

**Short Name:** Marker Event Toggle Initial State Specifies the initial state of the Marker Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Ready For Start:Advanced:Live Status

**Short Name:** Ready For Start Event Live Status Returns the live status of the specified Ready For Start Event.

Data Type	ViBoolean
Permissions	RO
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Ready For Start:Level:Active Level

**Short Name:** Ready For Start Event Active Level Specifies the output polarity of the Ready for Start Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Ready For Start:Output Terminal

**Short Name:** Ready For Start Event Output Terminal Specifies the destination terminal for the Ready for Start Event.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Advanced:Latched Status

Short Name: Started Event Latched Status

Returns the latched status of the specified Started Event.

Data Type	ViBoolean
Permissions	RO
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Advanced:Delay Value

Short Name: Started Event Delay

Specifies the amount of delay applied to a Started event with respect to the analog data from the NI signal generator. The default value is zero, which aligns the Started event with the analog output signal. A negative delay value indicates that the Started event precedes the analog data, while a positive delay value indicates that the Started event follows the analog data.

You can specify the units of the delay value by setting the <u>Started Event</u> <u>Delay Units</u> property.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None
# Routing and Event Configuration:Started:Advanced:Delay Units

Short Name: Started Event Delay Units

Specifies the units applied to the value of the <u>Started Event Delay</u> property.

#### **Defined Values**

Seconds	Delay is specified in seconds and then coerced up by NI- FGEN to the nearest Sample clock period.
Sample Clock Periods	Delay is specified in Sample clock periods and then coerced up by NI-FGEN to the nearest Sample clock period.

#### **Default Value**

Seconds

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Level:Active Level

Short Name: Started Event Active Level

Specifies the output polarity of the Started Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# Routing and Event Configuration:Started:Output Behavior

Short Name: Started Event Output Behavior

Specifies the output behavior for the Started Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Output Terminal

Short Name: Started Event Output Terminal

Specifies the destination terminal for the Started Event.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Pulse:Polarity

**Short Name:** Started Event Pulse Polarity Specifies the output polarity of the Started Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Pulse:Width Units

**Short Name:** Started Event Pulse Width Units Specifies the pulse width units for the Started Event.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Routing and Event Configuration:Started:Pulse:Width Value

**Short Name:** Started Event Pulse Width Value Specifies the pulse width value for the Started Event.

Data Type	ViReal64
Permissions	R/W
Channel Based	No
High-Level VI	None

# Script Output:Markers Count

Short Name: Markers Count

Returns the number of markers supported by the device. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_SCRIPT.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Script Output:Script To Generate

#### Short Name: Script To Generate

Specifies which script the generator produces. To configure the generator to run a particular script, set this property to the name of the script. Use the niFgen Write Script VI to create multiple scripts. Use this property when the Output Mode property is set to NIFGEN\_VAL\_OUTPUT\_SCRIPT.



**Note** The signal generator must not be in the Generating state when you change this property. To change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	ViString
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# Script Output:Script Triggers Count

Short Name: Script Triggers Count

Specifies the number of script triggers supported by the device. Use this property when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_SCRIPT.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Script Output:Streaming:Space Available in Streaming Waveform

Short Name: SpaceAvailInStreamingWfm

Indicates the space available (in samples) in the streaming waveform for writing new data. During generation, this available space may be in multiple locations with, for example, part of the available space at the end of the streaming waveform and the rest at the beginning. In this situation, writing a block of waveform data the size of the total space available in the streaming waveform causes NI-FGEN to return an error, as NI-FGEN will not wrap the data from the end of the waveform to the beginning and cannot write data past the end of the waveform buffer.

To avoid writing data past the end of the waveform, write new data to the waveform in a fixed size that is an integer divisor of the total size of the streaming waveform.

Used in conjunction with the <u>Streaming Waveform Handle</u> or <u>Streaming</u> <u>Waveform Name</u> property.

Data Type	Vilnt32
Permissions	RO
Channel Based	No
High-Level VI	None

# Script Output:Streaming:Streaming Waveform Name

Short Name: Streaming Waveform Name

Specifies the name of the waveform used to continuously stream data during generation. This property defaults to an empty string when no streaming waveform is specified.

Use in conjunction with the <u>Space Available in Streaming Waveform</u> property.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Standard Function Output: Amplitude**

#### Short Name: Amplitude

This property controls the amplitude of the standard waveform that the signal generator will produce. This value is the amplitude at the output terminal.

For example, to produce a waveform ranging from -5.00 to +5.00 volts, set the amplitude to 10.00 volts.

#### Default Value: None

Units: Volts peak-to-peak (V<sub>pk-pk</sub>)



**Note** This parameter does not affect signal generator behavior when you set the <u>Waveform</u> property to NIFGEN\_VAL\_WFM\_DC.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VIs	niFgen Configure Standard Waveform VI niFgen Configure Frequency List VI

# **Standard Function Output:Buffer Size**

#### Short Name: Buffer Size

This property contains the number of samples used in the standard function waveform buffer. This property is only valid on devices that implement Standard Function mode in software, and is read-only for all other devices.



**Notes** Refer to the <u>Standard Function Mode</u> topic for more information on the implementation of Standard Function Mode on your device.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None

## **Standard Function Output:DC Offset**

#### Short Name: DC Offset

This property controls the DC offset of the standard waveform that the signal generator will produce. This value is the offset at the output terminal. The value is the offset from ground to the center of the waveform you specify with the <u>Waveform</u> property.

For example, to configure a waveform with an amplitude of 10.00 volts to range from 0.00 volts to +10.00 volts, set the DC Offset to 5.00 volts.

#### Default Value: None

Units: Volts (V)

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VIs	niFgen Configure Standard Waveform VI niFgen Configure Frequency List VI

# **Standard Function Output: Duty Cycle High**

#### Short Name: Duty Cycle High

This property controls the duty cycle of the square wave the signal generator is producing. You specify this property as a percentage of the time the square wave is high in a cycle.

#### Default Value: 50%

Units: Percentage of time the waveform is high



**Note** This parameter only affects signal generator behavior when you set the <u>Waveform</u> property to NIFGEN\_VAL\_WFM\_SQUARE.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	None

# **Standard Function Output: Frequency**

#### Short Name: Frequency

This property controls the frequency of the standard waveform that the signal generator will produce.

#### Default Value: None

Units: Hertz (Hz)



**Notes** This parameter does not affect signal generator behavior when you set the <u>Waveform</u> property to NIFGEN\_VAL\_WFM\_DC.

For NIFGEN\_VAL\_WFM\_SINE, the range is between 0 and 16 MHz, but is between 0 and 1 MHz for all other waveforms

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Standard Waveform VI

# **Standard Function Output: Maximum Buffer Size**

#### Short Name: Maximum Buffer Size

This property sets the maximum number of samples that can be used in the standard function waveform buffer. Increasing this value may increase the quality of the waveform. This property is only valid on devices that implement Standard Function mode in software, and is readonly for all other devices.



**Note** Refer to the <u>Standard Function Mode</u> topic for more information on the implementation of Standard Function Mode on your device.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	None
### **Standard Function Output: Start Phase**

#### Short Name: Start Phase

This property controls horizontal offset of the standard waveform the signal generator will produce. You specify this property in degrees of one waveform cycle.

A start phase of 180 degrees means output generation begins halfway through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle, which is identical to a start phase of 0 degrees.

#### Default Value: None

Units: Degrees of one cycle



**Note** This parameter does not affect signal generator behavior when you set the <u>Waveform</u> property to NIFGEN\_VAL\_WFM\_DC.

Data Type	ViReal64
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Standard Waveform VI niFgen Configure Frequency List VI

### Standard Function Output:Sync Out Output Terminal

Short Name: Sync Out Output Terminal

Specifies the terminal to export the Sync Out signal. This property is not supported for all devices.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## **Standard Function Output:Waveform**

#### Short Name: Waveform

This channel-based property specifies which standard waveform the signal generator produces. Use this property only when the <u>Output Mode</u> property is set to NIFGEN\_VAL\_OUTPUT\_FUNC.

#### Default Value: NIFGEN\_VAL\_WFM\_SINE

#### **Defined Values:**

NIFGEN_VAL_WFM_SINE	Sinusoid waveform
NIFGEN_VAL_WFM_SQUARE	Square waveform
NIFGEN_VAL_WFM_TRIANGLE	Triangle waveform
NIFGEN_VAL_WFM_RAMP_UP	Positive ramp waveform
NIFGEN_VAL_WFM_RAMP_DOWN	Negative ramp waveform
NIFGEN_VAL_WFM_DC	Constant voltage
NIFGEN_VAL_WFM_NOISE	White noise
NIFGEN_VAL_WFM_USER	User-defined waveform, defined by niFgen Define User Standard Waveform VI

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Standard Waveform VI

# Triggering:Script:Digital Edge:Edge

Short Name: Script Trigger Digital Edge - Edge

Specifies the active edge for the Script Trigger. This property is used when the <u>Script Trigger Type</u> property is set to Digital Edge.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## **Triggering:Script:Digital Edge:Source**

Short Name: Script Trigger Digital Edge - Source

Specifies the source terminal for the Script Trigger. This property is used when the <u>Script Trigger Type</u> property is set to Digital Edge.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Triggering:Script:Digital Level:Source

Short Name: Script Trigger Digital Level - Source

Specifies the source terminal for the Script Trigger. This property is used when the <u>Script Trigger Type</u> property is set to Digital Level.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

## Triggering:Script:Digital Level:Trigger When

**Short Name:** Script Trigger Digital Level - Trigger When Specifies the active level for the Script Trigger. This property is used when the <u>Script Trigger Type</u> property is set to Digital Level.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Triggering:Script:Export Output Terminal

Short Name: Exported Script Trigger Output Terminal

Specifies the output terminal for the exported Script Trigger.

Setting this property to an empty string means that when you commit the session, the signal is removed from that terminal and, if possible, the terminal is tristated.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# Triggering:Script:Trigger Type

#### Short Name: Script Trigger Type

Specifies the Script Trigger type. Depending upon the value of this property, additional properties may need to be configured to fully configure the trigger.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

## Triggering:Start:Digital Edge:Edge

Short Name: Start Trigger Digital Edge - Edge

Specifies the active edge for the Start Trigger. This property is used only when the <u>Start Trigger Type</u> property is set to Digital Edge.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

### **Triggering:Start:Digital Edge:Source**

Short Name: Start Trigger Digital Edge - Source

Specifies the source terminal for the Start Trigger. This property is used only when the <u>Start Trigger Type</u> property is set to Digital Edge.

You can specify any valid source terminal for this property. Valid sources can be found in Measurement & Automation Explorer under the Device Routes tab.

Source terminals can be specified in two ways. If your device is named Dev1 and your terminal is PFI0, then the terminal can be specified as a fully qualified terminal name, "/Dev1/PFI0". You can also specify the terminal using PFI 0.

The following table provides common relative terminal names:

PFI0	PFI 0 on the front panel connector	
PFI1-PFI3	PFI <13> on the front panel DDC connector	
PXI_Trig0PXI_Trig7	PXI_TRIG<07>	
PXI_STAR	PXI_STAR trigger line supported in Slots 3–15	

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Triggering:Start:Export Output Terminal**

**Short Name:** Exported Start Trigger Output Terminal Specifies the destination terminal for exporting the Start Trigger. For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

Data Type	ViString
Permissions	R/W
Channel Based	No
High-Level VI	None

# Triggering:Start:Trigger Type

#### Short Name: Start Trigger Type

Specifies whether you want the Start Trigger to be a Digital Edge, or Software trigger. You can also choose 'None' as the value for this property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None

# **Triggering:Synchronization Source**

Short Name: Synchronization Source

Specify the source of the synchronization signal you want to use.

Default Value: NIFGEN\_VAL\_NONE

#### Defined Values:

NIFGEN_VAL_NONE	No Synchronization Source
NIFGEN_VAL_RTSI_0	RTSI line 0
NIFGEN_VAL_RTSI_1	RTSI line 1
NIFGEN_VAL_RTSI_2	RTSI line 2
NIFGEN_VAL_RTSI_3	RTSI line 3
NIFGEN_VAL_RTSI_4	RTSI line 4
NIFGEN_VAL_RTSI_5	RTSI line 5
NIFGEN_VAL_RTSI_6	RTSI line 6
NIFGEN_VAL_TTL0	PXI TRIG0 or VXI TTL0
NIFGEN_VAL_TTL1	PXI TRIG1 or VXI TTL1
NIFGEN_VAL_TTL2	PXI TRIG2 or VXI TTL2
NIFGEN_VAL_TTL3	PXI TRIG3 or VXI TTL3
NIFGEN_VAL_TTL4	PXI TRIG4 or VXI TTL4
NIFGEN_VAL_TTL5	PXI TRIG5 or VXI TTL5
NIFGEN_VAL_TTL6	PXI TRIG6 or VXI TTL6
NIFGEN_VAL_TTL7	PXI TRIG7 or VXI TTL7

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI None	

# **Triggering:Trigger Mode**

Short Name: Trigger Mode

Controls the trigger mode.



**Note** Refer to **Devices**»<**your device**>»**Triggering**»**Triggering Mode** in the *NI Signal Generators Help* for descriptions of the signal generator's specific behavior for supported trigger modes.

#### Default Value: Continuous

#### **Defined Values**:

NIFGEN_VAL_SINGLE	Single Trigger Mode
NIFGEN_VAL_CONTINUOUS	Continuous Trigger Mode
NIFGEN_VAL_STEPPED	Stepped Trigger Mode
NIFGEN_VAL_BURST	Burst Trigger Mode

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Trigger Mode VI

# **Triggering: Trigger Source**

#### Short Name: Trigger Source

This property controls which trigger source the signal generator uses.

After you call the <u>niFgen Initiate Generation VI</u>, the signal generator waits for the trigger you specify in this parameter. After it receives a trigger, the signal generator produces the number of cycles you specify in the <u>Repeat Count</u> property.

This is also the source for the trigger in the other trigger modes as specified by the <u>Trigger Mode</u> property.

**Default Value**: NIFGEN\_VAL\_IMMEDIATE

#### **Defined Values:**

Immediate—The signal generator does not wait for a trigger of any kind.
<b>External</b> —The signal generator waits for a trigger on the external trigger input.
PFI 0
PFI 1
PFI 2
PFI 3
<b>Software Trigger</b> —The signal generator waits until you call the niFgen Send Software Trigger VI.
RTSI line 0
RTSI line 1
RTSI line 2
RTSI line 3
RTSI line 4
RTSI line 5
RTSI line 6

NIFGEN_VAL_RTSI_7	RTSI line 7
NIFGEN_VAL_PXI_STAR	PXI Star
NIFGEN_VAL_TTL0	PXI TRIG0 or VXI TTL0
NIFGEN_VAL_TTL1	PXI TRIG1 or VXI TTL1
NIFGEN_VAL_TTL2	PXI TRIG2 or VXI TTL2
NIFGEN_VAL_TTL3	PXI TRIG3 or VXI TTL3
NIFGEN_VAL_TTL4	PXI TRIG4 or VXI TTL4
NIFGEN_VAL_TTL5	PXI TRIG5 or VXI TTL5
NIFGEN_VAL_TTL6	PXI TRIG6 or VXI TTL6

Note You cannot change this property while the device is generating a waveform. If you want to change the device configuration, call the <u>niFgen Abort Generation VI</u> or wait for the generation to complete.

Data Type	Vilnt32
Permissions	R/W
Channel Based	Yes
High-Level VI	niFgen Configure Trigger VI

### NI-FGEN Examples for LabVIEW

NI-FGEN ships with several examples for use with LabVIEW. These examples can help you get started developing an application, and they illustrate how to perform the most common operations with NI signal generators.

#### **Finding the Examples**

You can locate these examples in LabVIEW by selecting **Help\*Find Examples** to open the NI Example Finder, or you can navigate to the location of the example files specified in the <u>NI-FGEN Instrument Driver</u> <u>Readme</u>.

The NI Example Finder allows you to browse and search installed examples and examples on the <u>NI Developer Zone</u>.

### **Help for Examples**

For more information about these examples, use one of the following methods:

- Right-click on the VI icon in the upper right corner of the VI, and select **VI Properties**. In the Category drop-down list box, select **Documentation**.
- To view the help for individual controls and indicators or the VI, select **Help LabVIEW Context Help** to launch the Context Help window, and then place the cursor over a control.
## Routing and Event Configuration:Done:Advanced:Delay Units

Short Name: Done Event Delay Units

Specifies the units applied to the value of the **Done Event Delay** property.

## **Defined Values**

Seconds	Delay is specified in seconds and then coerced up by NI- FGEN to the nearest Sample clock period.
Sample Clock Periods	Delay is specified in Sample clock periods and then coerced up by NI-FGEN to the nearest Sample clock period.

**Default Value** 

Seconds

## Remarks

The following table lists the characteristics of this property.

Data Type	Vilnt32
Permissions	R/W
Channel Based	No
High-Level VI	None