

NI-FGEN C Function Reference

July 2007, 370449J-01

This help file contains information about the NI-FGEN functions, attributes, and values that you can use when programming an application on an NI signal generator.

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

To download the latest versions of NI signal generator documentation and examples, visit ni.com/instruments.

To comment on National Instruments documentation, refer to the [National Instruments Web site](#).

© 2001–2007 National Instruments Corporation. All rights reserved.

NI-FGEN Functions by Group

Class/Panel Name	Function Name
Initialize	niFgen_init
Initialize With Options	niFgen_InitWithOptions
Commit	niFgen_Commit
Reset	niFgen_reset
Close	niFgen_close
Configuration Functions	
Configure Operation Mode	niFgen_ConfigureOper
Configure Output Mode	niFgen_ConfigureOutpu
Configure Output Enabled	niFgen_ConfigureOutpu
Configure Output Impedance	niFgen_ConfigureOutpu
Enable Analog Filter	niFgen_EnableAnalogF
Disable Analog Filter	niFgen_DisableAnalogF
Enable Digital Filter	niFgen_EnableDigitalFil
Disable Digital Filter	niFgen_DisableDigitalFi
Enable Digital Patterning	niFgen_EnableDigitalPa
Disable Digital Patterning	niFgen_DisableDigitalP
Standard Function Output	
Configure Standard Waveform	niFgen_ConfigureStand
Define User Standard Waveform	niFgen_DefineUserStar
Clear User Standard Waveform	niFgen_ClearUserStanc
Configure Frequency	niFgen_ConfigureFrequ
Configure Amplitude	niFgen_ConfigureAmpli
Frequency List Output	
Create Frequency List	niFgen_CreateFreqList
Configure Frequency List	niFgen_ConfigureFreqL
Clear Frequency List	niFgen_ClearFreqList

Query Freq List Capabilities	niFgen_QueryFreqListC
Arbitrary Waveform Output	
Configure Gain	niFgen_ConfigureGain
Configure Sample Rate	niFgen_ConfigureSamp
Query Arb Waveform Capabilities	niFgen_QueryArbWfmC
Create Waveform F64	niFgen_CreateWavefor
Create Waveform I16	niFgen_CreateWavefor
Create Waveform Complex F64	niFgen_CreateWavefor
Create Waveform from File I16	niFgen_CreateWavefor
Create Waveform from File F64	niFgen_CreateWavefor
Create Waveform from File HWS	niFgen_CreateWavefor
Configure Arbitrary Waveform	niFgen_ConfigureArbW
Clear Arbitrary Waveform	niFgen_ClearArbWavef
Arbitrary Sequence Output	
Query Arb Sequence Capabilities	niFgen_QueryArbSeqC
Create Arbitrary Sequence	niFgen_CreateArbSequ
Create Advanced Arb Sequence	niFgen_CreateAdvance
Configure Arbitrary Sequence	niFgen_ConfigureArbSe
Clear Arbitrary Sequence	niFgen_ClearArbSeque
Clear Arbitrary Memory	niFgen_ClearArbMemoi
Incremental Waveform Write	
Allocate Waveform	niFgen_AllocateWavefo
Set Waveform Next Write Position	niFgen_SetWaveformN
Write Waveform	niFgen_WriteWaveform
Write Waveform Complex F64	niFgen_WriteWaveform
Write Binary 16 Waveform	niFgen_WriteBinary16W
Write Complex Binary 16 Waveform	niFgen_WriteComplexB
Configure Clock	
Configure Reference Clock	niFgen_ConfigureRefer

Configure Sample Clock Source
Configure Clock Mode
Adjust Sample Clock Relative Delay

Triggering and Synchronization

Configure Digital Edge Start Trigger
Configure Software Edge Start Trigger
Disable Start Trigger
Configure Digital Edge Script Trigger
Configure Digital Level Script Trigger
Configure Software Edge Script Trigger
Disable Script Trigger
Configure Trigger Mode
Send Software Edge Trigger
Disable Script Trigger
Configure Synchronization
Export Signal

5404 Routing Function

Route Signal Out

Script Output Functions

Allocate Named Waveform
Set Named Waveform Next Write Position
Write Named Waveform F64
Write Named Waveform I16
Write Named Waveform Complex F64
Write Named Waveform Complex I16
Delete Named Waveform
Write Script
Delete Script

Configure Onboard Signal Processing

[niFgen_ConfigureSamp](#)
[niFgen_ConfigureClock](#)
[niFgen_AdjustSampleC](#)

[niFgen_ConfigureDigita](#)
[niFgen_ConfigureSoftw](#)
[niFgen_DisableStartTrig](#)
[niFgen_ConfigureDigita](#)
[niFgen_ConfigureDigita](#)
[niFgen_ConfigureSoftw](#)
[niFgen_DisableScriptTri](#)
[niFgen_ConfigureTrigge](#)
[niFgen_SendSoftwareE](#)
[niFgen_DisableScriptTri](#)
[niFgen_ConfigureSyncf](#)
[niFgen_ExportSignal](#)

[niFgen_RouteSignalOut](#)

[niFgen_AllocateNamedW](#)
[niFgen_SetNamedWavef](#)
[niFgen_WriteNamedWa](#)
[niFgen_WriteNamedWa](#)
[niFgen_WriteNamedWa](#)
[niFgen_WriteNamedWa](#)
[niFgen_DeleteNamedW](#)
[niFgen_WriteScript](#)
[niFgen_DeleteScript](#)

Configure Custom FIR Filter Coefficients
Get FIR Filter Coefficients

[niFgen_ConfigureCustomFIRFilterCoefficients](#)
[niFgen_GetFIRFilterCoefficients](#)

Set/Get/Check Attribute

Set Attribute

Set Attribute ViInt32
Set Attribute ViReal64
Set Attribute ViString
Set Attribute ViBoolean
Set Attribute ViSession

[niFgen_SetAttributeViInt32](#)
[niFgen_SetAttributeViReal64](#)
[niFgen_SetAttributeViString](#)
[niFgen_SetAttributeViBoolean](#)
[niFgen_SetAttributeViSession](#)

Get Attribute

Get Attribute ViInt32
Get Attribute ViReal64
Get Attribute ViString
Get Attribute ViBoolean
Get Attribute ViSession

[niFgen_GetAttributeViInt32](#)
[niFgen_GetAttributeViReal64](#)
[niFgen_GetAttributeViString](#)
[niFgen_GetAttributeViBoolean](#)
[niFgen_GetAttributeViSession](#)

Check Attribute

Check Attribute ViInt32
Check Attribute ViReal64
Check Attribute ViString
Check Attribute ViBoolean
Check Attribute ViSession

[niFgen_CheckAttributeViInt32](#)
[niFgen_CheckAttributeViReal64](#)
[niFgen_CheckAttributeViString](#)
[niFgen_CheckAttributeViBoolean](#)
[niFgen_CheckAttributeViSession](#)

Waveform Control

Initiate Generation
Abort Generation
Send Software Edge Trigger

[niFgen_InitiateGeneration](#)
[niFgen_AbortGeneration](#)
[niFgen_SendSoftwareEdgeTrigger](#)

Utility Functions

Get Hardware State
Wait Until Done
Is Done

[niFgen_GetHardwareState](#)
[niFgen_WaitUntilDone](#)
[niFgen_IsDone](#)

Reset With Defaults	niFgen_ResetWithDefaults
Reset Device	niFgen_ResetDevice
Get Channel Name	niFgen_GetChannelName
Disable	niFgen_Disable
Self-Test	niFgen_self_test
Revision Query	niFgen_revision_query
Coercion Info	
Get Next Coercion Record	niFgen_GetNextCoercionRecord
Error Info	
Error-Query	niFgen_error_query
Error Handler	niFgen_ErrorHandler
Error Message	niFgen_error_message
Get Error	niFgen_GetError
Clear Error	niFgen_ClearError
Locking	
Lock Session	niFgen_LockSession
Unlock Session	niFgen_UnlockSession
Interchangeability Info	
Get Next Interchange Warning	niFgen_GetNextInterchangeWarning
Clear Interchange Warnings	niFgen_ClearInterchangeWarnings
Reset Interchange Check	niFgen_ResetInterchangeCheck
Calibration	
Initialize External Calibration	niFgen_InitExtCal
Close External Calibration	niFgen_CloseExtCal
Self Calibrate	niFgen_SelfCal
Analog Output Calibration	
Initialize Analog Output Calibration	niFgen_InitializeAnalogOutputCalibration
Write Binary 16 Analog Static Value	niFgen_WriteBinary16A
Cal Adjust Main Path Pre Amp Offset	niFgen_CalAdjustMainPathPreAmpOffset

Cal Adjust Main Path Pre Amp Gain	niFgen_CalAdjustMainF
Cal Adjust Main Path Post Amp Gain And Offset	niFgen_CalAdjustMainF
Cal Adjust Direct Path Gain	niFgen_CalAdjustDirect
Cal Adjust Main Path Output Impedance	niFgen_CalAdjustMainF
Cal Adjust Direct Path Output Impedance	niFgen_CalAdjustDirect
Oscillator Calibration	
Initialize Oscillator Frequency Calibration	niFgen_InitializeOscillat
Cal Adjust Oscillator Frequency	niFgen_CalAdjustOscillat
Cal ADC Calibration	
Initialize Cal ADC Calibration	niFgen_InitializeCalADC
Write Binary 16 Analog Static Value	niFgen_WriteBinary16A
Cal Adjust Cal ADC	niFgen_CalAdjustCalADC
Read Cal ADC	niFgen_ReadCalADC
Flatness Calibration Functions	
Initialize Flatness Calibration	niFgen_InitializeFlatnes
Cal Adjust Flatness	niFgen_CalAdjustFlatne
Calibration Utility Functions	
Get Self Cal Supported	niFgen_GetSelfCalSupp
Get Self Cal Last Date And Time	niFgen_GetSelfCalLastD
Get Ext Cal Last Date And Time	niFgen_GetExtCalLastD
Get Self Cal Last Temp	niFgen_GetSelfCalLastT
Get Ext Cal Last Temp	niFgen_GetExtCalLastT
Get Ext Cal Recommended Interval	niFgen_GetExtCalReco
Change Ext Cal Password	niFgen_ChangeExtCalP
Set Cal User Defined Info	niFgen_SetCalUserDefi
Get Cal User Defined Info	niFgen_GetCalUserDefi
Get Cal User Defined Info Max Size	niFgen_GetCalUserDefi
Read Current Temperature	niFgen_ReadCurrentTem
Restore Last Ext Cal Constants	niFgen_RestoreLastExt

Obsolete Functions

Configure Trigger Source	<u>niFgen_ConfigureTrigger</u>
Create Arb Waveform	<u>niFgen_CreateArbWave</u>
Create Binary 16 Arb Waveform	<u>niFgen_CreateBinary16</u>
Configure Ref Clock Source	<u>niFgen_ConfigureRefClock</u>
Configure Ref Clock Frequency	<u>niFgen_ConfigureRefClockFrequency</u>
Configure Update Clock Source	<u>niFgen_ConfigureUpdateClock</u>

niFgen_init

```
ViStatus niFgen_init (ViRsrc resourceName, ViBoolean idQuery,  
ViBoolean resetDevice, ViSession* vi);
```

Purpose

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 the **resourceName** parameter.
- If the **IDQuery** parameter is set to VI_TRUE, this function queries the device ID and checks that the ID is valid for NI-FGEN.
- If the **resetDevice** parameter is set to VI_TRUE, this function 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 function calls.

Parameters

Input

Name	Type	Description
resourceName	ViRsrc	Specifies the resource name of the device to

Example #	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1
2	NI-DAQmx device	<i>myDAQmxDevice</i>
3	NI-DAQmx device	DAQ:: <i>myDAQmxDevice</i>
4	NI-DAQmx device	DAQ::2
5	IVI logical name or IVI virtual name	<i>myLogicalName</i>

For Traditional NI-DAQ devices, the syntax is the device number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 2. To change the name of a NI-DAQmx device in MAX, right-click on the device icon and enter a new name.

An alternate syntax for NI-DAQmx devices combines the device name and the NI-DAQmx device name, as shown in Example 3. This convention allows for the use of an NI-DAQmx device in an application that was originally designed for a traditional NI-DAQ device.

device. For example, if the application expects to rename the NI-DAQmx device to 1 in MAX and change the resource name, as shown in Example 4.

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

You can also pass in the name of an IVI logical name or a virtual name configured with the IVI Configuration Store file. See Example 5. A logical name identifies a particular instrument. A virtual name identifies a specific instance of the instrument. You can also pass in the initial settings for the session.



Caution Traditional NI-DAQ and NI-Daqmx driver session names, or virtual names, are not case-sensitive. However, all IVI logical names, are case-sensitive. If you use the name of an IVI logical name or a virtual name, you must ensure that the name you use matches the name in the IVI Configuration Store file exactly, with the case of the characters.

idQuery ViBoolean Specifies whether you want NI-FGEN to perform an ID query when it initializes the device. When you set this parameter to VI_TRUE, NI-FGEN performs an ID query on the device that you initialize. NI-FGEN supports the devices that you can initialize.

Circumstances can arise where sending an ID query is undesirable. When you set this parameter to VI_FALSE, NI-FGEN does not perform an ID query. Instead, NI-FGEN initializes the device without performing an ID query.

Defined Values

VI_TRUE	Perform ID query
VI_FALSE	Skip ID query

Default Value: VI_TRUE

resetDevice ViBoolean Specifies whether you want to reset the device after the initialization procedure.

Defined Values

VI_TRUE	Reset device
VI_FALSE	Do not reset device

Default Value: VI_TRUE

Output

Name	Type	Description
vi	ViSession*	Returns a session handle that you can use to all subsequent NI-FGEN function calls.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_InitWithOptions

```
ViStatus niFgen_InitWithOptions (ViRsrc resourceName, ViBoolean idQuery,  
ViBoolean resetDevice, ViString optionString, ViSession *vi);
```

Purpose

Performs the following initialization actions:

- Creates a new IVI instrument session and optionally sets the initial state of the following session attributes: RangeCheck, QueryInstrStatus, Cache, Simulate, and RecordCoersions.
- Opens a session to the specified device using the interface and address that you specify for **resourceName**.
- If **IDQuery** is set to VI_TRUE, this function queries the device ID and checks that it is valid for NI-FGEN.
- If **resetDevice** is set to VI_TRUE, this function 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 function calls.

Parameters

Input

Name	Type	Description
resourceName	ViRsrc	Specifies the resource name of the device to i

Example #	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1
2	NI-DAQmx device	<i>myDAQmxName</i>
3	NI-DAQmx device	DAQ:: <i>myDeviceName</i>
4	NI-DAQmx device	DAQ::2
5	IVI logical name or IVI virtual name	<i>myLogicalName</i> or <i>myVirtualName</i>

For Traditional NI-DAQ devices, the syntax is just the number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name as shown in Example 2. Typical default names are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device on the name in MAX and entering a new name.

An alternate syntax for NI-DAQmx devices concatenates the device name, as shown in Example 3. This name is used to identify the use of an NI-DAQmx device in an application to a Traditional NI-DAQ device. For example, if the device name is Dev1, you can rename the NI-DAQmx device to 1 in the resource name, as shown in Example 4.

If you use the DAQ::*n* syntax and an NI-DAQmx device has a name with that same name, the NI-DAQmx device is selected.

You can also pass in the name of an IVI logical name or a virtual name configured with the IVI Configuration utility, as shown in Example 5. The IVI logical name identifies a particular virtual instrument. The IVI virtual name specifies the specific device and specifies the initial settings.



Caution Traditional NI-DAQ and NI-DAQ are case-sensitive. However, all IVI names, including virtual names in your program, are case-sensitive. If you use logical names or virtual names in your program, you must ensure that the name you use matches the name in the IVI Configuration file. Note that the case of the characters in the logical name or virtual name must match the case of the characters in the name in the IVI Configuration file.

idQuery	ViBoolean	Specify whether you want NI-FGEN to perform an ID query.
		When you set this parameter to VI_TRUE, NI-FGEN performs an ID query. NI-FGEN performs an ID query if the device you initialize is a type that it supports. Circumstances can arise where sending an ID query is undesirable. When you set this parameter to VI_FALSE, NI-FGEN does not send an ID query to the device without performing an ID query.
		Defined Values

VI_TRUE	Perform ID query
VI_FALSE	Skip ID query

Default Value: VI_TRUE

resetDevice	ViBoolean	Specifies whether you want to reset the device after the configuration procedure.
		Defined Values

VI_TRUE	Reset device
VI_FALSE	Do not reset device

Default Value: VI_TRUE

optionString	ViString	Sets the initial value of certain session attributes. The syntax for optionString is <i><attributeName> = <value></i> where <i>attributeName</i> is the name of the attribute and <i>value</i> is the value the attribute is set to. To set multiple attributes, separate them with a semicolon. If you pass NULL or an empty string for this parameter, the attribute values are not set.
--------------	----------	--

default values for these attributes. You can override these default values by assigning a value explicitly in a string that you pass to the `optionString` argument.

You do not have to specify all of the attributes listed above. However, if you do not specify one of the attributes, it uses its default value.

If simulation is enabled (`Simulate=1`), you may want to simulate. To specify a device, enter the device name in the `Device` column of the **optionString**.

`DriverSetup=Model:<driver model number>;BoardType:<board type>;BoardSize:<size of onboard memory in bytes>`

Syntax Examples

Device	optionString
NI PXI-5404	<code>DriverSetup=Model:5404;BoardType:PXI;BoardSize:16384</code>
NI PCI-5411	<code>DriverSetup=Model:5411;BoardType:PCI;BoardSize:16384</code>

Attributes and Defined Values

Attribute Name	Attribut
RangeCheck	NIFGEN_ATTR_RANGE_CHECK
QueryInstrStatus	NIFGEN_ATTR_QUERY_INSTRUMENT_STATUS
Cache	NIFGEN_ATTR_CACHE
Simulate	NIFGEN_ATTR_SIMULATE

Default Values: "Simulate=0,RangeCheck=1,QueryInstrStatus=0,Cache=0"

Output

Name	Type	Description
vi	ViSession	Returns a session handle that you can use to make subsequent NI-FGEN function calls.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_Commit

ViStatus niFgen_Commit (ViSession vi);

Purpose

Causes a transition to the Committed state. This function verifies attribute values, reserves the device, and commits the attribute values to the device. If the attributes values are all valid, the device configuration matches the session configuration.

In the Committed state, you can load waveforms, scripts, and sequences into memory. If any attributes are changed, NI-FGEN implicitly transitions back to the Idle state, where you can program all session properties before applying them to the device. This function has no effect if the device is already in the Committed state, and returns a successful status value. This function returns an error if it is called in the Generating state.

Using this function has the following benefits:

- The [niFgen_InitiateGeneration](#) function occurs faster because the device is configured.
- Routes are committed, so signals are exported or imported.
- The PLL and all DCMs are locked.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_reset

```
ViStatus niFgen_Reset (ViSession vi);
```

Purpose

Resets the instrument to a known state. This function aborts the generation, clears all routes, and resets session attributes to the default values. This function 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 function exhibits the same behavior as the [niFgen_ResetDevice](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_close

```
ViStatus niFgen_close (ViSession vi);
```

Purpose

Performs the following operations:

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



Note After calling `niFgen_close`, you cannot use NI-FGEN again until you call [niFgen_init](#) or [niFgen_InitWithOptions](#).

Not all signal routes established by calling the [niFgen_ExportSignal](#) and [niFgen_RouteSignalOut](#) functions are released when the NI-FGEN session is closed. The following table shows what will happen to a signal route on your device when the NI-FGEN session is closed.

Routes To	NI 5401/5411/5431	Other
Front Panel	Remain connected	Remain connected
RTSI/PXI Backplane	Remain connected	Disconnected

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Configuration Functions

Configure Operation Mode	niFgen_ConfigureOperationMode
Configure Output Mode	niFgen_ConfigureOutputMode
Configure Output Enabled	niFgen_ConfigureOutputEnabled
Configure Output Impedance	niFgen_ConfigureOutputImpedance
Enable Analog Filter	niFgen_EnableAnalogFilter
Disable Analog Filter	niFgen_DisableAnalogFilter
Enable Digital Filter	niFgen_EnableDigitalFilter
Disable Digital Filter	niFgen_DisableDigitalFilter
Enable Digital Patterning	niFgen_EnableDigitalPatterning
Disable Digital Patterning	niFgen_DisableDigitalPatterning

niFgen_ConfigureOperationMode

```
ViStatus niFgen_ConfigureOperationMode (ViSession vi,  
ViConstString channelName, ViInt32 operationMode);
```

Purpose

Determines how the signal generator produces waveforms. NI signal generators support only Continuous Operation mode. To control trigger mode, use the [niFgen_ConfigureTriggerMode](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the operation mode.
operationMode	ViInt32	Specifies the operation mode you want the signal generator to use. NI-FGEN sets the NIFGEN_ATTR_OPERATION_MODE attribute to this value. Only one value is supported. Defined Value: NIFGEN_VAL_OPERATE_CONTINUOUS

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureOutputMode

ViStatus niFgen_ConfigureOutputMode (ViSession vi, ViInt32 outputMode);

Purpose

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

For example, you can choose to generate 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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions function and identifies a particular instrument session.
outputMode	ViInt32	Specifies the output mode that you want the signal generator to use. The value you specify determines which functions and attributes you can use to control the waveform the signal generator produces.

Refer to the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute for more information on setting this parameter.

Defined Values

NIFGEN_VAL_OUTPUT_FUNC	Standard Function mode — Generates standard function waveforms as sine, square, triangle, and so on.
NIFGEN_VAL_OUTPUT_FREQ_LIST	Frequency mode — Generates standard function using a list of frequencies you define.
NIFGEN_VAL_OUTPUT_ARB	Arbitrary waveform mode —

	Generates waveforms user-created/pro waveform & of numeric
NIFGEN_VAL_OUTPUT_SEQ	Arbitrary sequence mode — Generates downloaded waveforms in order you specify.
NIFGEN_VAL_OUTPUT_SCRIPT	Script mode — Allows you to use scripting to link and loop multiple waveforms in complex combinations.

Default Value: NIFGEN_VAL_OUTPUT_FUNC

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureOutputEnabled

```
ViStatus niFgen_ConfigureOutputEnabled (ViSession vi,  
ViConstString channelName, ViBoolean enabled);
```

Purpose

Enables or disables the signal generator output.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to enable the output. Default Value: "0"
enabled	ViBoolean	Specifies whether you want to enable or disable the output. NI-FGEN uses this value to set the <u>NIFGEN_ATTR_OUTPUT_ENABLED</u> attribute.

Defined Values

VI_TRUE	Enable the output
VI_FALSE	Disable the output

Default Value: VI_TRUE

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureOutputImpedance

```
ViStatus niFgen_ConfigureOutputImpedance (ViSession vi,  
ViConstString channelName, ViReal64 impedance);
```

Purpose

Configures the output impedance for the channel you specify.

Parameters

Input

Name	Type	Description				
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.				
channelName	ViConstString	Specifies the channel name for which you want to set the output impedance. Default Value: "0"				
impedance	ViReal64	Specifies the impedance value that you want the signal generator to use. NI-FGEN sets the NIFGEN_ATTR_OUTPUT_IMPEDANCE attribute to this value. Units: Ω (ohms) Defined Values: <table border="1"><tr><td>NIFGEN_VAL_50_OHMS</td><td>50 Ω of impedance</td></tr><tr><td>NIFGEN_VAL_75_OHMS</td><td>75 Ω of impedance</td></tr></table> Default Value: NIFGEN_VAL_50_OHMS	NIFGEN_VAL_50_OHMS	50 Ω of impedance	NIFGEN_VAL_75_OHMS	75 Ω of impedance
NIFGEN_VAL_50_OHMS	50 Ω of impedance					
NIFGEN_VAL_75_OHMS	75 Ω of impedance					

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_EnableAnalogFilter

```
ViStatus niFgen_EnableAnalogFilter (ViSession vi,  
ViConstString channelName, ViReal64 filterCorrectionFrequency);
```

Purpose

Enables the analog filter. You can also specify the filter correction frequency of the analog filter. This sets the [**NIFGEN_ATTR_ANALOG_FILTER_ENABLED**](#) attribute to VI_TRUE. This function is valid in Arbitrary Waveform, Arbitrary Sequence, and Script output modes. This function can also be used in Standard Function and Frequency List output modes for user-defined waveforms.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to enable the analog filter. Default Value: "0"
filterCorrectionFrequency	ViReal64	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 attenuation at that frequency. To disable amplitude correction, set filterCorrectionFrequency to 0. For Standard function output mode, the filter correction frequency typically should be set to the same value as the frequency of the standard waveform. Units: hertz Default Value: 0

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DisableAnalogFilter

```
ViStatus niFgen_DisableAnalogFilter (ViSession vi,  
ViConstString channelName);
```

Purpose

Disables the analog filter. This function sets the [**NIFGEN_ATTR_ANALOG_FILTER_ENABLED**](#) attribute to VI_FALSE. This function is valid in Arbitrary Waveform, Arbitrary Sequence, and Script modes. This function can also be used in Standard Function and Frequency List modes for user-defined waveforms.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to disable the analog filter. Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_EnableDigitalFilter

```
ViStatus niFgen_EnableDigitalFilter (ViSession vi,  
ViConstString channelName);
```

Purpose

Enables the digital filter by setting the [**NIFGEN_ATTR_DIGITAL_FILTER_ENABLED**](#) attribute to VI_TRUE. This function is valid in Arbitrary Waveform, Arbitrary Sequence, and Script output modes. This function can also be used in Standard Function and Frequency List output modes for user-defined waveforms.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to enable the digital filter. Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DisableDigitalFilter

```
ViStatus niFgen_DisableDigitalFilter (ViSession vi,  
ViConstString channelName);
```

Purpose

Disables the digital filter. This function sets the [**NIFGEN_ATTR_DIGITAL_FILTER_ENABLED**](#) attribute to VI_FALSE. This function is valid in Arbitrary Waveform, Arbitrary Sequence, and Script modes. This function can also be used in Standard Function and Frequency List modes for user-defined waveforms.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to disable the digital filter. Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_EnableDigitalPatterning

```
ViStatus niFgen_EnableDigitalPatterning (ViSession vi,  
ViConstString channelName);
```

Purpose

Enables digital patterning on the signal generator. This function sets the [NIFGEN_ATTR_DIGITAL_PATTERN_ENABLED](#) attribute to VI_TRUE.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to enable the digital patterning. Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DisableDigitalPatterning

```
ViStatus niFgen_DisableDigitalPatterning (ViSession vi,  
ViConstString channelName);
```

Purpose

Disables digital patterning on the signal generator. This function sets the [NIFGEN_ATTR_DIGITAL_PATTERN_ENABLED](#) attribute to VI_FALSE.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to disable digital patterning. Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Standard Function Output Functions

Configure Standard Waveform	niFgen_ConfigureStandardWaveform
Define User Standard Waveform	niFgen_DefineUserStandardWaveform
Clear User Standard Waveform	niFgen_ClearUserStandardWaveform
Configure Frequency	niFgen_ConfigureFrequency
Configure Amplitude	niFgen_ConfigureAmplitude

niFgen_ConfigureStandardWaveform

```
ViStatus niFgen_ConfigureStandardWaveform (ViSession vi,  
ViConstString channelName, ViInt32 waveform, ViReal64 amplitude,  
ViReal64 dcOffset, ViReal64 frequency, ViReal64 startPhase);
```

Purpose

Configures the attributes of the signal generator that affect standard waveform generation (the [NIFGEN_ATTR_FUNC_WAVEFORM](#), [NIFGEN_ATTR_FUNC_AMPLITUDE](#), [NIFGEN_ATTR_FUNC_DC_OFFSET](#), [NIFGEN_ATTR_FUNC_FREQUENCY](#), and [NIFGEN_ATTR_FUNC_START_PHASE](#) attributes).



Note You must use the [niFgen_ConfigureOutputMode](#) function with the **outputMode** parameter set to NIFGEN_VAL_OUTPUT_FUNC before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitWithOptions functions and identifies the session.
channelName	ViConstString	Specifies the channel name for which you want to generate the waveform.
waveform	ViInt32	Specifies the standard waveform that you want to produce. NI-FGEN sets the NIFGEN_ATTR_WFM to this value.

Defined Values

NIFGEN_VAL_WFM_SINE	Sinusoidal
NIFGEN_VAL_WFM_SQUARE	Square
NIFGEN_VAL_WFM_TRIANGLE	Triangle
NIFGEN_VAL_WFM_RAMP_UP	Positive ramp
NIFGEN_VAL_WFM_RAMP_DOWN	Negative ramp
NIFGEN_VAL_WFM_DC	Constant DC
NIFGEN_VAL_WFM_NOISE	White noise
NIFGEN_VAL_WFM_USER	User-defined waveform with the niFgen_Waveform function

Default Value: NIFGEN_VAL_WFM_SINE

amplitude	ViReal64	Specifies the amplitude of the standard waveform that you want to produce. This value is the voltage at the output terminal. NI-FGEN sets the NIFGEN_ATTR_AMPLITUDE to this value. For example, to produce a waveform ranging from -10.00 V to 10.00 V, set the amplitude to 10.00 V.
-----------	----------	--

		 Note This parameter does not affect when you set the waveform parameter niFgen_ConfigureStandardWaveform for NIFGEN_VAL_WFM_DC.
		Units: peak-to-peak voltage Default Value: None
dcOffset	ViReal64	<p>Specifies the DC offset of the standard waveform generator to produce. This value is the offset from ground to the center of the <u>NIFGEN_ATTR_FUNC_DC_OFFSET</u> attribute.</p> <p>For example, to configure a waveform with a range from 0.00 to +10.00 V, set the DC Offset to 5.00 V.</p> <p>Units: volts Default Value: None</p>
frequency	ViReal64	<p>Specifies the frequency of the standard waveform generator to produce. NI-FGEN sets the <u>NIFGEN_ATTR_FUNC_FREQUENCY</u> attribute.</p> <p> Notes This parameter does not affect when you set the waveform parameter niFgen_ConfigureStandardWaveform for NIFGEN_VAL_WFM_DC.</p> <p>Frequency ranges vary from device to device. See the <u>Features supported</u> topic for your device.</p> <p>Units: hertz Default Value: None</p>
startPhase	ViReal64	<p>Specifies the horizontal offset of the standard waveform generator to produce. Specify this attribute per waveform cycle. NI-FGEN sets the <u>NIFGEN_ATTR_FUNC_START_PHASE</u> attribute. A start phase of 180 degrees means output generates the waveform 180 degrees out of phase. A start phase of 360 degrees means output generates the waveform 360 degrees out of phase.</p>

entire waveform cycle, which is identical to



Note This parameter does not affect when you set the **waveform** parameter to NIFGEN_VAL_WFM_DC.

Units: degrees of one cycle

Default Value: 0.00

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DefineUserStandardWaveform

```
ViStatus niFgen_DefineUserStandardWaveform (ViSession vi,  
ViConstString channelName, ViInt32 waveformSize,  
ViReal64[] waveformdataArray);
```

Purpose

Defines a user waveform for use in either Standard Function output mode or Frequency List output mode.

To select the waveform, set the **waveform** parameter to NIFGEN_VAL_WFM_USER with either the [niFgen_ConfigureStandardWaveform](#) or the [niFgen_CreateFreqList](#) function.

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 **outputMode** to NIFGEN_VAL_OUTPUT_FUNC or NIFGEN_VAL_OUTPUT_FREQ_LIST using the [niFgen_ConfigureOutputMode](#) function before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to define a user standard waveform. Default Value: "0"
waveformSize	ViInt32	Specifies the size of the arbitrary waveform you want to create. Default Value: 16384
waveformDataArray ViReal64[]		Specifies the array of data you want to use for the new arbitrary waveform. The array must have at least as many elements as the value that you specify in waveformSize . You must normalize the data points in the array to be between -1.00 and +1.00. Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearUserStandardWaveform

```
ViStatus niFgen_ClearUserStandardWaveform (ViSession vi,  
ViConstString channelName);
```

Purpose

Clears the user-defined waveform created by the [niFgen_DefineUserStandardWaveform](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name from which you want to clear a user standard waveform.

Default Value: "0"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureFrequency

```
ViStatus niFgen_ConfigureFrequency (ViSession vi,  
ViConstString channelName, ViReal64 frequency);
```

Purpose

Configures the frequency of the standard waveform that you want the signal generator to produce.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure a standard waveform. Default Value: "0"
frequency	ViReal64	Specifies the frequency of the standard waveform that you want the signal generator to produce. NI-FGEN sets the NIFGEN_ATTR_FUNC_FREQUENCY attribute to this value.



Notes This parameter does not affect signal generator behavior when you set the **waveform** parameter of the `niFgen_ConfigureStandardWaveform` function to `NIFGEN_VAL_WFM_DC`.

Frequency ranges vary from device to device. For information about the frequencies supported on your device, refer to the [Features supported](#) topic for your device.

Units: hertz

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureAmplitude

```
ViStatus niFgen_ConfigureAmplitude (ViSession vi,  
ViConstString channelName, ViReal64 amplitude);
```

Purpose

Configures the amplitude of the standard waveform that you want the signal generator to produce.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure a standard waveform. Default Value: "0"
amplitude	ViReal64	Specifies the amplitude of the standard waveform that you want the signal generator to produce. This value is the amplitude at the output terminal. NI-FGEN sets the NIFGEN_ATTR_FUNC_AMPLITUDE attribute to this value. For example, to produce a waveform ranging from -5.00 to +5.00 V, set the amplitude to 10.00 V.



Note This parameter does not affect signal generator behavior when you set the **waveform** parameter of the `niFgen_ConfigureStandardWaveform` function to `NIFGEN_VAL_WFM_DC`.

Units: peak-to-peak voltage

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Frequency List Output Functions

Create Frequency List [niFgen_CreateFreqList](#)

Configure Frequency List [niFgen_ConfigureFreqList](#)

Clear Frequency List [niFgen_ClearFreqList](#)

Query Freq List Capabilities [niFgen_QueryFreqListCapabilities](#)

niFgen_CreateFreqList

```
ViStatus niFgen_CreateFreqList (ViSession vi, ViInt32 waveform,  
ViInt32 frequencyListLength, ViReal64[] frequencyArray,  
ViReal64[] durationArray, ViInt32* frequencyListHandle);
```

Purpose

Creates a frequency list from an array of frequencies (**frequencyArray**) and an array of durations (**durationArray**). The arrays must be the same size, and this must also be the size of the the **frequencyListLength**. The signal generator generates each frequency for the given amount of time and then proceeds to the next frequency. When the end of the list is reached, the signal generator starts over at the beginning of the list. The function returns a handle that identifies the frequency list (the **frequencyListHandle**). You can pass this handle to [**niFgen_ConfigureFreqList**](#) to specify what frequency list you want the signal generator to produce.



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

You must call the [**niFgen_ConfigureOutputMode**](#) function to set the output mode to NIFGEN_VAL_OUTPUT_FREQ_LIST before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is one of the niFgen_InitWithOptions functions and session.
waveform	ViInt32	Specifies the standard waveform that you produce. NI-FGEN sets the NIFGEN_AT to this value.

Defined Values

NIFGEN_VAL_WFM_SINE	Sine
NIFGEN_VAL_WFM_SQUARE	Square
NIFGEN_VAL_WFM_TRIANGLE	Triangle
NIFGEN_VAL_WFM_RAMP_UP	Pulse
NIFGEN_VAL_WFM_RAMP_DOWN	Necklace
NIFGEN_VAL_WFM_DC	Constant
NIFGEN_VAL_WFM_NOISE	White noise
NIFGEN_VAL_WFM_USER	User-defined

Default Value: NIFGEN_VAL_WFM_SINE

frequencyListLength	ViInt32	Specifies the number of steps in the frequency list. The value must be between the minimum and maximum frequency list lengths from minimumFrequencyListLength and maximumFrequencyListLength parameters in the niFgen_QueryFreqListC . frequency and duration must each be a multiple of the list length.
---------------------	---------	---

Default Value: None

frequencyArray	ViReal64[]	Specifies the array of frequencies to form the frequency list. This array must have at least as many elements as frequencyListLength . Each frequency corresponds to a corresponding durationArray element that specifies the duration for which the frequency is repeated.
durationArray	ViReal64[]	Specifies the array of durations to form the frequency list. This array must have at least as many elements as the frequencyListLength . Each duration corresponds to a corresponding frequencyArray element that specifies the duration in seconds to generate the corresponding frequency.

Output

Name	Type	Description
frequencyListHandle	VlInt32*	Returns the handle that identifies the new frequency list. Set this handle to niFgen_ConfigureFreqList to change the frequency list sequence.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureFreqList

```
ViStatus niFgen_ConfigureFreqList (ViSession vi, ViConstString channelName,  
ViInt32 frequencyListHandle, ViReal64 amplitude, ViReal64 dcOffset,  
ViReal64 startPhase);
```

Purpose

Configures the attributes of the signal generator that affect frequency list generation (the [NIFGEN_ATTR_FREQ_LIST_HANDLE](#), [NIFGEN_ATTR_FUNC_AMPLITUDE](#), [NIFGEN_ATTR_FUNC_DC_OFFSET](#), and [NIFGEN_ATTR_FUNC_START_PHASE](#) attributes).



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

You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_FREQ_LIST before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the frequency list. Default Value: "0"
frequencyListHandle	ViInt32	Specifies the handle of the frequency list that you want the signal generator to produce. NI-FGEN sets the NIFGEN_ATTR_FREQ_LIST_HANDLE attribute to this value. You can create frequency list using the niFgen_CreateFreqList function. The niFgen_CreateFreqList function returns handle that you use to identify the list. Default Value: None
amplitude	ViReal64	Specifies the amplitude of the standard waveform that you want the signal generator to produce. This value is the amplitude at the output terminal. NI-FGEN sets the NIFGEN_ATTR_FUNC_AMPLITUDE attribute to this value. For example, to produce a waveform ranging from -5.00 to +5.00 V, set the amplitude to 10.00 V.



Note This parameter does not affect signal generator behavior

when you set the **waveform** parameter of the niFgen_ConfigureStandardWavefor function to NIFGEN_VAL_WFM_DC.

Units: peak-to-peak voltage

Default Value: None

dcOffset

ViReal64

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 offset is from ground to the center of the waveform). NI-FGEN sets the [NIFGEN_ATTR_FUNC_DC_OFFSET](#) attribute to this value.

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

Units: volts

Default Value: None

startPhase

ViReal64

Specifies the horizontal offset of the standard waveform you want the signal generator to produce. Specify this attribute in degrees of one waveform cycle. NI-FGEN sets the [NIFGEN_ATTR_FUNC_START_PHASE](#) attribute to this value. 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.



Note This parameter does no

affect signal generator behavio
when you set the **waveform**
parameter to
NIFGEN_VAL_WFM_DC.

Units: degrees of one cycle

Default Value: None degrees

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearFreqList

ViStatus niFgen_ClearFreqList (ViSession vi, ViInt32 frequencyListHandle);

Purpose

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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
frequencyListHandle	ViInt32	Specifies the handle of the frequency list that you want the signal generator to produce. NI-FGEN sets the NI_FGEN_ATTR_FREQ_LIST_HANDLE attribute to this value. You can create a frequency list using the niFgen_CreateFreqList function. The niFgen_CreateFreqList function returns a handle that you use to identify the list.

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_QueryFreqListCapabilities

```
ViStatus niFgen_QueryFreqListCapabilities (ViSession vi,  
ViInt32* maximumNumberOfFreqLists,  
ViInt32* minimumFrequencyListLength,  
ViInt32* maximumFrequencyListLength,  
ViReal64* minimumFrequencyListDuration,  
ViReal64* maximumFrequencyListDuration,  
ViReal64* frequencyListDurationQuantum);
```

Purpose

Returns the attributes of the signal generator that are related to creating frequency lists. These attributes are

NIFGEN_ATTR_MAX_NUM_FREQ_LISTS,
NIFGEN_ATTR_MIN_FREQ_LIST_LENGTH,
NIFGEN_ATTR_MAX_FREQ_LIST_LENGTH,
NIFGEN_ATTR_MIN_FREQ_LIST_DURATION,
NIFGEN_ATTR_MAX_FREQ_LIST_DURATION, and
NIFGEN_ATTR_FREQ_LIST_DURATION_QUANTUM.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session from the niFgen_init or the niOpen functions and identifies a parallel session.

Output

Name	Type	Description
maximumNumberOfFreqLists	ViInt32	Returns the maximum number of frequency lists the signal generator allows. It obtains this value from the NIFGEN_ATTR_MAX_NUM_FREQUENCIES attribute.
minimumFrequencyListLength	ViInt32	Returns the minimum number of frequency lists the signal generator allows in a frequency list. It obtains this value from the NIFGEN_ATTR_MIN_FREQ_LIST_LENGTH attribute.
maximumFrequencyListLength	ViInt32	Returns the maximum number of frequency lists the signal generator allows in a frequency list. It obtains this value from the NIFGEN_ATTR_MAX_FREQ_LIST_LENGTH attribute.
minimumFrequencyListDuration	ViReal64	Returns the minimum duration of a frequency list step the signal generator allows in a step of a frequency list. It obtains this value from the NIFGEN_ATTR_MIN_FREQ_LIST_DURATION attribute.
maximumFrequencyListDuration	ViReal64	Returns the maximum duration of a frequency list step the signal generator allows in a step of a frequency list. It obtains this value from the NIFGEN_ATTR_MAX_FREQ_LIST_DURATION attribute.
frequencyListDurationQuantum	ViReal64	Returns the quantum of which a frequency list is a multiple.

value from the
NIFGEN_ATTR_FREQ_LIST
attribute.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Arbitrary Waveform Output Functions

Configure Gain	niFgen_ConfigureGain
Configure Sample Rate	niFgen_ConfigureSampleRate
Query Arb Waveform Capabilities	niFgen_QueryArbWfmCapabilities
Create Waveform F64	niFgen_CreateWaveformF64
Create Waveform I16	niFgen_CreateWaveformI16
Create Waveform Complex F64	niFgen_CreateWaveformComplexF64
Create Waveform from File I16	niFgen_CreateWaveformFromFileI16
Create Waveform from File F64	niFgen_CreateWaveformFromFileF64
Create Waveform from File HWS	niFgen_CreateWaveformFromFileHWS
Configure Arbitrary Waveform	niFgen_ConfigureArbWaveform
Clear Arbitrary Waveform	niFgen_ClearArbWaveform

niFgen_ConfigureGain

```
ViStatus niFgen_ConfigureGain (ViSession vi, ViConstString channelName,  
ViReal64 gain);
```

Purpose

Configures the amount of gain to apply to the waveform.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the gain. Default Value: "0"
gain	ViReal64	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 a range of -1.00 to +1.00 V. You can use this parameter to scale the arbitrary waveform to other ranges. NI-FGEN sets the <u>NIFGEN_ATTR_ARB_GAIN</u> attribute to this value. For example, to configure the output signal to range from -2.00 to +2.00 V, set the gain to 2.00. Units: volts Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureSampleRate

ViStatus niFgen_ConfigureSampleRate (ViSession vi, ViReal64 sampleRate);

Purpose

Configures the sample rate attribute, which determines the rate at which the signal generator produces arbitrary waveforms. When you configure the signal generator to produce an arbitrary sequence, this value is the sample rate for all arbitrary waveforms in the sequence.



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

You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_ARB, NIFGEN_VAL_OUTPUT_SEQ, or NIFGEN_VAL_OUTPUT_SCRIPT before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
sampleRate	ViReal64	Specifies the sample rate at which you want the signal generator to generate arbitrary waveforms. NI-FGEN sets the <u>NIFGEN_ATTR_ARB_SAMPLE_RATE</u> attribute to this value. Units: samples/s Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_QueryArbWfmCapabilities

```
ViStatus niFgen_QueryArbWfmCapabilities (ViSession vi,  
ViInt32* maximumNumberofWaveforms, ViInt32* waveformQuantum,  
ViInt32* minimumWaveformSize, ViInt32* maximumWaveformSize);
```

Purpose

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.



Note If you do not want to obtain the waveform quantum, pass a value of VI_NULL for this parameter.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session obtained from the niFgen_init or niFgen_InitWithOptions function. This parameter identifies a particular instrument session.

Output

Name	Type	Description
maximumNumberofWaveforms	Vilnt32*	Returns the maximum number of waveforms that the signal generator allows. NI-FGEN obtains this value from the NIFGEN_ATTR_MAX_NUM_WAVEFORMS attribute.
waveformQuantum	Vilnt32*	The size (number of points) of waveform must be a multiple of this quantum value. This parameter specifies the quantum value that the signal generator uses. NI-FGEN returns this value from the NIFGEN_ATTR_WAVEFORM_QUANTUM attribute. For example, when this attribute has a value of 8, all waveform sizes are multiples of 8.
minimumWaveformSize	Vilnt32*	Returns the minimum number of points that the signal generator allows for a waveform. NI-FGEN obtains this value from the NIFGEN_ATTR_MIN_WAVEFORM_SIZE attribute.
maximumWaveformSize	Vilnt32*	Returns the maximum number of points that the signal generator allows for a waveform. NI-FGEN obtains this value from the NIFGEN_ATTR_MAX_WAVEFORM_SIZE attribute.

from the
NIFGEN_ATTR_MAX_WAVEF
attribute.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformF64

```
ViStatus niFgen_CreateWaveformF64 (ViSession vi,  
ViConstString channelName, ViInt32 waveformSize,  
ViReal64[] waveformdataArray, ViInt32* waveformHandle);
```

Purpose

Creates an onboard waveform from binary F64 (floating point double) data for use in Arbitrary Waveform output mode or Arbitrary Sequence output mode. The **waveformHandle** returned can later be used 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.



Note You must call the [niFgen_ConfigureOutputMode](#) function to set the **outputMode** parameter to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Default Value: "0"
waveformSize	ViInt32	Specifies the size of the arbitrary waveform that you want created. The size must meet the following restrictions: <ul style="list-style-type: none">• The size must be less than or equal to the maximum waveform size that the device allows.• The size must be greater than or equal to the minimum waveform size that the device allows.• The size must be an integer multiple of the device waveform quantum. You can obtain these values from the maximumWaveformSize , minimumWaveformSize , and waveformQuantum parameters in the niFgen_QueryArbWfmCapabilities function.

Default Value: None

waveformdataArray ViReal64[]

Specifies the array of data you want to use for the new arbitrary waveform. The array must have at least as many elements as the value that you specify in **waveformSize**.

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

Default Value: None

Output

Name	Type	Description
waveformHandle	VlInt32*	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformI16

```
ViStatus niFgen_CreateWaveformI16 (ViSession vi,  
ViConstString channelName, ViInt32 waveformSize,  
ViInt16[] waveformdataArray, ViInt32* waveformHandle);
```

Purpose

Creates an onboard waveform from binary I16 data for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **waveformHandle** returned can later be used 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.



Note You must call the [niFgen_ConfigureOutputMode](#) function to set the **outputMode** parameter to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Default Value: "0"
waveformSize	ViInt32	Specifies the size of the arbitrary waveform that you want created. The size must meet the following restrictions: <ul style="list-style-type: none">• The size must be less than or equal to the maximum waveform size that the device allows.• The size must be greater than or equal to the minimum waveform size that the device allows.• The size must be an integer multiple of the device waveform quantum. You can obtain these values from the maximumWaveformSize , minimumWaveformSize , and waveformQuantum parameters in the niFgen_QueryArbWfmCapabilities function.

Default Value: None

waveformdataArray ViReal64[]

Specifies the array of data you want to use for the new arbitrary waveform. The array must have at least as many elements as the value that you specify in **waveformSize**.

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

Default Value: None

Output

Name	Type	Description
waveformHandle	VlInt32*	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformComplexF64

```
ViStatus niFgen_CreateWaveformComplexF64 (ViSession vi,  
ViConstString channelName, ViInt32 numberofSamples,  
niFgen_ComplexF64[] waveformdataArray, ViInt32* waveformHandle);
```

Purpose

Creates an onboard waveform from complex double-precision floating-point data for use with the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute set to Arbitrary Waveform or Arbitrary Sequence output mode on devices with the [NIFGEN_ATTR_OUTPUT_ENABLED](#) attribute set to VI_TRUE and the [NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE](#) attribute set to NIFGEN_VAL_OSP_COMPLEX. The **waveformHandle** returned by the function 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.



Note You must call the [niFgen_ConfigureOutputMode](#) function to set the **outputMode** parameter to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Default Value: "0"
numberofSamples	ViInt32	Specifies the size of the arbitrary waveform that you want created. The size must meet the following restrictions: <ul style="list-style-type: none">• The size must be less than or equal to the maximum waveform size that the device allows.• The size must be greater than or equal to the minimum waveform size that the device allows.• The size must be an integer multiple of the device waveform quantum. You can obtain these values from the maximumWaveformSize , minimumWaveformSize , and waveformQuantum parameters in <u>niFgen_QueryArbWfmCapabilities</u> .

Default Value: None

`waveformdataArray ViReal64[]` Specifies the array of data you want to use for the new arbitrary waveform. The array must have at least as many elements as the value that you specify in **waveformSize**. You must normalize the data points in the array to be between –1.00 and +1.00.

Default Value: None

Output

Name	Type	Description
<code>waveformHandle</code>	<code>ViInt32*</code>	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformFromFileI16

```
ViStatus niFgen_CreateWaveformFromFileI16 (ViSession vi,  
ViConstString channelName, ViConstString filename, ViInt32 byteOrder,  
ViInt32* waveformHandle);
```

Purpose

Takes the binary I16 data from the specified file and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **waveformHandle** returned can later be used 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 i16 data (values between -32768 and +32767) is assumed to represent -1 to +1 V. Use the

[**NIFGEN_ATTR_DIGITAL_GAIN**](#) attribute to generate different voltage outputs.

You must call the [**niFgen_ConfigureOutputMode**](#) function to set the **outputMode** parameter to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Default Value: "0"
fileName	ViConstString	The full path and name of the file where waveform data resides.
byteOrder	ViInt32	Specifies the byte order of the data in the file.  Note 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 between different processors.)

Defined Values

NIFGEN_VAL_LITTLE_ENDIAN	LittleEndian Data—least significant bit is stored at the lowest address followed by the other bits.
--------------------------	--

		bits, in of increas signific
NIFGEN_VAL_BIG_ENDIAN	Big Endian Data—most significant bit is stored at the lowest address followed by the other bits, in decreasing order of significance.	Big Endian Data—most significant bit is stored at the lowest address followed by the other bits, in decreasing order of significance.

Default Value:

NIFGEN_VAL_LITTLE_ENDIAN

Output

Name	Type	Description
waveformHandle	Vilnt32*	The handle that identifies the new waveform. This handle is used later when referring to the waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformFromFileF64

```
ViStatus niFgen_CreateWaveformFromFileF64 (ViSession vi,  
ViConstString channelName, ViConstString filename, ViInt32 byteOrder,  
ViInt32* waveformHandle);
```

Purpose

This function takes the F64 (floating point double) data from the specified file and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **waveformHandle** returned can later be used 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 F64 data must be between -1.0 and +1.0 V. Use the [NIFGEN_ATTR_DIGITAL_GAIN](#) attribute to generate different voltage outputs.

You must call the [niFgen_ConfigureOutputMode](#) function to set the **outputMode** parameter to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Defined Value: "0" Default Value: "0"
fileName	ViConstString	The full path and name of the file where waveform data resides.
byteOrder	ViInt32	Specifies the byte order of the data in the file.  Note 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 between different processors.)

Defined Values

NIFGEN_VAL_LITTLE_ENDIAN	Little Endian—least significant bit is stored at the lowest address
--------------------------	---

		follows the other bits, in order of increasing significance
NIFGEN_VAL_BIG_ENDIAN	Big Endian Data— most significant bit is stored at the lowest address followed by the other bits, in order of decreasing significance	

Default Value:

NIFGEN_VAL_LITTLE_ENDIAN

Output

Name	Type
waveformHandle	VlInt32*

Description

The handle that identifies the new waveform.
This handle is used later when referring to the waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateWaveformFromFileHWS

```
ViStatus niFgen_CreateWaveformFromFileHWS (ViSession vi,  
ViConstString channelName, ViConstString filename,  
ViBoolean useRateFromWaveform, ViBoolean useGain&OffsetFromWaveform,  
ViInt32* waveformHandle);
```

Purpose

Takes the waveform data from the specified HWS (Hierarchical Waveform Storage) file and creates an onboard waveform for use in Arbitrary Waveform or Arbitrary Sequence output mode. The **waveformHandle** returned by the function 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 rate, gain and offset with the data. If the `useRateFromWaveform` and `useGain&OffsetFromWaveform` inputs are set to `VI_TRUE`, this function also sets those properties on NI-FGEN.



Notes If you choose to have this function set the gain and offset properties for you, you should **not** use the [niFgen_ConfigureArbWaveform](#) or [niFgen_ConfigureArbSequence](#) functions, as they also set the gain and offset, thereby overriding the values set by this function. Instead, use the [NI-FGEN_ATTR_ARB_WAVEFORM_HANDLE](#) or [NI-FGEN_ATTR_ARB_SEQUENCE_HANDLE](#) attributes.

HWS files may contain I16 or DBL data (depending on how it was saved). This function automatically adapts to either data type. If the file contains DBL data, it must be between -1.0 and +1.0. Check the "Scale Data to +/- 1V" option in the Analog Waveform Editor to ensure your data is between -1.0 and +1.0.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to create the waveform. Defined Value: "0" Default Value: "0"
fileName	ViConstString	The full path and name of the file where the waveform data resides.
useRateFromWaveform	ViBoolean	If you set this parameter input to VI_TRUE and if onboard signal processing (OSP) is enabled, the 'rate' from the waveform is interpreted as the data rate, and FGEN sets the data rate attribute for you. In all other cases, it is interpreted as the sample rate, and FGEN sets the sample rate attribute for you.

Defined Values

VI_TRUE	Use rate from waveform.
VI_FALSE	Do not use rate from waveform.

Default Value: VI_TRUE

useGain&OffsetFromWaveform ViBoolean

If this input is set to VI_TRUE, NI-FGEN retrieves the gain and offset values from the specified HWS file and applies them to the NI-FGEN driver.

Defined Values

VI_TRUE	Use gain and offset from waveform.
VI_FALSE	Do not use gain and offset from waveform.

Default Value: VI_TRUE

Output

Name	Type	Description
waveformHandle	ViInt32*	The handle that identifies the new waveform. This handle is used later when referring to this

waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureArbWaveform

```
ViStatus niFgen_ConfigureArbWaveform (ViSession vi,  
ViConstString channelName, ViInt32 waveformHandle, ViReal64 gain,  
ViReal64 offset);
```

Purpose

Configures the attributes of the signal generator that affect arbitrary waveform generation (the [NIFGEN_ATTR_ARB_WAVEFORM_HANDLE](#), [NIFGEN_ATTR_ARB_GAIN](#), and [NIFGEN_ATTR_ARB_OFFSET](#) attributes).



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

You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_ARB before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session
channelName	ViConstString	Specifies the channel name for which you want to configure an arbitrary waveform. Default Value: "0"
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform you want the signal generator produce. NI-FGEN sets the NIFGEN_ATTR_ARB_WAVEFORM_HANDLE attribute to this value. You can create an arbitrary waveform using one of the following niFgen Create Waveform functions: <ul style="list-style-type: none">• niFgen_CreateWaveformF64• niFgen_CreateWaveformI16• niFgen_CreateWaveformFromFileI16• niFgen_CreateWaveformFromFileF64• niFgen_CreateWaveformFromFileHV These functions return a handle that you can use to identify the waveform. Default Value: None
gain	ViReal64	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 a range of -1.00 to +1.00 V. You can use this parameter to scale the arbitrary waveform to other ranges. NI-FGEN sets the NIFGEN_ATTR_ARB_GAIN attribute to the specified value.

value.

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

Units: volts

Default Value: None

offset

ViReal64

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 a range of –1.00 to +1.00 V. You can use this parameter to shift the range of the arbitrary waveform. FGEN sets the

[NIFGEN_ATTR_ARB_OFFSET](#) attribute to this value.

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 the offset to 1.00.

Units: volts

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearArbWaveform

ViStatus niFgen_ClearArbWaveform (ViSession vi, ViInt32 waveformHandle);

Purpose

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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
waveformHandle	Vilnt32	<p>Specifies the handle of the arbitrary waveform that you want the signal generator to remove.</p> <p>You can create multiple arbitrary waveforms using one of the following niFgen Create Waveform functions:</p> <ul style="list-style-type: none">• niFgen_CreateWaveformF64• niFgen_CreateWaveformI16• niFgen_CreateWaveformFromFileI16• niFgen_CreateWaveformFromFileF64• niFgen_CreateWaveformFromFileHWS

Defined Value:

NIFGEN_VAL_ALL_WAVEFORMS—
Remove all waveforms from the signal generator.

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Arbitrary Sequence Output Functions

Query Arb Sequence Capabilities	niFgen_QueryArbSeqCapabilities
Create Arbitrary Sequence	niFgen_CreateArbSequence
Create Advanced Arb Sequence	niFgen_CreateAdvancedArbSequence
Configure Arbitrary Sequence	niFgen_ConfigureArbSequence
Clear Arbitrary Sequence	niFgen_ClearArbSequence
Clear Arbitrary Memory	niFgen_ClearArbMemory

niFgen_QueryArbSeqCapabilities

```
ViStatus niFgen_QueryArbSeqCapabilities (ViSession vi,  
ViInt32* maximumNumberOfSequences, ViInt32* minimumSequenceLength,  
ViInt32* maximumSequenceLength, ViInt32* maximumLoopCount);
```

Purpose

Returns the attributes of the signal generator that are related to creating arbitrary sequences (the [NIFGEN_ATTR_MAX_NUM_SEQUENCES](#), [NIFGEN_ATTR_MIN_SEQUENCE_LENGTH](#), [NIFGEN_ATTR_MAX_SEQUENCE_LENGTH](#), and [NIFGEN_ATTR_MAX_LOOP_COUNT](#) attributes).

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session obtained from the niFgen_init or niFgen_InitWithOptions function. This parameter identifies a particular instrument session.

Output

Name	Type	Description
maximumNumberOfSequences	Vilnt32*	Returns the maximum number of waveform sequences that the generator allows. NI-FGEN obtains this value from the NIFGEN_ATTR_MAX_NUM_SEQUENCES attribute.
minimumSequenceLength	Vilnt32*	Returns the minimum number of waveforms the signal generator can repeat in a sequence. NI-FGEN obtains this value from the NIFGEN_ATTR_MIN_SEQUENCE_LENGTH attribute.
maximumSequenceLength	Vilnt32*	Returns the maximum number of waveforms the signal generator can repeat in a sequence. NI-FGEN obtains this value from the NIFGEN_ATTR_MAX_SEQUENCE_LENGTH attribute.
maximumLoopCount	Vilnt32*	Returns the maximum number of times the signal generator can repeat a waveform in a sequence. NI-FGEN obtains this value from the NIFGEN_ATTR_MAX_LOOP_COUNT attribute.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateArbSequence

```
ViStatus niFgen_CreateArbSequence (ViSession vi, ViInt32 sequenceLength,  
ViInt32[] waveformHandlesArray, ViInt32[] loopCountsArray,  
ViInt32* sequenceHandle);
```

Purpose

Creates an arbitrary sequence from an array of waveform handles and an array of corresponding loop counts. 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.

The function returns a handle that identifies the sequence. You pass this handle to the [niFgen_ConfigureArbSequence](#) function to specify what arbitrary sequence you want the signal generator to produce.



Note You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
sequenceLength	Vilnt32	Specifies the number of waveforms in new arbitrary sequence that you want to create. The value you pass must be between the minimum and maximum sequence lengths that the signal generator allows. You can obtain the minimum and maximum sequence length from minimumSequenceLength and maximumSequenceLength in the niFgen_QueryArbSeqCapabilities function.
waveformHandlesArray	Vilnt32[]	<p>Default Value: None</p> <p>Specifies the array of waveform handles from which you want to create a new arbitrary sequence. The array must have at least as many elements as the value that you specify in sequenceLength. Each waveformHandlesArray element has a corresponding loopCountsArray element that indicates how many times that waveform is repeated. You obtain waveform handles when you create arbitrary waveforms with the niFgen_AllocateWaveform function or one of the following niFgen CreateWaveform functions:</p> <ul style="list-style-type: none">• niFgen_CreateWaveformF64• niFgen_CreateWaveformI16• niFgen_CreateWaveformFromFile

- [niFgen_CreateWaveformFromFile](#)
- [niFgen_CreateWaveformFromFile](#)

Default Value: None

loopCountsArray	VlInt32[]	<p>Specifies the array of loop counts you want to use to create a new arbitrary sequence. The array must have at least as many elements as the value that you specify in the sequenceLength parameter. Each loopCountsArray element corresponds to a waveformHandlesArray element and indicates how many times to repeat the waveform. Each element of the loopCountsArray must be less than or equal to the maximum number of loop counts that the signal generator allows. You can obtain the maximum loop count from maximumLoopCount in the niFgen_QueryArbSeqCapabilities function.</p>
		<p>Default Value: None</p>

Output

Name	Type	Description
sequenceHandle	VlInt32*	Returns the handle that identifies the arbitrary sequence. You can pass this handle to niFgen_ConfigureArbSequence to generate the arbitrary sequence.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CreateAdvancedArbSequence

```
ViStatus niFgen_ConfigureArbSequence (ViSession vi,  
ViConstString channelName, ViInt32 sequenceHandle, ViReal64 gain,  
ViReal64 offset);
```

Purpose

Creates an arbitrary sequence from an array of waveform handles and an array of corresponding loop counts. 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.

The function returns a handle that identifies the sequence. You pass this handle to the [niFgen_ConfigureArbSequence](#) function to specify what arbitrary sequence you want the signal generator to produce.

The niFgen_CreateAdvancedArbSequence function extends on [niFgen_CreateArbSequence](#) by adding the ability to set the number of samples in each sequence step and to set marker locations.



Note You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
sequenceLength	Vilnt32	Specifies the number of waveforms in new arbitrary sequence that you want to create. The value you pass must be between the minimum and maximum sequence lengths that the signal generator allows. You can obtain the minimum and maximum sequence length from minimumSequenceLength and maximumSequenceLength in the niFgen_QueryArbSeqCapabilities function.
waveformHandlesArray	Vilnt32[]	<p>Default Value: None</p> <p>Specifies the array of waveform handles from which you want to create a new arbitrary sequence. The array must have at least as many elements as the value that you specify in sequenceLength. Each waveformHandlesArray element has a corresponding loopCountsArray element that indicates how many times that waveform is repeated. You obtain waveform handles when you create arbitrary waveforms with the niFgen_AllocateWaveform function or one of the following niFgen CreateWaveform functions:</p> <ul style="list-style-type: none">• niFgen_CreateWaveformF64• niFgen_CreateWaveformI16• niFgen_CreateWaveformFromFile

- [niFgen_CreateWaveformFromFile](#)
- [niFgen_CreateWaveformFromFile](#)

Default Value: None

loopCountsArray	Vilnt32[]	<p>Specifies the array of loop counts you want to use to create a new arbitrary sequence. The array must have at least as many elements as the value that you specify in the sequenceLength parameter. Each loopCountsArray element corresponds to a waveformHandlesArray element and indicates how many times to repeat the waveform. Each element of the loopCountsArray must be less than or equal to the maximum number of loop counts that the signal generator allows. You can obtain the maximum loop count from maximumLoopCount in the niFgen_QueryArbSeqCapabilities function.</p>
sampleCountsArray	Vilnt32[]	<p>Specifies the array of sample counts that you want to use to create a new arbitrary sequence. The array must have at least as many elements as the value you specify in the sequenceLength parameter. Each sampleCountsArray element corresponds to a waveformHandlesArray element and indicates the subset, in samples, of the given waveform to generate. Each element of the sampleCountsArray must be larger than the minimum waveform size, a multiple of the waveform quantization and no larger than the number of samples in the corresponding waveform. You can obtain these values by calling the niFgen_QueryArbSeqCapabilities function.</p>

Default Value: None

sampleCountsArray	Vilnt32[]	<p>Specifies the array of sample counts that you want to use to create a new arbitrary sequence. The array must have at least as many elements as the value you specify in the sequenceLength parameter. Each sampleCountsArray element corresponds to a waveformHandlesArray element and indicates the subset, in samples, of the given waveform to generate. Each element of the sampleCountsArray must be larger than the minimum waveform size, a multiple of the waveform quantization and no larger than the number of samples in the corresponding waveform. You can obtain these values by calling the niFgen_QueryArbSeqCapabilities function.</p>
-------------------	-----------	---

[niFgen_QueryArbWfmCapabilities](#) funct

Default Value: None

markerLocationArray	Vilnt32	<p>Specifies the array of marker locations where you want a marker to be generated in the sequence. The array must have at least as many elements as the value you specify in the sequenceLength parameter. Each markerLocationArray element corresponds to a waveformHandlesArray element and indicates where in the waveform a marker is to generate. The marker location must be less than the size of the waveform marker is in. The markers are coerced to the nearest marker quantum and the coerced values are returned in the coercedMarkersArray parameter.</p> <p>If you do not want a marker generated in a particular sequence stage, set the marker to NIFGEN_VAL_NO_MARKER.</p>
---------------------	---------	---

Defined Value:

NIFGEN_VAL_NO_MARKER

Default Value: None

Output

Name	Type	Description
coercedMarkersArray	Vilnt32[]	Returns an array of all given markers are coerced (rounded) to the nearest marker quantum. Not all devices coerce markers.
sequenceHandle	Vilnt32*	Returns the handle that identifies the arbitrary sequence. You can pass this handle to <u>niFgen_ConfigureArbSequence</u> to generate the arbitrary sequence.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureArbSequence

```
ViStatus niFgen_ConfigureArbSequence (ViSession vi, ViInt32 sequenceLength,  
ViInt32[] waveformHandlesArray, ViInt32[] loopCountsArray,  
ViInt32* sequenceHandle);
```

Purpose

Configures the signal generator attributes that affect arbitrary sequence generation. These attributes are the

[NIFGEN_ATTR_ARB_SEQUENCE_HANDLE](#),

[NIFGEN_ATTR_ARB_GAIN](#), and [NIFGEN_ATTR_ARB_OFFSET](#)



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

You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session
channelName	ViConstString	Specifies the channel name from which you want to configure an arbitrary sequence. Default Value: "0"
sequenceHandle	ViInt32	Specifies the handle of the arbitrary sequence that you want the signal generator to produce. NI-FGEN sets the NIFGEN_ATTR_ARB_SEQUENCE_HANDLE attribute to this value. You can create an arbitrary sequence using the niFgen_CreateArbSequence function. The niFgen_CreateArbSequence function returns a handle that you use to identify the sequence. Default Value: None
gain	ViReal64	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 a range of -1.00 to +1.00 V. You can use this parameter to scale the arbitrary waveform to other ranges. NI-FGEN sets the NIFGEN_ATTR_ARB_GAIN attribute to this value. For example, to configure the output signal to range from -2.00 to +2.00 V, set the gain to 2.00. Units: volts

Default Value: None

offset ViReal64 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 a range –1.00 to +1.00 V. You can use this parameter to shift the range of the arbitrary waveform. NI-FGEN sets the [NIFGEN_ATTR_ARB_OFFSET](#) attribute to this value.

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 the offset to 1.00.

Units: volts

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearArbSequence

ViStatus niFgen_ClearArbSequence (ViSession vi, ViInt32 sequenceHandle);

Purpose

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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
sequenceHandle	Vilnt32	Specifies the handle of the arbitrary sequence that you want the signal generator to remove. You can create an arbitrary sequence using the niFgen_CreateArbSequence function. The niFgen_CreateArbSequence function returns a handle that you use to identify the sequence.

Defined Value:

NIFGEN_VAL_ALL_SEQUENCES—
Remove all sequences from the signal generator

Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearArbMemory

```
ViStatus niFgen_ClearArbMemory (ViSession vi);
```

Purpose

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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Incremental Waveform Write Functions

Allocate Waveform	<u>niFgen_AllocateWaveform</u>
Set Waveform Next Write Position	<u>niFgen_SetWaveformNextWritePosition</u>
Write Waveform	<u>niFgen_WriteWaveform</u>
Write Complex F64 Waveform	<u>niFgen_WriteWaveformComplexF64</u>
Write Binary 16 Waveform	<u>niFgen_WriteBinary16Waveform</u>
Write Complex Binary 16 Waveform	<u>niFgen_WriteComplexBinary16Waveform</u>

niFgen_AllocateWaveform

```
ViStatus niFgen_AllocateWaveform (ViSession vi, ViConstString channelName,  
ViInt32 waveformSize, ViInt32* waveformHandle);
```

Purpose

Specifies the size of a waveform so that it can be allocated in onboard memory before loading the associated data. Data can then be loaded in smaller chunks with the Write (Binary16)Waveform functions.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to allocate the waveform. Default Value: "0"
waveformSize	ViInt32	Specifies, in samples, the size of the waveform to allocate.

Output

Name	Type	Description
waveformHandle	ViInt32*	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetWaveformNextWritePosition

```
ViStatus niFgen_SetWaveformNextWritePosition (ViSession vi,  
ViConstString channelName, ViInt32 waveformHandle, ViInt32 relativeto,  
ViInt32 offset);
```

Purpose

Sets the position in the waveform at which the next waveform data is written. This function allows you to write to arbitrary locations within the waveform. These settings apply only to the next write to the waveform specified by the **waveformHandle** parameter. Subsequent writes to that waveform begin where the last write left off, unless this function is called again. The waveformHandle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or one of the following niFgen CreateWaveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is created by niFgen_init or the niFgen_InitWithOptions function for a particular instrument session.
channelName	ViConstString	Specifies the channel on which to the waveform is loaded. Default Value: "0"
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform allocated with the niFgen_AllocateWaveform function.
relativeTo	ViInt32	Specifies the reference position in the waveform. relativeTo and offset together determine where to start the waveform.
Defined Values		
NIFGEN_VAL_WAVEFORM_POSITION_		
NIFGEN_VAL_WAVEFORM_POSITION_		
offset	ViInt32	Specifies the offset from relativeTo at which to insert data into the waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteWaveform

```
ViStatus niFgen_WriteWaveform (ViSession vi, ViConstString channelName,  
ViInt32 waveformHandle, ViInt32 size, ViReal64[] data);
```

Purpose

Writes floating-point data to the waveform in onboard memory. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or one of the following niFgen CreateWaveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	The channel onto which the waveform data should be loaded. Default Value: "0"
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform previously allocated with the niFgen_AllocateWaveform function.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	ViInt64[]	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteWaveformComplexF64

```
ViStatus niFgen_WriteWaveformComplexF64 (ViSession vi,  
ViConstString channelName, ViInt32 numberOfSamples,  
niFgen_ComplexF64[] data, ViInt32 waveformHandle);
```

Purpose

Writes complex data to the waveform in onboard memory on devices with the [NIFGEN_ATTR_OUTPUT_ENABLED](#) attribute set to VI_TRUE and the [NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE](#) attribute set to NIFGEN_VAL_OSP_COMPLEX. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or to one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the waveform data should be loaded. Default Value: "0"
numberOfSamples	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	niFgen_ComplexF64[]	Specifies the array of data to load into the waveform. You must normalize the data points in the array to be between -1.00 and +1.00. The array must have at least as many elements as the value in the numberOfSamples parameter.
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform previously allocated with niFgen_AllocateWaveform .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteBinary16Waveform

```
ViStatus niFgen_WriteBinary16Waveform (ViSession vi,  
ViConstString channelName, ViInt32 waveformHandle, ViInt32 size,  
ViInt16[] data);
```

Purpose

Writes binary data to the waveform in onboard memory. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) or the [niFgen_CreateWaveformI16](#) function.

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel on which to the waveform data should be loaded. Default Value: "0"
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform previously allocated with the niFgen_AllocateWaveform function.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	ViInt16 []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size . The binary data is left-justified.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteComplexBinary16Waveform

```
ViStatus niFgen_WriteComplexBinary16Waveform (ViSession vi,  
ViConstString channelName, ViInt32 waveformHandle, ViInt32 size,  
NIComplexI16[] data);
```

Purpose

Writes binary data to the waveform in onboard memory. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) or the [niFgen_CreateWaveformI16](#) function.

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies channel on which to load the waveform data. Default Value: "0"
waveformHandle	ViInt32	Specifies the handle of the arbitrary waveform previously allocated with the niFgen_AllocateWaveform function.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	NIComplexI16 []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size . The binary data is left-justified.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Configure Clock Functions

Configure Reference Clock	niFgen_ConfigureReferenceClock
Configure Sample Clock Source	niFgen_ConfigureSampleClockSource
Configure Clock Mode	niFgen_ConfigureClockMode
Adjust Sample Clock Relative Delay	niFgen_AdjustSampleClockRelativeDelay

niFgen_ConfigureReferenceClock

```
ViStatus niFgen_ConfigureReferenceClock (ViSession vi,  
ViConstString referenceClockSource, ViReal64 referenceClockFrequency);
```

Purpose

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.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
referenceClockSource	ViConstString	Specifies the source for the Reference clock. 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. The Reference clock phase-locks with the signal generator Sample clock timebase to allow the frequency stability and accuracy of the sample clock timebase to match that of the Reference clock.



Note 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
"PXI_Clk10"	10 MHz PCI backplane reference clock
"ClkIn"	Clk In Front panel connector
"OnboardRefClk"	Onboard reference clock
"RTSI7"	RTSI line 7
"RefIn"	Ref In front panel connector

Default Value: "None"

referenceClockFrequency ViReal64

The reference clock frequency in hertz.

Default Value: 10000000

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureSampleClockSource

```
ViStatus niFgen_ConfigureSampleClockSource (ViSession vi,  
ViConstString sampleClockSource);
```

Purpose

Sets the source of the Sample clock of the signal generator. Samples are generated based on the the Sample clock cycles. The Sample clock source can be internal or external.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
sampleClockSource	ViConstString	Specifies the sample clock source that you want the signal generator to use.



Note 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 trigger line
"PXI_Trig0"	PXI trigger line 0 or RTSI line 0
"PXI_Trig1"	PXI trigger line 1 or RTSI line 1
"PXI_Trig2"	PXI trigger line 2 or RTSI line 2
"PXI_Trig3"	PXI trigger line 3

	or RTSI line 3
"PXI_Trig4"	PXI trigger line 4 or RTSI line 4
"PXI_Trig5"	PXI trigger line 5 or RTSI line 5
"PXI_Trig6"	PXI trigger line 6 or RTSI line 6
"PXI_Trig7"	PXI trigger line 7 or RTSI line 7
"DDC_ClkIn"	Sample clock from DDC connector

Default Value: "OnboardClock"

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureClockMode

ViStatus niFgen_ConfigureClockMode (ViSession vi, ViInt32 clockMode);

Purpose

Selects the clock mode for the signal generator.

For signal generators that support it, this function allows you to switch the sample clock to High-Resolution Sampling mode.

When in Divide-Down Sampling mode, you can only set the sample rate to certain frequencies, based on dividing down the update clock.

However, in High-Resolution Sampling mode, you can set the sample rate to any value.

-  **Note** The signal generator must not be in the Generating state when you call this function.
-  **Note** You must call the [niFgen_ConfigureOutputMode](#) function to set the output mode to NIFGEN_VAL_OUTPUT_SEQ, NIFGEN_VAL_OUTPUT_ARB, or NIFGEN_VAL_OUTPUT_SCRIPT before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
clockMode	ViInt32	Sets the clock mode of the signal generator.

Defined Values

NIFGEN_VAL_DIVIDE_DOWN	Divide down sampling — Sample rates are generated by dividing the source frequency.
NIFGEN_VAL_HIGH_RESOLUTION	High resolution sampling — 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
clocking
modes.

Default Value: Depends on the device

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_AdjustSampleClockRelativeDelay

```
ViStatus niFgen_AdjustSampleClockRelativeDelay (ViSession vi,  
ViReal64 adjustmentTime);
```

Purpose

Delays (or phase shifts) the sample clock, which delays the generated signal. Delaying the sample clock can be useful when synchronizing 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 function is called. To delay an external sample clock, use the [NIFGEN_ATTR_SAMPLE_CLOCK_ABSOLUTE_DELAY](#) attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
adjustmentTime	ViReal64	Specifies the amount of time to adjust the sample clock delay in seconds. Default Value: 0

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Triggering and Synchronization Functions

Configure Digital Edge Start Trigger	niFgen_ConfigureDigitalEdgeStartI
Configure Software Edge Start Trigger	niFgen_ConfigureSoftwareEdgeSta
Disable Start Trigger	niFgen_DisableStartTrigger
Configure Digital Edge Script Trigger	niFgen_ConfigureDigitalEdgeScript
Configure Digital Level Script Trigger	niFgen_ConfigureDigitalLevelScri
Configure Software Edge Script Trigger	niFgen_ConfigureSoftwareEdgeScri
Disable Script Trigger	niFgen_DisableScriptTrigger
Configure Trigger Mode	niFgen_ConfigureTriggerMode
Send Software Edge Trigger	niFgen_SendSoftwareEdgeTrigger
Disable Script Trigger	niFgen_DisableScriptTrigger
Configure Synchronization	niFgen_ConfigureSynchronization
Export Signal	niFgen_ExportSignal

niFgen_ConfigureDigitalEdgeStartTrigger

```
ViStatus niFgen_ConfigureDigitalEdgeStartTrigger (ViSession vi,  
ViConstString source, ViInt32 edge);
```

Purpose

Configures the Start trigger for digital edge triggering.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
source	ViConstString	Specifies which trigger source the signal generator uses.
edge	ViInt32	Specifies the edge to detect.

Defined Values

NIFGEN_VAL_RISING_EDGE	Occurs when the signal transitions from low level to high level.
NIFGEN_VAL_FALLING_EDGE	Occurs when the signal transitions from high level to low level.

Default Value:

NIFGEN_VAL_RISING_EDGE

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureSoftwareEdgeStartTrigger

ViStatus niFgen_ConfigureSoftwareEdgeStartTrigger (ViSession vi);

Purpose

Configures the Start trigger for software edge triggering.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DisableStartTrigger

ViStatus niFgen_DisableStartTrigger (ViSession vi);

Purpose

Disables the Start trigger.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureDigitalEdgeScriptTrigger

```
ViStatus niFgen_ConfigureDigitalEdgeScriptTrigger (ViSession vi,  
ViConstString triggerID, ViConstString source, ViInt32 edge);
```

Purpose

Configures the specified Script trigger for digital edge triggering.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
triggerID	ViConstString	Specifies the Script trigger used for triggering.
source	ViConstString	Specifies which trigger source the signal generator uses.
edge	ViInt32	Specifies the edge to detect.

Defined Values

NIFGEN_VAL_RISING_EDGE	Occurs when the signal transitions from low level to high level.
NIFGEN_VAL_FALLING_EDGE	Occurs when the signal transitions from high level to low level.

Default Value:

NIFGEN_VAL_RISING_EDGE

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureDigitalLevelScriptTrigger

```
ViStatus niFgen_ConfigureDigitalLevelScriptTrigger (ViSession vi,  
ViConstString triggerID, ViConstString source, ViInt32 triggerWhen);
```

Purpose

Configures the specified Script trigger for digital level triggering.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
triggerID	ViConstString	Specifies the Script trigger used for triggering.
source	ViConstString	Specifies which trigger source the signal generator uses.
triggerWhen	ViInt32	Configures the specified Script trigger for digital level triggering.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureSoftwareEdgeScriptTrigger

ViStatus niFgen_ConfigureSoftwareEdgeScriptTrigger (ViSession vi,
ViConstString triggerID);

Purpose

Configures the specified Script trigger for software edge triggering.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
triggerID	ViConstString	Specifies the Script trigger used for triggering.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DisableScriptTrigger

ViStatus niFgen_DisableScriptTrigger (ViSession vi, ViConstString triggerID);

Purpose

Disables the specified Script trigger.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
triggerID	ViConstString	Specifies the Script trigger used for triggering.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureTriggerMode

```
ViStatus niFgen_ConfigureTriggerMode (ViSession vi,  
ViConstString channelName, ViInt32 triggerMode);
```

Purpose

Sets the trigger mode for the signal generator.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the trigger mode. Default Value: "0"
triggerMode	ViInt32	Sets the trigger mode.

Defined Values

NIFGEN_VAL_SINGLE	The waveform that you describe in the sequence list generates only once by going through the entire staging list. Only one trigger is required to start the waveform generation. You can use single trigger mode with
-------------------	---

	<p>the output mode in any mode. After a trigger is received, the waveform generation starts from the first stage and continues through to the last stage.</p>
NIFGEN_VAL_CONTINUOUS	<p>The waveform that you describe in the staging list generates infinitely by repeatedly cycling through the staging list. After a trigger is received, the waveform generation starts from the first stage and continues</p>

	<p>through to the last stage. After the last stage is completed, the waveform generation loops back to the start of the first stage and continues until it is stopped. Only one trigger is required to start the waveform generation.</p>
NIFGEN_VAL_STEPPED	<p>After a start trigger is received, the waveform described by the first stage generates. Then, the device waits for the next trigger</p>

signal. On the next trigger, the waveform described by the second stage generates, and so on. Once the staging list is exhausted, the waveform generation returns to the first stage and continues to repeat the cycle.

NIFGEN_VAL_BURST

After a start trigger is received, the waveform described by the first stage generates until another trigger is received. At the next

trigger, the buffer of the previous stage completes, and then the waveform described by the second stage generates. Once the staging list is exhausted, the waveform generation returns to the first stage and continues to repeat the cycle. In Frequency List mode, the duration instruction is ignored, and the trigger switches the frequency

to the next
frequency
in the list.

Default Value:
NIFGEN_VAL_CONTINUOUS

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SendSoftwareEdgeTrigger

ViStatus niFgen_SendSoftwareEdgeTrigger (ViSession vi, viInt32 trigger,
viConstString triggerID);

Purpose

Sends a software edge to the specified trigger.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
trigger	ViInt32	Specifies the type of trigger to send.

Defined Values

NIFGEN_VAL_START_TRIGGER	Transitions the device from an Idle state to a Generation state where the device can respond to sample clocks.
NIFGEN_VAL_SCRIPT_TRIGGER	Allows you to create sophisticated generation operations that are determined by the generation script.

Default Value: NIFGEN_VAL_START_TRIGGER

triggerID ViConstString Identifies the specific trigger to be used.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ConfigureSynchronization

```
ViStatus niFgen_ConfigureSynchronization (ViSession vi,  
ViConstString channelName, ViInt32 synchronizationSource);
```

Purpose

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 the [niFgen_ExportSignal](#) function (use the [niFgen_RouteSignalOut](#) function for the NI 5404), and other signal generators should receive the signal by calling the [niFgen_ConfigureSynchronization](#) function.



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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the synchronization signal. Default Value: "0"
synchronizationSource ViInt32		Specify the source of the synchronization signal you want to

Defined Values

NIFGEN_VAL_NONE	No synchronization source
NIFGEN_VAL_RTSI_0	RTSI 0 or PXI_Trig 0
NIFGEN_VAL_RTSI_1	RTSI 1 or PXI_Trig 1
NIFGEN_VAL_RTSI_2	RTSI 2 or PXI_Trig 2
NIFGEN_VAL_RTSI_3	RTSI 3 or PXI_Trig 3
NIFGEN_VAL_RTSI_4	RTSI 4 or PXI_Trig 4
NIFGEN_VAL_RTSI_5	RTSI 5 or PXI_Trig 5
NIFGEN_VAL_RTSI_6	RTSI 6 or PXI_Trig 6

NIFGEN_VAL_TTL0	TTL 0
NIFGEN_VAL_TTL1	TTL 1
NIFGEN_VAL_TTL2	TTL 2
NIFGEN_VAL_TTL3	TTL 3
NIFGEN_VAL_TTL4	TTL 4
NIFGEN_VAL_TTL5	TTL 5
NIFGEN_VAL_TTL6	TTL 6

Default Value: NIFGEN_VAL_NOVAL

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ExportSignal

```
ViStatus niFgen_ExportSignal (ViSession vi, ViInt32 signal,  
ViConstString signalIdentifier, ViConstString outputTerminal);
```

Purpose

Exports various signals, clocks, and events from the signal generator to the RTSI lines, front panel, or other external terminals.



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

You can clear a previously routed signal by exporting the signal to "" (empty string).

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from niFgen_init or the niFgen_InitWithOptions function for a particular instrument session.
signal	ViInt32	Specifies the source of the signal to route.

Defined Values

NIFGEN_VAL_ONBOARD_REFERENCE_C

NIFGEN_VAL_SYNC_OUT

NIFGEN_VAL_START_TRIGGER

NIFGEN_VAL_MARKER_EVENT

NIFGEN_VAL_SAMPLE_CLOCK_TIMEBASE

NIFGEN_VAL_SYNCHRONIZATION

NIFGEN_VAL_SAMPLE_CLOCK
NIFGEN_VAL_REFERENCE_CLOCK
NIFGEN_VAL_SCRIPT_TRIGGER
NIFGEN_VAL_READY_FOR_START_EVENT
NIFGEN_VAL_STARTED_EVENT
NIFGEN_VAL_DONE_EVENT
NIFGEN_VAL_DATA_MARKER_EVENT

signalIdentifier ViConstString Specifies which instance of the selected signal to export.

Defined Values

"" (empty string)	Default (for non instance-specific values)
"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

outputTerminal ViConstString Specifies the output terminal to export the signal.



Note The following **Defined Values**

possible output terminals. For a complete list of the com terminals available on your device, refer to the **Terminals** tab in MAX.

Defined Values

"" (empty string)	Do not export signal
"PFI0"	PFI line 0
"PFI1"	PFI line 1
"PFI4"	PFI line 4
"PFI5"	PFI line 5
"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_ClkOut"	Clock out from DDC connector
"PXI_Star"	PXI Star trigger line

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_RouteSignalOut

```
ViStatus niFgen_RouteSignalOut (ViSession vi, ViConstString channelName,  
ViInt32 routeSignalFrom, ViInt32 routeSignalTo);
```

Purpose

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



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

You can clear a previously routed signal by routing NIFGEN_VAL_NONE to the destination terminal.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is one of the niFgen_InitWithOptions functions and identifies the instrument session.
channelName	ViConstString	Specifies the channel name for which you want to route signals.
routeSignalFrom	Vilnt32	Various signals can be routed out the RT output. See the Defined Values table below.

NIFGEN_VAL_NONE

NIFGEN_VAL_MARKER

NIFGEN_VAL_SYNC_OUT

NIFGEN_VAL_OUT_START_TRIGGER

NIFGEN_VAL_BOARD_CLOCK

NIFGEN_VAL_SYNCHRONIZATION

NIFGEN_VAL_SOFTWARE_TRIG

NIFGEN_VAL_OUT_UPDATE

NIFGEN_VAL_REF_OUT

NIFGEN_VAL_PXI_CLK10

NIFGEN_VAL_PXI_STAR

NIFGEN_VAL_PFI_0

NIFGEN_VAL_RTSI_0

NIFGEN_VAL_RTSI_1

NIFGEN_VAL_RTSI_2

NIFGEN_VAL_RTSI_3

NIFGEN_VAL_RTSI_4

NIFGEN_VAL_RTSI_5

NIFGEN_VAL_RTSI_6

NIFGEN_VAL_RTSI_7

NIFGEN_VAL_REF_CLOCK_RTSI_CLOCK

NIFGEN_VAL_ONBOARD_REFERENCE

NIFGEN_VAL_UPDATE_CLOCK

NIFGEN_VAL_PLL_REF_SOURCE

routeSignalTo Vilnt32

The possible RTSI lines to which you can route signals.

Defined Values

NIFGEN_VAL_RTSI_0

NIFGEN_VAL_RTSI_1

NIFGEN_VAL_RTSI_2

NIFGEN_VAL_RTSI_3

NIFGEN_VAL_RTSI_4

NIFGEN_VAL_RTSI_5
NIFGEN_VAL_RTSI_6
NIFGEN_VAL_RTSI_7
NIFGEN_VAL_REF_CLOCK_RTSI_CLOCK
NIFGEN_VAL_REF_OUT
NIFGEN_VAL_PFI_0
NIFGEN_VAL_PFI_1
NIFGEN_VAL_PFI_4
NIFGEN_VAL_PFI_5
NIFGEN_VAL_PXI_STAR
NIFGEN_VAL_PXI_CLK10

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Script Output Functions

Allocate Named Waveform	<u>niFgen_AllocateNamedWaveform</u>
Set Named Waveform Next Write Position	<u>niFgen_SetNamedWaveformNext</u>
Write Named Waveform F64	<u>niFgen_WriteNamedWaveformF64</u>
Write Named Waveform I16	<u>niFgen_WriteNamedWaveformI16</u>
Write Named Waveform Complex F64	<u>niFgen_WriteNamedWaveformComplexF64</u>
Write Named Waveform Complex I16	<u>niFgen_WriteNamedWaveformComplexI16</u>
Delete Named Waveform	<u>niFgen_DeleteNamedWaveform</u>
Write Script	<u>niFgen_WriteScript</u>
Delete Script	<u>niFgen_DeleteScript</u>

niFgen_AllocateNamedWaveform

```
ViStatus niFgen_AllocateNamedWaveform (ViSession vi,  
ViConstString channelName, ViConstString waveformName,  
ViInt32 waveformSize);
```

Purpose

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 niFgen Write (Binary16) Waveform functions.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to allocate the named waveform. Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.
waveformSize	ViInt32	Specifies the size of the waveform to allocate in samples.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetNamedWaveformNextWritePosition

```
ViStatus niFgen_SetNamedWaveformNextWritePosition (ViSession vi,  
ViConstString channelName, ViConstString waveformName, ViInt32 relativeto,  
ViInt32 offset);
```

Purpose

Sets the position in the waveform to which data is written at the next write. This function allows you to write to arbitrary locations within the waveform. These settings apply only to the next write to the waveform specified by the **waveformHandle** parameter. Subsequent writes to that waveform begin where the last write left off, unless this function is called again. The waveformHandle passed in must have been created with a call to one of the following functions:

- [niFgen_AllocateWaveform](#)
- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions function call.
channelName	ViConstString	Specifies the channel onto which the waveform is loaded.
		Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the waveform.
relativeTo	ViInt32	Specifies the reference position in the waveform. This and offset together determine where to start loading the waveform.
		Defined Values
		NIFGEN_VAL_WAVEFORM_POSITION_S
		NIFGEN_VAL_WAVEFORM_POSITION_C
offset	ViInt32	Specifies the offset from the relativeTo position at which to start loading the data into the waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteNamedWaveformF64

```
ViStatus niFgen_WriteNamedWaveformF64 (ViSession vi,  
ViConstString channelName, ViConstString waveformName, ViInt32 size,  
ViReal64[] data);
```

Purpose

Writes floating-point data to the waveform in onboard memory. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or to one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling [niFgen_SetWaveformNextWritePosition](#).

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the waveform data should be loaded. Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	ViReal64 []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteNamedWaveformI16

```
ViStatus niFgen_WriteNamedWaveformI16 (ViSession vi,  
ViConstString channelName, ViConstString waveformName, ViInt32 size,  
ViInt16[] data);
```

Purpose

Writes binary data to the named waveform in onboard memory. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or to one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the waveform data should be loaded. Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	ViInt16 []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteNamedWaveformComplexF64

```
ViStatus niFgen_WriteNamedWaveformComplexF64 (ViSession vi,  
ViConstString channelName, ViConstString waveformName, ViInt32 size,  
NIComplexNumber[] data);
```

Purpose

Writes complex floating-point data to the named waveform in onboard memory on devices with the OSP Enabled property set to VI_TRUE and the Data Processing Mode property set to Complex. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or to one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the waveform data should be loaded.
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	NIComplexNumber []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteNamedComplexWaveformI16

```
ViStatus niFgen_WriteNamedComplexWaveformI16 (ViSession vi,  
ViConstString channelName, ViConstString waveformName, ViInt32 size,  
NIComplexI16[] data);
```

Purpose

Writes complex binary data to the named waveform in onboard memory on devices with the OSP Enabled property set to VI_TRUE and the Data Processing Mode property set to Complex. The waveform handle passed in must have been created by a call to the [niFgen_AllocateWaveform](#) function or to one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

The write position and offset can be set by calling the [niFgen_SetWaveformNextWritePosition](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the waveform data should be loaded. Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.
size	ViInt32	Specifies the number of samples to load into the waveform. Default Value: 0
data	NIComplexI16 []	Specifies the array of data to load into the waveform. The array must have at least as many elements as the value in size .

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DeleteNamedWaveform

```
ViStatus niFgen_DeleteNamedWaveform (ViSession vi,  
ViConstString channelName, ViConstString waveformName);
```

Purpose

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 function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the named waveform is loaded. Default Value: "0"
waveformName	ViConstString	Specifies the name to associate with the allocated waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteScript

ViStatus niFgen_WriteScript (ViSession vi, ViConstString channelName,
ViConstString script);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel on which the script is loaded. Default Value: "0"
script	ViConstString	Contains the text of the script you want to use for your generation operation.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_DeleteScript

```
ViStatus niFgen_DeleteScript (ViSession vi, ViConstString channelName,  
ViConstString scriptName);
```

Purpose

Deletes the specified script from onboard memory.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel onto which the script is loaded. Default Value: "0"
scriptName	ViConstString	Specifies the name of the script you want to delete. The script name appears in the text of the script following the script keyword.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Configure Onboard Signal Processing Functions

Configure Custom FIR Filter Coefficients [niFgen_ConfigureCustomFIRFilte](#)

Get FIR Filter Coefficients [niFgen_GetFIRFilterCoefficients](#)

niFgen_ConfigureCustomFIRFilterCoefficients

```
ViStatus niFgen_ConfigureCustomFIRFilterCoefficients (ViSession vi,  
ViConstString channelName, ViInt32 numberofCoefficients,  
ViReal64[] coefficientsArray);
```

Purpose

Sets the FIR filter coefficients used by the NI 5441 onboard signal processing block. The values are coerced to the closest settings achievable by the signal generator.

Refer to [FIR Filter](#) for more information about FIR filter coefficients.



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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the operation mode. Defined Value: "0"
numberOfCoefficients	ViInt32	Specifies the number of coefficients. The NI 5441 requires 95.
coefficientsArray	ViReal64[]	Specifies the array of data the onboard signal processor uses for the FIR filter coefficients. For the NI 5441, provide a symmetric array of 95 coefficients to this parameter. The array must have at least as many elements as the value that you specify in the numberOfCoefficients parameter. The coefficients should range between -1.00 and +1.00.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetFIRFilterCoefficients

```
ViStatus niFgen_GetFIRFilterCoefficients (ViSession vi,  
ViConstString channelName, ViInt32 arraySize, ViReal64[] coefficientsArray,  
ViInt32* numberofCoefficientsRead);
```

Purpose

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 attribute (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_ConfigureCustomFIRFilterCoefficients](#) function coerced to the quantized values used by the device.

The coefficients are between –1.00 and +1.00.

To use this function, first call an instance of the niFgen_GetFIRFilterCoefficients function with the **coefficientsArray** parameter set to VI_NULL. Calling the function in this state returns the current size of the **coefficientsArray** as the value of the **numberOfCoefficientsRead** parameter. Create an array of this size, and call the niFgen_GetFIRFilterCoefficients function a second time, passing the new array as the **coefficientsArray** parameter and the size as the **arraySize** parameter. This second function call populate the array with the FIR filter coefficients.

Refer to [FIR Filter](#) for more information about FIR filter coefficients.

Default Value: None

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the operation mode. Defined Value: "0"
arraySize	ViInt32	Specifies the size of the coefficient array
coefficientsArray	ViReal64[]	Specifies the array of data the onboard signal processor uses for the FIR filter coefficients. For the NI 5441, provide a symmetric array of 95 coefficients to this parameter. The array must have at least as many elements as the value that you specify in the numberOfCoefficients parameter. The coefficients should range between -1.00 and +1.00.
numberOfCoefficientsRead	ViInt32[]	Specifies the array of data containing the number of

coefficients you want to read.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Set/Get/Check Attribute Functions

Set Attribute

Set Attribute ViInt32	niFgen_SetAttributeViInt32
Set Attribute ViReal64	niFgen_SetAttributeViReal64
Set Attribute ViString	niFgen_SetAttributeViString
Set Attribute ViBoolean	niFgen_SetAttributeViBoolean
Set Attribute ViSession	niFgen_SetAttributeViSession

Get Attribute

Get Attribute ViInt32	niFgen_GetAttributeViInt32
Get Attribute ViReal64	niFgen_GetAttributeViReal64
Get Attribute ViString	niFgen_GetAttributeViString
Get Attribute ViBoolean	niFgen_GetAttributeViBoolean
Get Attribute ViSession	niFgen_GetAttributeViSession

Check Attribute

Check Attribute ViInt32	niFgen_CheckAttributeViInt32
Check Attribute ViReal64	niFgen_CheckAttributeViReal64
Check Attribute ViString	niFgen_CheckAttributeViString
Check Attribute ViBoolean	niFgen_CheckAttributeViBoolean
Check Attribute ViSession	niFgen_CheckAttributeViSession

niFgen_SetAttributeViInt32

```
ViStatus niFgen_SetAttributeViInt32 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViInt32 attributeValue);
```

Purpose

Sets the value of a ViInt32 attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

NI-FGEN contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetAttributeViReal64

```
ViStatus niFgen_SetAttributeViReal64 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViReal64 attributeValue);
```

Purpose

Sets the value of a ViReal64 attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

NI-FGEN contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetAttributeViString

```
ViStatus niFgen_SetAttributeViString (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViConstString attributeValue);
```

Purpose

Sets the value of a ViString attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

NI-FGEN contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetAttributeViBoolean

```
ViStatus niFgen_SetAttributeViBoolean (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViBoolean attributeValue);
```

Purpose

Sets the value of a ViBoolean attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

NI-FGEN contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetAttributeViSession

```
ViStatus niFgen_SetAttributeViSession (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViSession attributeValue);
```

Purpose

Sets the value of a ViSession attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

NI-FGEN contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetAttributeViInt32

```
ViStatus niFgen_GetAttributeViInt32 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViInt32*(attributeValue);
```

Purpose

Queries the value of a ViInt32 attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string (""). Default Value: "" (empty string)

attributeID ViAttr Specifies the ID of an attribute.

Output

Name	Type	Description
attributeValue	Vilnt32*	Returns the current value of the attribute. Pass the address of a Vilnt32 variable.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetAttributeViReal64

```
ViStatus niFgen_GetAttributeViReal64 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViReal64* attributeValue);
```

Purpose

Queries the value of a ViReal64 attribute.

You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string (""). Default Value: "" (empty string)

attributeID ViAttr Specifies the ID of an attribute.

Output

Name	Type	Description
attributeValue	ViReal64*	Returns the current value of the attribute. Pass the address of a ViReal64 variable.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetAttributeViString

```
ViStatus niFgen_GetAttributeViString (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViInt32 arraySize,  
ViChar[] attributeValue);
```

Purpose

Queries the value of a ViString attribute.

You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

You must provide a ViChar array to serve as a buffer for the value. You pass the number of bytes in the buffer as the **arraySize** parameter. If the current value of the attribute, including the terminating NUL byte, is larger than the size you indicate in the **arraySize** parameter, the function copies **arraySize** – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the array size you must pass to get the entire value. For example, if the value is "123456" and **arraySize** is 4, the function places "123" into the buffer and returns 7.

If you want to call this function just to get the required array size, you can pass 0 for **arraySize** and **VI_NULL** for the **attributeValue** buffer.

If you want the function to fill in the buffer regardless of the number of bytes in the value, pass a negative number for the **arraySize** parameter.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
arraySize	ViInt32	Specifies the number of bytes in the ViChar array you specify for the attributeValue parameter.
		If the current value of the attribute, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies arraySize – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the array size you must pass to get the entire value. For example, if the value is "123456" and arraySize is 4, the function places "123" into the buffer and returns 7.
		If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.
		If you pass 0, you can pass VI_NULL for the attributeValue buffer parameter.

Output

Name	Type	Description
attributeValue	ViChar[]	<p>The buffer in which the function returns the current value of the attribute. The buffer must be a ViChar data type and have at least as many bytes as indicated in the arraySize parameter.</p> <p>If the current value of the attribute, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies <code>arraySize – 1</code> bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the array size you must pass to get the entire value. For example, if the value is "123456" and arraySize is 4, the function places "123" into the buffer and returns 7.</p> <p>If you specify 0 for the arraySize parameter, you can pass <code>VI_NULL</code> for this parameter.</p>

Return Value

Name	Type	Description
------	------	-------------

Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call niFgen_error_message . To obtain additional information about the error condition, call niFgen_GetError . To clear the error information from NI-FGEN, call niFgen_ClearError .
--------	----------	--

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetAttributeViBoolean

```
ViStatus niFgen_GetAttributeViBoolean (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViBoolean* attributeValue);
```

Purpose

Queries the value of a ViBoolean attribute.

You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string (""). Default Value: "" (empty string)

attributeID ViAttr Specifies the ID of an attribute.

Output

Name	Type	Description
attributeValue	ViBoolean*	Returns the current value of the attribute. Pass the address of a ViBoolean variable.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetAttributeViSession

```
ViStatus niFgen_GetAttributeViSession (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViSession* attributeValue);
```

Purpose

Queries the value of a ViSession attribute.

You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string (""). Default Value: "" (empty string)

attributeID ViAttr Specifies the ID of an attribute.

Output

Name	Type	Description
attributeValue	ViSession*	Returns the current value of the attribute. Pass the address of a ViSession variable.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CheckAttributeViInt32

```
ViStatus niFgen_CheckAttributeViInt32 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViInt32 attributeValue);
```

Purpose

Checks the validity of a value you specify for a VInt32 attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CheckAttributeViReal64

```
ViStatus niFgen_CheckAttributeViReal64 (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViReal64 attributeValue);
```

Purpose

Checks the validity of a value you specify for a ViReal64 attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CheckAttributeViString

```
ViStatus niFgen_CheckAttributeViString (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViConstString attributeValue);
```

Purpose

Checks the validity of a value you specify for a ViString attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CheckAttributeViBoolean

```
ViStatus niFgen_CheckAttributeViBoolean (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViBoolean attributeValue);
```

Purpose

Checks the validity of a value you specify for a ViBoolean attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CheckAttributeViSession

```
ViStatus niFgen_CheckAttributeViSession (ViSession vi,  
ViConstString channelName, ViAttr attributeID, ViSession attributeValue);
```

Purpose

Checks the validity of a value you specify for a ViSession attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel on which to check the attribute value if the attribute is channel-based. If the attribute is not channel-based, then pass VI_NULL or an empty string ("").
		Default Value: "" (empty string)
attributeID	ViAttr	Specifies the ID of an attribute.
attributeValue	ViInt32	Specifies the value to which you want to set the attribute.
		 Note Some of the values might not be valid depending on the current settings of the instrument session.
		Default Value: None

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Waveform Control Functions

Initiate Generation

[niFgen_InitiateGeneration](#)

Abort Generation

[niFgen_AbortGeneration](#)

Send Software Edge Trigger

[niFgen_SendSoftwareEdgeTrigger](#)

niFgen_InitiateGeneration

```
ViStatus niFgen_InitiateGeneration (ViSession vi);
```

Purpose

Initiates signal generation. If you want to abort signal generation, call the [niFgen_AbortGeneration](#) function. Once the signal generation is aborted, you can call the niFgen_InitiateGeneration function to cause the signal generator to produce a signal again.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_AbortGeneration

ViStatus niFgen_AbortGeneration (ViSession vi);

Purpose

Aborts any previously initiated signal generation. If you want to abort signal generation, call this function. Call the [niFgen_InitiateGeneration](#) function to cause the signal generator to produce a signal again.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Utility Functions

Get Hardware State	niFgen_GetHardwareState
Wait Until Done	niFgen_WaitUntilDone
Is Done	niFgen_IsDone
Reset With Defaults	niFgen_ResetWithDefaults
Reset Device	niFgen_ResetDevice
Get Channel Name	niFgen_GetChannelName
Disable	niFgen_Disable
Self-Test	niFgen_self_test
Revision Query	niFgen_revision_query

Coercion Info

Get Next Coercion Record [niFgen_GetNextCoercionRecord](#)

Error Info

Error-Query	niFgen_error_query
Error Handler	niFgen_ErrorHandler
Error Message	niFgen_error_message
Get Error	niFgen_GetError
Clear Error	niFgen_ClearError

Locking

Lock Session	niFgen_LockSession
Unlock Session	niFgen_UnlockSession

niFgen_GetHardwareState

ViStatus niFgen_GetHardwareState (ViSession vi, ViInt32* state);

Purpose

Returns the current hardware state of the device and, if the device is in the hardware error state, the current hardware error.



Note Hardware states do not necessarily correspond to NI-FGEN states.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
state	VlInt32*	Returns the Hardware state of the signal generator.

Defined Values

NIFGEN_VAL_IDLE	The generator is in the idle state.
NIFGEN_VAL_WAITING_FOR_START_TRIGGER	The generator is waiting for a start trigger.
NIFGEN_VAL_RUNNING	The generator is in the running state.
NIFGEN_VAL_DONE	The generator has completed successfully.
NIFGEN_VAL_HARDWARE_ERROR	The generator has encountered a hardware error.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WaitUntilDone

ViStatus niFgen_WaitUntilDone (ViSession vi, ViInt32 maxtime);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
maxTime	ViInt32	The timeout value (in milliseconds).

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_IsDone

ViStatus niFgen_IsDone (ViSession vi, ViBoolean* done);

Purpose

Determines whether the current generation is complete. This function sets the **done** parameter to VI_TRUE if the session is in the Idle or Committed states.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
done	ViBoolean*	Returns information about the completion of waveform generation.

Defined Values

VI_TRUE	Generation is complete.
VI_FALSE	Generation is not complete.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ResetWithDefaults

```
ViStatus niFgen_ResetWithDefaults (ViSession vi);
```

Purpose

Resets the instrument and reapplies initial user-specified settings from the logical name that was used to initialize the session. If the session was created without a logical name, this function is equivalent to the [niFgen_reset](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ResetDevice

```
ViStatus niFgen_ResetDevice (ViSession vi);
```

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetChannelName

ViStatus niFgen_GetChannelName (ViSession vi, ViInt32 index,
ViInt32 buffersize, ViChar[] channelName);

Purpose

The function returns the channel string that is in the channel table at an index you specify.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
index	ViInt32	Specifies a 1-based index into the channel table.
bufferSize	ViInt32	Specifies the size of the buffer for the channel string.

Output

Name	Type	Description
channelName	ViConstString	Returns the channel string that is in the channel table at the index you specify. Do not modify the contents of the channel string.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_Disable

ViStatus niFgen_Disable (ViSession vi);

Purpose

Places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_self_test

```
ViStatus niFgen_self_test (ViSession vi, ViInt16* selfTestResult,  
ViChar[] selfTestMessage);
```

Purpose

Runs the instrument self-test routine and returns the test result(s).



Note When used on some signal generators, the device is reset after the niFgen_self_test function runs. If you use the niFgen_self_test function, your device may not be in its previously configured state after the function runs.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
selfTestResult	Vilnt16	Contains the value returned from the instrument self-test. Zero means success. For any other code, refer to Error and Status Information .

Self-Test Code	Description
0	Passed self-test
1	Self-test failed

selfTestMessage ViChar[]	Returns the self-test response string from the instrument. Refer to Error and Status Information for an explanation of the string contents. You must pass a ViChar array with at least 256 bytes.
--------------------------	--

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_revision_query

```
ViStatus niFgen_revision_query (ViSession vi,  
ViChar[] instrumentDriverRevision, ViChar[] firmwareRevision);
```

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
instrumentDriverRevision	ViChar[]	Returns the NI-FGEN software revision numbers in the form of a string. You must pass a ViChar array with at least 256 bytes.
firmwareRevision	ViChar[]	Returns the instrument firmware revision numbers in the form of a string. You must pass a ViChar array with at least 256 bytes.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetNextCoercionRecord

ViStatus niFgen_GetNextCoercionRecord (ViSession vi, ViInt32 bufferSize,
ViChar[] coercionRecord);

Purpose

Returns the coercion information associated with the IVI session. This function retrieves and clears the oldest instance in which the NI-FGEN coerced a value you specified to another value.

If you set the [NIFGEN_ATTR_RECORD_COERCIONS](#) attribute to VI_TRUE, NI-FGEN keeps a list of all coercions it makes on ViInt32 or ViReal64 values that you pass to NI-FGEN functions. You use this function to retrieve information from that list.

If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies **bufferSize** – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and **bufferSize** is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI_NULL for the **coercionRecord** buffer parameter.

The function returns an empty string in the **coercionRecord** parameter if no coercion records remain for the session.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
bufferSize	ViInt32	<p>Specifies the number of bytes in the ViChar array you specify for the coercionRecord parameter.</p> <p>If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies bufferSize – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size that you must pass to get the entire value. For example, if the value is "123456" and the buffer size is 4, the function places "123" into the buffer and returns 7.</p> <p>If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.</p> <p>If you pass 0, you can pass VI_NULL for the coercionRecord buffer parameter.</p>

Default Value: None

Output

Name	Type	Description
coercionRecord	ViChar[]	<p>Returns the next coercion record for the IVI session. If there are no coercion records, the function returns an empty string.</p> <p>The buffer must contain at least as many elements as the value you specify with the</p>

bufferSize parameter. If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate with the **bufferSize** parameter, the function copies **bufferSize** – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size that you must pass to get the entire value. For example, if the value is "123456" and **bufferSize** is 4, the function places "123" into the buffer and returns 7.

This parameter returns an empty string if no coercion records remain for the session.

Return Value

Name	Type	Description
------	------	-------------

Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call niFgen_error_message . To obtain additional information about the error condition, call niFgen_GetError . To clear the error information from NI-FGEN, call niFgen_ClearError .
--------	----------	--

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_error_query

```
ViStatus niFgen_error_query (ViSession vi, ViInt32* errorCode,  
ViChar[] errorMessage);
```

Purpose

Reads an error code and a message from the instrument error queue.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
errorCode	Vilnt32*	Returns the error code read from the instrument error queue.
errorMessage	ViChar[]	Returns the error message string read from the instrument error message queue. You must pass a ViChar array with at least 256 bytes.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ErrorHandler

```
ViStatus niFgen_ErrorHandler (ViSession vi, ViStatus errorCode,  
ViChar[] errorMessage);
```

Purpose

Converts a status code returned by an NI-FGEN function into a user-readable string and returns any error elaborations.

Parameters

Input

Name	Type	Description
vi	ViSession	<p>Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.</p> <p>You can pass VI_NULL for this parameter. Passing VI_NULL is useful when one of the initialize functions fail.</p>
errorCode	ViStatus	<p>Specifies the status parameter that is returned from any of the NI-FGEN functions.</p> <p>Default Value: VI_SUCCESS</p>

Output

Name	Type	Description
errorMessage	ViChar[]	<p>Returns the error message string read from the instrument error message queue.</p> <p>You must pass a ViChar array with at least 256 bytes.</p>

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ErrorMessage

ViStatus niFgen_ErrorMessage (ViSession vi, ViStatus errorCode,
ViChar[] errorMessage);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	<p>Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.</p> <p>You can pass VI_NULL for this parameter. Passing VI_NULL is useful when one of the initialize functions fail.</p>
errorCode	ViStatus	<p>Specifies the status parameter that is returned from any of the NI-FGEN functions.</p> <p>Default Value: 0 (VI_SUCCESS)</p>

Output

Name	Type	Description
errorMessage	ViChar[]	<p>Returns the error message string read from the instrument error message queue.</p> <p>You must pass a ViChar array with at least 256 bytes.</p>

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetError

```
ViStatus niFgen_GetError (ViSession vi, ViStatus* errorCode,  
ViInt32 errorDescriptionBufferSize, ViChar[] errorDescription);
```

Purpose

Returns the error information associated with an IVI session or with the current execution thread. If you specify a valid IVI session for the **vi** parameter, this function retrieves and then clears the error information for the session. If you pass VI_NULL for the **vi** parameter, this function retrieves and then clears the error information for the current execution thread.

The IVI Engine also maintains this error information separately for each thread. This feature is useful if you do not have a session handle to pass to the niFgen_GetError or [niFgen_ClearError](#) functions. This occurs when a call to [niFgen_init](#) or [niFgen_InitWithOptions](#) fails.

Parameters

Input

Name	Type	Description
vi	ViSession	<p>Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.</p> <p>You can pass VI_NULL for this parameter. Passing VI_NULL is useful when one of the initialize functions fail.</p>
errorDescriptionBufferSize	ViInt32	<p>Default Value: VI_NULL</p> <p>Specifies the size of the errorDescription array.</p> <p>You can determine the array size needed to store the entire error description by setting this parameter to 0. The function then ignores the errorDescription buffer, which may be set to VI_NULL, and gives as its return value the required buffer size. You can then call the function a second time using the correct buffer size.</p>

Output

Name	Type	Description
errorCode	ViStatus*	<p>The error code for the session or execution thread.</p> <p>A value of VI_SUCCESS (0) indicates that no error occurred. A</p>

positive value indicates a warning. A negative value indicates an error.

You can call [niFgen_error_message](#) to get a text description of the value.

If you are not interested in this value, you can pass VI_NULL.

errorDescription

ViChar[]

The error description string for the session or execution thread. If the error code is non zero, the description string can further describe the error or warning condition.

If you are not interested in this value, you can pass VI_NULL. Otherwise, you must pass a ViChar array of a size specified with the **errorDescriptionBufferSize** parameter.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearError

```
ViStatus niFgen_ClearError (ViSession vi);
```

Purpose

Clears the error information for the current execution thread and the IVI session you specify. If you pass VI_NULL for the **vi** parameter, this function clears the error information only for the current execution thread.

This function sets the error code to VI_SUCCESS (0), and sets the error description string to "" (empty string) .

The IVI Engine also maintains this error information separately for each thread. This feature is useful if you do not have a session handle to pass to the [niFgen_ClearError](#) or the [niFgen_GetError](#) function, which occurs when a call to the [niFgen_init](#) or the [niFgen_InitWithOptions](#) function fails.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_LockSession

ViStatus niFgen_LockSession (ViSession vi, ViBoolean* callerHasLock);

Purpose

Obtains a multithread lock on the instrument session. Before it does so, this function waits until all other execution threads have released their locks on the instrument session.

Other threads might have obtained a lock on this session in the following ways:

- Your application called the `niFgen_LockSession` function.
- A call to the NI-FGEN locked the session.
- A call to the IVI Engine locked the session.

After your call to the `niFgen_LockSession` function returns successfully, no other threads can access the instrument session until you call the [`niFgen_UnlockSession`](#) function.

Use the `niFgen_LockSession` function and the `niFgen_UnlockSession` function around a sequence of calls to NI-FGEN functions if you require that the instrument retain its settings through the end of the sequence.

You can safely make nested calls to the `niFgen_LockSession` function within the same thread. To completely unlock the session, you must balance each call to the `niFgen_LockSession` function with a call to the `niFgen_UnlockSession` function. If, however, you use the **callerHasLock** parameter in all calls to the `niFgen_LockSession` function and the `niFgen_UnlockSession` function within a function, the IVI Engine locks the session only once within the function regardless of the number of calls you make to the `niFgen_LockSession` function. This configuration allows you to call the `niFgen_UnlockSession` function just once at the end of the function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
callerHasLock	ViBoolean*	Keeps track of whether you obtained a lock and therefore need to unlock the session. Pass the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to VI_FALSE. Pass the address of the same local variable to any other calls you make to the niFgen_LockSession function or the niFgen_UnlockSession function in the same function.

This parameter serves as a convenience. If you do not want to use this parameter, pass VI_NULL.

This parameter is an input/output parameter. The niFgen_LockSession function and the niFgen_UnlockSession function each inspect the current value and take the following actions:

- If the value is VI_TRUE, the niFgen_LockSession function does not lock the session again. If the value is VI_FALSE, the niFgen_LockSession function obtains the lock and sets the value of the parameter to VI_TRUE.
- If the value is VI_FALSE, the niFgen_UnlockSession function does

not attempt to unlock the session. If the value is VI_TRUE, the niFgen_UnlockSession function releases the lock and sets the value of the parameter to VI_FALSE.

Thus, you can call the niFgen_UnlockSession function at the end of your function without worrying about whether you actually have the lock.

Example:

```
ViStatus TestFunc (ViSession vi, ViInt32 flags)
{
    ViStatus error = VI_SUCCESS;
    ViBoolean haveLock = VI_FALSE;

    if (flags & BIT_1)
    {
        viCheckErr( niFgen_LockSession(vi,
            &haveLock));
        viCheckErr( TakeAction1(vi));
        if (flags & BIT_2)
        {
            viCheckErr( niFgen_UnlockSession(vi,
                &haveLock));
            viCheckErr( TakeAction2(vi));
            viCheckErr( niFgen_LockSession(vi,
                &haveLock));
        }
        if (flags & BIT_3)
            viCheckErr( TakeAction3(vi));
    }
}
```

Error:

```
/*
```

At this point, you cannot really be sure that you have the lock. Fortunately, the haveLock variable takes care of that for you.

```
*/
```

```
niFgen_UnlockSession(vi, &haveLock);  
return error;
```

```
}
```

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_UnlockSession

ViStatus niFgen_UnlockSession (ViSession vi, ViBoolean* callerHasLock);

Purpose

Releases a lock that you acquired on an instrument session using the [cviniFgen_LockSession](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Output

Name	Type	Description
callerHasLock	ViBoolean*	<p>Keeps track of whether you obtain a lock and therefore need to unlock the session.</p> <p>This parameter serves as a convenience. If you do not want to use this parameter, pass VI_NULL.</p> <p>Pass the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to VI_FALSE. Pass the address of the same local variable to any other calls you make to the niFgen_LockSession function or the niFgen_UnlockSession function in the same function.</p> <p>The parameter is an input/output parameter. The niFgen_LockSession function and the niFgen_UnlockSession function each inspect the current value and take the following actions:</p> <ul style="list-style-type: none">• If the value is VI_TRUE, the niFgen_LockSession function does not lock the session again. If the value is VI_FALSE, the niFgen_LockSession function obtains the lock and sets the value of the parameter to VI_TRUE.• If the value is VI_FALSE, the

niFgen_UnlockSession function does not attempt to unlock the session. If the value is VI_TRUE, the niFgen_UnlockSession function releases the lock and sets the value of the parameter to VI_FALSE.

Thus, you can, call the niFgen_UnlockSession function at the end of your function without worrying about whether you actually have the lock.

Example:

```
ViStatus TestFunc (ViSession vi, ViInt32 flags)
{
    ViStatus error = VI_SUCCESS;
    ViBoolean haveLock = VI_FALSE;

    if (flags & BIT_1)
    {
        viCheckErr(niFgen_LockSession(vi,
            &haveLock));
        viCheckErr( TakeAction1(vi));
        if (flags & BIT_2)
        {
            viCheckErr( niFgen_UnlockSession(vi,
                &haveLock));
            viCheckErr( TakeAction2(vi));
            viCheckErr( niFgen_LockSession(vi,
                &haveLock));
        }
        if (flags & BIT_3)
            viCheckErr( TakeAction3(vi));
    }
}
```

Error:

```
/*
At this point, you cannot really be sure that
you have the lock. Fortunately, the haveLock
variable takes care of that for you.
*/
niFgen_UnlockSession(vi, &haveLock);
return error;
}
```

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Interchangeability Info Functions

Get Next Interchange Warning [niFgen_GetNextInterchangeWarning](#)

Clear Interchange Warnings [niFgen_ClearInterchangeWarnings](#)

Reset Interchange Check [niFgen_ResetInterchangeCheck](#)

niFgen_GetNextInterchangeWarning

ViStatus niFgen_GetNextInterchangeWarning (ViSession vi, ViInt32 bufferSize,
ViChar[] interchangeWarning);

Purpose

Returns the interchangeability warnings associated with the IVI session. It retrieves and clears the oldest instance in which the class driver recorded an interchangeability warning. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. Use this function to retrieve interchangeability warnings.

NI-FGEN performs interchangeability checking when the [NIFGEN_ATTR_INTERCHANGE_CHECK](#) attribute is set to VI_TRUE.

The function returns an empty string in the **interchangeWarning** parameter if no interchangeability warnings remain for the session.

In general, NI-FGEN generates interchangeability warnings when an attribute that affects the behavior of the instrument is in a state that you did not specify.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
bufferSize	ViInt32	<p>Specifies the number of bytes in the ViChar array you specify for the interchangeWarning parameter.</p> <p>If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies bufferSize – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size that you must pass to get the entire value. For example, if the value is "123456" and bufferSize is 4, the function places "123" into the buffer and returns 7.</p> <p>If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.</p> <p>If you pass 0, you can pass VI_NULL for the interchangeWarning buffer parameter.</p>

Default Value: None

Output

Name	Type	Description
interchangeWarning	ViChar[]	Returns the next interchange warning

for the IVI session. If there are no interchange warnings, the function returns an empty string.

The buffer must contain at least as many elements as the value you specify with the **bufferSize** parameter. If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate with the **bufferSize** parameter, the function copies **bufferSize** – 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size that you must pass to get the entire value. For example, if the value is "123456" and **bufferSize** is 4, the function places "123" into the buffer and returns 7.

This parameter returns an empty string if no interchangeability warnings remain for the session.

Return Value

Name	Type	Description
------	------	-------------

Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call niFgen_error_message . To obtain additional information about the error condition, call niFgen_GetError . To clear the error information from NI-FGEN, call niFgen_ClearError .
--------	----------	--

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ClearInterchangeWarnings

ViStatus niFgen_ClearInterchangeWarnings (ViSession vi);

Purpose

Clears the list of current interchange warnings.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ResetInterchangeCheck

ViStatus niFgen_ResetInterchangeCheck (ViSession vi);

Purpose

When developing a complex test system that consists of multiple test modules, it is generally a good idea to design the test modules so that they can run in any order. To do so requires ensuring that each test module completely configures the state of each instrument it uses. If a particular test module does not completely configure the state of an instrument, the state of the instrument depends on the configuration from a previously executed test module. If you execute the test modules in a different order, the behavior of the instrument and therefore the entire test module is likely to change. This change in behavior is generally instrument specific and represents an interchangeability problem.

You can use this function to test for such cases. After you call this function, the interchangeability checking algorithms in the specific driver ignore all previous configuration operations. By calling this function at the beginning of a test module, you can determine whether the test module has dependencies on the operation of previously executed test modules.

This function does not clear the interchangeability warnings from the list of previously recorded interchangeability warnings. If you want to guarantee that the [niFgen_GetNextInterchangeWarning](#) function only returns those interchangeability warnings that are generated after calling this function, you must clear the list of interchangeability warnings. You can clear the interchangeability warnings list by repeatedly calling the [niFgen_GetNextInterchangeWarning](#) function until no more interchangeability warnings are returned. If you are not interested in the content of those warnings, you can call the [niFgen_ClearInterchangeWarnings](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Calibration Functions

Initialize External Calibration

[niFgen_InitExtCal](#)

Close External Calibration

[niFgen_CloseExtCal](#)

Self Calibrate

[niFgen_SelfCal](#)

Analog Output Calibration

Initialize Analog Output Calibration

[niFgen_InitializeAnalogO](#)

Write Binary 16 Analog Static Value

[niFgen_WriteBinary16An](#)

Cal Adjust Main Path Pre Amp Offset

[niFgen_CalAdjustMainPa](#)

Cal Adjust Main Path Pre Amp Gain

[niFgen_CalAdjustMainPa](#)

Cal Adjust Main Path Post Amp Gain And Offset

[niFgen_CalAdjustMainPa](#)

Cal Adjust Direct Path Gain

[niFgen_CalAdjustDirectP](#)

Cal Adjust Main Path Output Impedance

[niFgen_CalAdjustMainPa](#)

Cal Adjust Direct Path Output Impedance

[niFgen_CalAdjustDirectP](#)

Oscillator Calibration

Initialize Oscillator Frequency Calibration

[niFgen_InitializeOscillato](#)

Cal Adjust Oscillator Frequency

[niFgen_CalAdjustOscillat](#)

Cal ADC Calibration

Initialize Cal ADC Calibration

[niFgen_InitializeCalADC](#)

Write Binary 16 Analog Static Value

[niFgen_WriteBinary16An](#)

Cal Adjust Cal ADC

[niFgen_CalAdjustCalAD](#)

Read Cal ADC

[niFgen_ReadCalADC](#)

Calibration Utility

Get Self Cal Supported

[niFgen_GetSelfCalSuppor](#)

Get Self Cal Last Date And Time

[niFgen_GetSelfCalLastDa](#)

Get Ext Cal Last Date And Time

[niFgen_GetExtCalLastDa](#)

Get Self Cal Last Temp

[niFgen_GetSelfCalLastTe](#)

Get Ext Cal Last Temp

[niFgen_GetExtCalLastTe](#)

Get Ext Cal Recommended Interval

[niFgen_GetExtCalRecom](#)

Change Ext Cal Password	<u>niFgen_ChangeExtCalPas</u>
Set Cal User Defined Info	<u>niFgen_SetCalUserDefine</u>
Get Cal User Defined Info	<u>niFgen_GetCalUserDefine</u>
Get Cal User Defined Info Max Size	<u>niFgen_GetCalUserDefine</u>
Read Current Temperature	<u>niFgen_ReadCurrentTemp</u>
Restore Last Ext Cal Constants	<u>niFgen_RestoreLastExtCa</u>

niFgen_InitExtCal

```
ViStatus niFgen_InitExtCal (ViRsrc resourceName, ViConstString password,  
ViSession* vi);
```

Purpose

Creates and initializes a special NI-FGEN external calibration session. The ViSession returned is an NI-FGEN session that can be used to configure the device using normal attributes and functions. However, flags have been set that allow you to program an external calibration procedure using the special calibration attributes and functions.

Parameters

Input

Name	Type	Description
resourceName	ViRsrc	Specifies the resource name of the device

Example #	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1
2	NI-DAQmx device	<i>myDAQmxDevice</i>
3	NI-DAQmx device	DAQ:: <i>myDAQmxDevice</i>
4	NI-DAQmx device	DAQ::2
5	IVI logical name or IVI virtual name	<i>myLogicalName</i>

For Traditional NI-DAQ devices, the syntax is the device number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 2. NI-DAQmx devices in MAX are Dev1 or PX. To change an NI-DAQmx device by right-clicking on the device and entering a new name.

An alternate syntax for NI-DAQmx devices is *DAQmx device name*, as shown in Example 3. This convention allows for the use of an NI-DAC application that was originally designed for

device. For example, if the application expects to rename the NI-DAQmx device to 1 in MAX, change the resource name, as shown in Example 1.

If you use the DAQ::*n* syntax and an NI-D/A already exists with that same name, the NI matched first.

You can also pass in the name of an IVI logical name or virtual name configured with the IVI Config in Example 5. A logical name identifies a physical instrument. A virtual name identifies a specification. The initial settings for the session.



Caution Traditional NI-DAQ and NI- are not case-sensitive. However, all logical names, are case-sensitive. If driver session names, or virtual names must ensure that the name you use in the IVI Configuration Store file exactly, with the case of the characters.

password **ViConstString** The calibration password required to open session to the device.

Output

Name	Type	Description
vi	ViSession	Returns a session handle that you can use all subsequent NI-FGEN function calls.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CloseExtCal

ViStatus niFgen_CloseExtCal (ViSession vi, ViInt32 action);

Purpose

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.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
action	ViInt32	Specifies the action to perform upon closing.

Defined Values

NIFGEN_VAL_EXT_CAL_ABORT	No changes are made to the calibration constants and data in the EEPROM.
NIFGEN_VAL_EXT_CAL_COMMIT	The new calibration constants and data determined during the external calibration session are stored in the onboard EEPROM.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SelfCal

ViStatus niFgen_SelfCal (ViSession vi);

Purpose

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.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Analog Output Calibration Functions

Initialize Analog Output Calibration	niFgen_InitializeAnalogO
Write Binary 16 Analog Static Value	niFgen_WriteBinary16An
Cal Adjust Main Path Pre Amp Offset	niFgen_CalAdjustMainPa
Cal Adjust Main Path Pre Amp Gain	niFgen_CalAdjustMainPa
Cal Adjust Main Path Post Amp Gain And Offset	niFgen_CalAdjustMainPa
Cal Adjust Direct Path Gain	niFgen_CalAdjustDirectP
Cal Adjust Main Path Output Impedance	niFgen_CalAdjustMainPa
Cal Adjust Direct Path Output Impedance	niFgen_CalAdjustDirectP

niFgen_InitializeAnalogOutputCalibration

```
ViStatus niFgen_InitializeAnalogOutputCalibration (ViSession vi);
```

Purpose

This function sets up the device to start the analog output calibration.
The session handle should be the handle returned by [niFgen_InitExtCal](#).

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_WriteBinary16AnalogStaticValue

```
ViStatus niFgen_WriteBinary16AnalogStaticValue (ViSession vi,  
ViConstString channelName, ViInt16 value);
```

Purpose

Writes a 16-bit value to the DAC, which can be generated as a DC voltage. This function writes to the DAC only within an external calibration session.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to write the value. Default Value: "0"
value	ViInt16	Specifies the value to write.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustMainPathPreAmpOffset

```
ViStatus niFgen_CalAdjustMainPathPreAmpOffset (ViSession vi,  
ViConstString channelName, ViInt32 configuration, ViInt32[] gainDACValues,  
ViInt32[] offsetDACValues, ViReal64[] measuredOutputs);
```

Purpose

Calculates calibration constants pertaining to the preamplifier 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. Pass the configuration data, as well as the measurements, to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the pre-amp stage configuration.
gainDACValues	ViInt32[]	Specifies an array of the values programmed to the gain calibration DAC during this calibration stage.
offsetDACValues	ViInt32[]	Specifies an array of the values programmed to the offset calibration DAC during this calibration stage.
measuredOutputs	ViReal64[]	Specifies an array of the analog output voltages measured during this calibration stage.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustMainPathPreAmpGain

```
ViStatus niFgen_CalAdjustMainPathPreAmpGain (ViSession vi,  
ViConstString channelName, ViInt32 configuration, ViInt32[] mainDACValues,  
ViInt32[] gainDACValues, ViInt32[] offsetDACValues,  
ViReal64[] measuredOutputs);
```

Purpose

Calculates calibration constants pertaining to the preamplifier 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 voltage. Pass the configuration data, as well as the measurements to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the preamplifier stage configuration.
mainDACValues	ViInt32[]	Specifies an array of the values programmed to the main output DAC during this calibration stage.
gainDACValues	ViInt32[]	Specifies an array of the values programmed to the gain calibration DAC during this calibration stage.
offsetDACValues	ViInt32[]	Specifies an array of the values programmed to the offset calibration DAC during this calibration stage.
measuredOutputs	ViReal64[]	Specifies an array of the analog output voltages measured during this calibration stage.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustMainPathPostAmpGainAndOffset

```
ViStatus niFgen_CalAdjustMainPathPostAmpGainAndOffset (ViSession vi,  
ViConstString channelName, ViInt32 configuration, ViInt32[] mainDACValues,  
ViInt32[] gainDACValues, ViInt32[] offsetDACValues,  
ViReal64[] measuredOutputs);
```

Purpose

Calculates calibration constants pertaining to the postamplifier 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. Pass the configuration data, as well as the measurements to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the postamplifier stage configuration.
mainDACValues	ViInt32[]	Specifies an array of the values programmed to the main output DAC during this calibration stage.
gainDACValues	ViInt32[]	Specifies an array of the values programmed to the gain calibration DAC during this calibration stage.
offsetDACValues	ViInt32[]	Specifies an array of the values programmed to the offset calibration DAC during this calibration stage.
measuredOutputs	ViReal64[]	Specifies an array of the analog output voltages measured during this calibration stage.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustDirectPathGain

```
ViStatus niFgen_CalAdjustDirectPathGain (ViSession vi,  
ViConstString channelName, ViInt32[] mainDACValues,  
ViInt32[] gainDACValues, ViReal64[] measuredOutputs);
```

Purpose

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 voltage. Pass the configuration data, as well as the measurements, to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
mainDACValues	ViInt32[]	Specifies an array of the values programmed to the main output DAC during this calibration stage.
gainDACValues	ViInt32[]	Specifies an array of the values programmed to the gain calibration DAC during this calibration stage.
measuredOutputs	ViReal64[]	Specifies an array of the analog output voltages measured during this calibration stage.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustMainPathOutputImpedance

```
ViStatus niFgen_CalAdjustMainPathOutputImpedance (ViSession vi,  
ViConstString channelName, ViInt32 configuration, ViReal64 loadImpedance,  
ViReal64 measuredSourceVoltage, ViReal64 measuredVoltageAcrossLoad);
```

Purpose

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. Pass the configuration data, as well as the measurements, to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the main path output impedance configuration.
loadImpedance	ViReal64	Specifies the impedance of load across which the measurement passed in as measuredVoltageAcrossLoad is taken.
measuredSourceVoltage	ViReal64	Specifies the analog output voltage measured across a very high-impedance load.
measuredVoltageAcrossLoad	ViReal64	Specifies the analog output voltage measured across the load impedance specified in loadImpedance parameter.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustDirectPathOutputImpedance

```
ViStatus niFgen_CalAdjustDirectPathOutputImpedance (ViSession vi,  
ViConstString channelName, ViInt32 configuration, ViReal64 loadImpedance,  
ViReal64 measuredSourceVoltage, ViReal64 measuredVoltageAcrossLoad);
```

Purpose

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. Pass the configuration data, as well as the measurements, to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the direct path output impedance configuration.
loadImpedance	ViReal64	Specifies the impedance of load across which the measurement passed in as measuredVoltageAcrossL is taken.
measuredSourceVoltage	ViReal64	Specifies the analog output voltage measured across a very high-impedance load.
measuredVoltageAcrossLoad	ViReal64	Specifies the analog output voltage measured across the load impedance specified in loadImpedance parameter.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Oscillator Calibration Functions

Initialize Oscillator Frequency Calibration [niFgen_InitializeOscillatorFrequency](#)
Cal Adjust Oscillator Frequency [niFgen_CalAdjustOscillatorFrequency](#)

niFgen_InitializeOscillatorFrequencyCalibration

ViStatus niFgen_InitializeOscillatorFrequencyCalibration (ViSession vi);

Purpose

Sets up the device to start the VCXO calibration.

The session handle should be the handle returned by the [niFgen_InitExtCal](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustOscillatorFrequency

```
ViStatus niFgen_CalAdjustOscillatorFrequency (ViSession vi,  
ViReal64 desiredFrequency, ViReal64 measuredFrequency);
```

Purpose

Calculates calibration constants pertaining to the VCXO. During external calibration, you can generate sine waves and take measurements of the resulting output frequency. Pass the desired and measured frequencies to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_InitExtCal</u> function and identifies a particular instrument session.
desiredFrequency	ViReal64	Specifies the expected frequency of the output waveform.
measuredFrequency	ViReal64	Specifies the actual (measured) frequency of the output waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Cal ADC Calibration Functions

Initialize Cal ADC Calibration	<u>niFgen_InitializeCalADCCalibration</u>
Write Binary 16 Analog Static Value	<u>niFgen_WriteBinary16AnalogStaticValue</u>
Cal Adjust Cal ADC	<u>niFgen_CalAdjustCalADC</u>
Read Cal ADC	<u>niFgen_ReadCalADC</u>

niFgen_InitializeCalADCCalibration

```
ViStatus niFgen_InitializeCalADCCalibration (ViSession vi);
```

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustCalADC

```
ViStatus niFgen_calAdjustCalADC (ViSession vi,  
ViReal64[] voltagesMeasuredExternally,  
ViReal64[] voltagesMeasuredWithCalADC);
```

Purpose

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. Pass the measured voltages to this function to allow NI-FGEN to calculate the appropriate calibration constants and store them in the onboard EEPROM when the calibration session is committed.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
voltagesMeasuredExternally	ViReal64[]	Specifies an array of analog output voltages measured with an external instrument.
voltagesMeasuredWithCalADC	ViReal64[]	Specifies an array of analog output voltages measured with the onboard calibration ADC.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ReadCalADC

```
ViStatus niFgen_ReadCalADC (ViSession vi,  
ViInt32 numberofReadstoAverage, ViBoolean returnCalibratedValue,  
ViReal64* calADCValue);
```

Purpose

Takes one or more voltage measurements from the onboard calibration ADC and returns the value or the average value. The signal that the ADC actually measures can be specified using the [NIFGEN_ATTR_CAL_ADC_INPUT](#) attribute. 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 **returnCalibratedValue** parameter is VI_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.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.
numberOfReadsToAverage	ViInt32	Specifies the number of measurements to be taken and averaged to determine the return value.
returnCalibratedValue	ViBoolean	Specifies whether the voltage returned from the ADC should be adjusted to account for the gain and offset of the ADC.

Output

Name	Type	Description
calADCValue	ViReal64*	Specifies the average of the voltage measurements taken from the onboard calibration ADC.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Flatness Calibration Functions

Initialize Flatness Calibration [niFgen_InitializeFlatnessCalibration](#)

Cal Adjust Flatness [niFgen_CalAdjustFlatness](#)

niFgen_InitializeFlatnessCalibration

ViStatus niFgen_InitializeFlatnessCalibration (ViSession vi);

Purpose

Initializes an external calibration session to calibrate flatness.

The session handle should be the handle returned by the [niFgen_InitExtCal](#) function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_CalAdjustFlatness

ViStatus niFgen_CalAdjustFlatness (ViSession vi, ViConstString channelName,
ViInt32 configuration, ViReal64 requestedAmplitude, ViReal64[] frequencies,
ViReal64[] measuredAmplitudes, ViInt32 numberOfMeasurements);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.
channelName	ViConstString	Specifies the name of the channel being calibrated. Default Value: "0"
configuration	ViInt32	Specifies the main path configuration.
requestedAmplitude	ViReal64	Specifies the desired amplitude of the output waveform.
frequencies	ViReal64[]	Specifies the frequencies of the output waveform.
measuredAmplitudes	ViReal64[]	Specifies the actual (measured) amplitudes of the output waveform.
numberOfMeasurements	ViInt32	Specifies the number of measurements to take.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Calibration Utility Functions

Get Self Cal Supported	<u>niFgen_GetSelfCalSupported</u>
Get Self Cal Last Date And Time	<u>niFgen_GetSelfCalLastDateAndTime</u>
Get Ext Cal Last Date And Time	<u>niFgen_GetExtCalLastDateAndTime</u>
Get Self Cal Last Temp	<u>niFgen_GetSelfCalLastTemp</u>
Get Ext Cal Last Temp	<u>niFgen_GetExtCalLastTemp</u>
Get Ext Cal Recommended Interval	<u>niFgen_GetExtCalRecommendedInterval</u>
Change Ext Cal Password	<u>niFgen_ChangeExtCalPassword</u>
Set Cal User Defined Info	<u>niFgen_SetCalUserDefinedInfo</u>
Get Cal User Defined Info	<u>niFgen_GetCalUserDefinedInfo</u>
Get Cal User Defined Info Max Size	<u>niFgen_GetCalUserDefinedInfoMaxSize</u>
Read Current Temperature	<u>niFgen_ReadCurrentTemperature</u>
Restore Last Ext Cal Constants	<u>niFgen_RestoreLastExtCalConstants</u>

niFgen_GetSelfCalSupported

```
ViStatus niFgen_GetSelfCalSupported (ViSession vi,  
ViBoolean* selfCalSupported);
```

Purpose

Returns whether the device supports self-calibration.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions function and identifies a particular instrument session.

Output

Name	Type	Description
selfCalSupported	ViBoolean*	Returns whether the device supports self-calibration.

Defined Values

VI_TRUE	Self–calibration is supported.
VI_FALSE	Self–calibration is not supported.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetSelfCalLastDateAndTime

```
ViStatus niFgen_GetSelfCalLastDateAndTime (ViSession vi, ViInt32* year,  
ViInt32* month, ViInt32* day, ViInt32* hour, ViInt32* minute);
```

Purpose

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 function returns 14 for the **hours** parameter and 30 for the **minutes** parameter.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
year	ViInt32*	Specifies the year of the last successful calibration.
month	ViInt32*	Specifies the month of the last successful calibration.
day	ViInt32*	Specifies the day of the last successful calibration.
hour	ViInt32*	Specifies the hour of the last successful calibration.
minute	ViInt32*	Specifies the minute of the last successful calibration.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetExtCalLastDateAndTime

ViStatus niFgen_GetExtCalLastDateAndTime (ViSession vi, ViInt32* year,
ViInt32* month, ViInt32* day, ViInt32* hour, ViInt32* minute);

Purpose

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 returns 14 for the **hours** parameter and 30 for the **minutes** parameter.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
year	ViInt32*	Specifies the year of the last successful calibration.
month	ViInt32*	Specifies the month of the last successful calibration.
day	ViInt32*	Specifies the day of the last successful calibration.
hour	ViInt32*	Specifies the hour of the last successful calibration.
minute	ViInt32*	Specifies the minute of the last successful calibration.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetSelfCalLastTemp

ViStatus niFgen_GetSelfCalLastTemp (ViSession vi, ViReal64* temperature);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
temperature	ViReal64*	Specifies the temperature at the last successful calibration.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetExtCalLastTemp

ViStatus niFgen_GetExtCalLastTemp (ViSession vi, ViReal64* temperature);

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
temperature	ViReal64*	Specifies the temperature at the last successful calibration.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetExtCalRecommendedInterval

```
ViStatus niFgen_GetExtCalRecommendedInterval (ViSession vi,  
ViInt32* months);
```

Purpose

Returns the recommended interval between external calibrations in months.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
months	ViInt32*	Specifies the recommended interval between external calibrations in months.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ChangeExtCalPassword

```
ViStatus niFgen_ChangeExtCalPassword (ViSession vi,  
ViConstString oldPassword, ViConstString newPassword);
```

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitExtCal</u> function and identifies a particular instrument session.
oldPassword	ViConstString	Specifies the old (current) external calibration password.
newPassword	ViConstString	Specifies the new (desired) external calibration password.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_SetCalUserDefinedInfo

ViStatus niFgen_SetCalUserDefinedInfo (ViSession vi, ViConstString info);

Purpose

Stores user-defined information in the onboard EEPROM. Call the [`niFgen_GetCalUserDefinedInfoMaxSize`](#) function to determine the number of characters that can be stored.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.
info	ViConstString	Specifies the information string that should be stored in the EEPROM.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetCalUserDefinedInfo

ViStatus niFgen_GetCalUserDefinedInfo (ViSession vi, ViString info);

Purpose

Retrieves user-defined information from the onboard EEPROM. Call the [niFgen_GetCalUserDefinedInfoMaxSize](#) function to determine the number of characters that can be retrieved. The buffer you provide should be the size of the maximum number of characters stored plus one termination character.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.
info	ViString	Specifies a string into which the user information is copied. This parameter must point to a character buffer large enough to hold the information string.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_GetCalUserDefinedInfoMaxSize

ViStatus niFgen_GetCalUserDefinedInfoMaxSize (ViSession vi,
ViInt32* infoSize);

Purpose

Returns the maximum number of characters (excluding the termination character) of user-defined information that can be stored in the onboard EEPROM.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
infoSize	ViInt32*	Specifies the maximum number of characters of user defined info that can be stored in the onboard EEPROM.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_ReadCurrentTemperature

```
ViStatus niFgen_ReadCurrentTemperature (ViSession vi,  
ViReal64* temperature);
```

Purpose

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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitExtCal function and identifies a particular instrument session.

Output

Name	Type	Description
Temperature	ViReal64*	Returns the current temperature read from onboard temperature sensors (degrees Celsius).

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niFgen_RestoreLastExtCalConstants

ViStatus niFgen_RestoreLastExtCalConstants (ViSession vi);

Purpose

Overwrites the current calibration constants with those from the last successful external calibration. This action effectively undoes any self-calibrations performed since the last time an external calibration was performed. This function should be used if a self-calibration produced invalid calibration constants.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_InitExtCal function and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE] niFgen_ConfigureTriggerSource

```
ViStatus niFgen_ConfigureTriggerSource (ViSession vi,  
ViConstString channelName, ViInt32 triggerSource);
```

Purpose

Configures the trigger source. The signal generator responds to a trigger depending on the operation mode in which the signal generator is operating.



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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
channelName	ViConstString	Specifies the channel name for which you want to configure the trigger source. Defined Value: "0"
triggerSource	ViInt32	Controls which trigger source the signal generator uses.

Defined Values

NIFGEN_VAL_IMMEDIATE	Immediate
NIFGEN_VAL_EXTERNAL	External (maps to PFI 0)
NIFGEN_VAL_SOFTWARE_TRIG	Software trigger
NIFGEN_VAL_PXI_STAR	PXI Star
NIFGEN_VAL_RTSI_0	RTSI 0 o PXI_Trig
NIFGEN_VAL_RTSI_1	RTSI 1 o PXI_Trig
NIFGEN_VAL_RTSI_2	RTSI 2 o PXI_Trig
NIFGEN_VAL_RTSI_3	RTSI 3 o PXI_Trig
NIFGEN_VAL_RTSI_4	RTSI 4 o PXI_Trig
NIFGEN_VAL_RTSI_5	RTSI 5 o PXI_Trig

NIFGEN_VAL_RTSI_6	RTSI 6 o PXI_Trig
NIFGEN_VAL_RTSI_7	RTSI 7 o PXI_Trig
NIFGEN_VAL_TTL0	TTL 0
NIFGEN_VAL_TTL1	TTL 1
NIFGEN_VAL_TTL2	TTL 2
NIFGEN_VAL_TTL3	TTL 3
NIFGEN_VAL_TTL4	TTL 4
NIFGEN_VAL_TTL5	TTL 5
NIFGEN_VAL_TTL6	TTL 6
NIFGEN_VAL_PFI_0	PFI 0
NIFGEN_VAL_PFI_1	PFI 1
NIFGEN_VAL_PFI_2	PFI 2
NIFGEN_VAL_PFI_3	PFI 3

Default Value: NIFGEN_VAL_IMMEDIATE

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE] niFgen_CreateArbWaveform

```
ViStatus niFgen_CreateArbWaveform (ViSession vi, ViInt32 waveformSize,  
ViReal64[] waveformdataArray, ViInt32* waveformHandle);
```

Purpose

[OBSOLETE] This function is obsolete. Use the [niFgen_CreateWaveformF64](#), [niFgen_CreateWaveformI16](#), or [niFgen_CreateWaveformComplexF64](#) function instead of this function.

Creates an arbitrary waveform and returns a handle that identifies that waveform. You can pass this handle to the [niFgen_ConfigureArbWaveform](#) function to produce that waveform. You can also use the handles this function returns to specify a sequence of arbitrary waveforms with the [niFgen_CreateArbSequence](#) function.

-  **Note** You must set the output mode to NIFGEN_VAL_OUTPUT_ARB before calling this function.
-  **Note** You must scale the data between –1.00 and +1.00. Use the **arbGain** parameter to generate different output voltages.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
waveformSize	ViInt32	Specifies the size of the arbitrary waveform that you want created.

The size must meet the following restrictions:

- The size must be less than or equal to the maximum waveform size that the device allows.
- The size must be greater than or equal to the minimum waveform size that the device allows.
- The size must be an integer multiple of the device waveform quantum.

You can obtain these values from the **maximumWaveformSize**, **minimumWaveformSize**, and **waveformQuantum** parameters in the [niFgen_QueryArbWfmCapabilities](#) function.

Default Value: None

waveformDataArray ViReal64[] Specifies the array of data you want to

use for the new arbitrary waveform. The array must have at least as many elements as the value that you specify in **waveformSize**.

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

Default Value: None

Output

Name	Type	Description
waveformHandle	VlInt32*	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE]

niFgen_CreateBinary16ArbWaveform

```
ViStatus niFgen_CreateBinary16ArbWaveform (ViSession vi,  
ViInt32 waveformSize, ViInt16[] waveformdataArray,  
ViInt32* waveformHandle);
```

Purpose

[OBSOLETE] This function is obsolete. Use the [niFgen_CreateWaveformI16](#) function instead of this function.

Creates an arbitrary waveform from binary data and returns a handle that identifies that waveform. You can pass this handle to the [niFgen_ConfigureArbWaveform](#) function to produce that waveform. You can also use the handles this function returns to specify a sequence of arbitrary waveforms with the [niFgen_CreateArbSequence](#) function.



Note You must set the output mode to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ before calling this function.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the <u>niFgen_init</u> or the <u>niFgen_InitWithOptions</u> functions and identifies a particular instrument session.
waveformSize	ViInt32	Specifies the size of the arbitrary waveform that you want created.

The size must meet the following restrictions:

- The size must be less than or equal to the maximum waveform size that the device allows.
- The size must be greater than or equal to the minimum waveform size that the device allows.
- The size must be an integer multiple of the device waveform quantum.

You can obtain these values from the **maximumWaveformSize**, **minimumWaveformSize**, and **waveformQuantum** parameters in [niFgen_QueryArbWfmCapabilities](#).

Default Value: None

waveformDataArray ViReal64[] Specifies the array of data you want to use for the new arbitrary waveform. The array must have at least as many

elements as the value that you specify in **waveformSize**.

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

Default Value: None

Output

Name	Type	Description
waveformHandle	VlInt32*	The handle that identifies the new waveform. This handle is used later when referring to this waveform.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE] niFgen_ConfigureRefClockSource

ViStatus niFgen_ConfigureRefClockSource (ViSession vi,
ViInt32 referenceClockSource);

Purpose

Configures the signal generator reference clock source. The signal generator uses the reference clock to derive frequencies and sample rates when generating waveforms.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is the niFgen_init or the niFgen_InitWithOp and identifies a particular instrument session.
referenceClockSource	ViInt32	<p>Specifies the reference clock source the signal generator to use. NI-FGEN sets NIFGEN_ATTR_REF_CLOCK_SOURCE this value.</p> <p>The signal generator derives the frequency and sample rates that it uses to generate waveforms from the source you specify.</p> <p>For example, when you set this parameter to NIFGEN_VAL_REF_CLOCK_INTERNAL, the generator uses the signal it receives at its internal clock terminal as the reference clock.</p>

Defined Values

NIFGEN_VAL_REF_CLOCK_INTERNAL
NIFGEN_VAL_REF_CLOCK_EXTERNAL
NIFGEN_VAL_REF_CLOCK_RTSI_CLK
NIFGEN_VAL_REF_CLOCK_TTL7
NIFGEN_VAL_PXI_CLK10
NIFGEN_VAL_REF_IN

NIFGEN_VAL_RTSI_0
NIFGEN_VAL_RTSI_1
NIFGEN_VAL_RTSI_2
NIFGEN_VAL_RTSI_3
NIFGEN_VAL_RTSI_4
NIFGEN_VAL_RTSI_5
NIFGEN_VAL_RTSI_6
NIFGEN_VAL_RTSI_7
NIFGEN_VAL_CLK_IN

Default Value:

NIFGEN_VAL_REF_CLOCK_INTERNAL

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE]

niFgen_ConfigureRefClockFrequency

ViStatus niFgen_ConfigureRefClockFrequency (ViSession vi,
ViReal64 referenceClockFrequency);

Purpose

Configures the signal generator reference clock frequency. The signal generator uses the reference clock to derive frequencies and sample rates when generating waveforms.



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

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
referenceClockFrequency	ViReal64	The reference clock frequency in Hz. Default Value: 10000000

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE]

niFgen_ConfigureUpdateClockSource

ViStatus niFgen_ConfigureUpdateClockSource (ViSession vi,
ViInt32 updateClockSource);

Purpose

Sets the source of the update clock of the signal generator. The source can be internal or external.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.
updateClockSource	ViInt32	Specifies the update clock source.

Defined Values

NIFGEN_VAL_INTERNAL	Internal clock source
NIFGEN_VAL_EXTERNAL	External clock source
NIFGEN_VAL_PXI_STAR	PXI Star
NIFGEN_VAL_RTSI_0	RTSI 0 or PXI_Trig 0
NIFGEN_VAL_RTSI_1	RTSI 1 or PXI_Trig 1
NIFGEN_VAL_RTSI_2	RTSI 2 or PXI_Trig 2
NIFGEN_VAL_RTSI_3	RTSI 3 or PXI_Trig 3
NIFGEN_VAL_RTSI_4	RTSI 4 or PXI_Trig 4
NIFGEN_VAL_RTSI_5	RTSI 5 or

	PXI_Trig 5
NIFGEN_VAL_RTSI_6	RTSI 6 or PXI_Trig 6
NIFGEN_VAL_RTSI_7	RTSI 7 or PXI_Trig 7
NIFGEN_VAL_CLK_IN	CLK IN front panel connector
NIFGEN_VAL_DDC_CLK_IN	Digital Data & Control clock in

Default Value:

NIFGEN_VAL_INTERNAL

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

[OBSOLETE] niFgen_SendSoftwareTrigger

ViStatus niFgen_SendSoftwareTrigger (ViSession vi);

Purpose

Sends a command to trigger the signal generator.



Note This function can act as an override for an external edge trigger. However, the NI 5401/5411/5431 do not support overriding an external digital edge trigger.

Parameters

Input

Name	Type	Description
vi	ViSession	Identifies your instrument session. vi is obtained from the niFgen_init or the niFgen_InitWithOptions functions and identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You can examine the status code from each call to an NI-FGEN function to determine if an error occurred. To obtain a text description of the status code, call the niFgen_error_message function. To obtain additional information about the error condition, call the niFgen_GetError function. To clear the error information from NI-FGEN, call the niFgen_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Return Value

Reports the status of this operation. To obtain a text description of the status code, call [niFgen_error_message](#). To obtain additional information concerning the error condition, use [niFgen_GetError](#) and [niFgen_ClearError](#).

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Expand this Book for a list of Attributes in this Group

The following topics provide more information about the attributes in this group. The topics include data types, descriptions, defined values, important notes, and other information.

NIFGEN_ATTR_ARB_SEQUENCE_HANDLE

IviFgenArbSeq Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W Channel None [niFgen_ConfigureArbSequence](#)

Description

This channel-based attribute identifies which sequence the signal generator produces. You can create multiple sequences using the [niFgen_CreateArbSequence](#) function. The niFgen_CreateArbSequence function returns a handle that you can use to identify the particular sequence. To configure the signal generator to produce a particular sequence, set this attribute to the sequence handle.

Use this attribute only when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_SEQ.

NIFGEN_ATTR_MAX_LOOP_COUNT

IviFgenArbSeq Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbSeqCapabilitie](#)

Description

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

NIFGEN_ATTR_MAX_NUM_SEQUENCES

IviFgenArbSeq Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbSeqCapabilitie](#)

Description

Returns the maximum number of arbitrary sequences that the signal generator allows. Typically, this value is constant for the signal generator.

NIFGEN_ATTR_MAX_SEQUENCE_LENGTH

IviFgenArbSeq Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbSeqCapabilitie](#)

Description

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

NIFGEN_ATTR_MIN_SEQUENCE_LENGTH

IviFgenArbSeq Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbSeqCapabilitie](#)

Description

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

NIFGEN_ATTR_ARB_GAIN

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureArbWaveform](#)

Description

This channel-based attribute 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 attribute to scale the arbitrary waveform to other ranges.

For example, when you set this attribute to 2.0, the output signal ranges from –2.0 V to +2.0 V.

Use this attribute when the [**NIFGEN_ATTR_OUTPUT_MODE**](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB, NIFGEN_VAL_OUTPUT_SCRIPT or NIFGEN_VAL_OUTPUT_SEQ.

NIFGEN_ATTR_ARB_WAVEFORM_HANDLE

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W Channel None [niFgen_ConfigureArbWaveform](#)

Description

This channel-based attribute identifies which arbitrary waveform the signal generator produces. You can create multiple arbitrary waveforms using one of the following niFgen Create Waveform functions:

- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)

These functions return a handle that you can use to identify the particular waveform. To configure the signal generator to produce a particular waveform, set this attribute to the waveform handle.

Use this attribute only when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB.

NIFGEN_ATTR_ARB_MARKER_POSITION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the position for a marker to be asserted in the arbitrary waveform. This attribute defaults to -1 when no marker position is specified. Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB.

Use the [niFgen_ExportSignal](#) function to export the marker signal.

NIFGEN_ATTR_ARB_OFFSET

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureArbWaveform](#)

Description

This channel-based attribute specifies the value that 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 attribute to shift the arbitrary waveform range.

For example, when you set this attribute to 1.0, the output signal ranges from 2.0 V to 0.0 V.

Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB, NIFGEN_VAL_OUTPUT_SCRIPT, or NIFGEN_VAL_OUTPUT_SEQ.

Units: volts

NIFGEN_ATTR_ARB_REPEAT_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies number of times to repeat the arbitrary waveform when **triggerMode** parameter of the [niFgen_ConfigureTriggerMode](#) function is set to NIFGEN_VAL_SINGLE or NIFGEN_VAL_STEPPED. This attribute is ignored if the **triggerMode** is set to NIFGEN_VAL_CONTINUOUS or NIFGEN_VAL_BURST. Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB.

When used during [streaming](#), this attribute specifies the number of times to repeat the streaming waveform (the onboard memory allocated for streaming).

NIFGEN_ATTR_DATA_TRANSFER_BLOCK_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_DIRECT_DMA_ENABLED

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enable the device for direct DMA writes. When enabled, all Create Waveform and Write Waveform calls that are given a data address in the direct DMA window will download data residing on the the direct DMA device to the NI signal generator onboard memory.

NIFGEN_ATTR_DIRECT_DMA_WINDOW_ADDRESS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Direct DMA window address.

NIFGEN_ATTR_DIRECT_DMA_WINDOW_SIZE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Direct DMA window size, in bytes *not* in samples.

NIFGEN_ATTR_FILE_TRANSFER_BLOCK_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	Yes	None

Description

The number of samples at a time to read from the file and download to onboard memory. Used in conjunction with the [niFgen_CreateWaveformI16](#) and [niFgen_WriteWaveform](#) functions.

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

NIFGEN_ATTR_STREAMING_WAVEFORM_HAND

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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

Used in conjunction with the

[NIFGEN_ATTR_STREAMING_SPACE_AVAILABLE_IN_WAVEFORM](#)
attribute.

NIFGEN_ATTR_STREAMING_SPACE_AVAILABLE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

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

[NIFGEN_ATTR_STREAMING_WAVEFORM_HANDLE](#) or
[NIFGEN_ATTR_STREAMING_WAVEFORM_NAME](#) attributes.

NIFGEN_ATTR_MAX_NUM_WAVEFORMS

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbWfmCapabiliti](#)

Description

Returns the maximum number of arbitrary waveforms that the signal generator allows. Typically, this value is constant for the signal generator.

NIFGEN_ATTR_MAX_WAVEFORM_SIZE

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbWfmCapabiliti](#)

Description

Returns the size (in samples) of the largest waveform that can be created. This attribute reflects the space currently available, taking into account previously allocated waveforms and instructions.

NIFGEN_ATTR_MIN_WAVEFORM_SIZE

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbWfmCapabiliti](#)

Description

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

NIFGEN_ATTR_OSP_CIC_FILTER_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enables or disables the CIC filter.



Note The `NIFGEN_ATTR_OSP_CIC_FILTER_ENABLED` and `NIFGEN_ATTR_OSP_FIR_FILTER_ENABLED` attributes must have the same enable/disable setting.

NIFGEN_ATTR_OSP_CIC_FILTER_GAIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

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 attribute overrides the value set by NI-FGEN.

NIFGEN_ATTR_OMP_CIC_FILTER_INTERPOLATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

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 [NIFGEN_ATTR_OMP_IQ_RATE](#) attribute.

NIFGEN_ATTR_OSP_FIR_FILTER_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enables or disables the FIR filter.



Note The [NIFGEN_ATTR_OSP_CIC_FILTER_ENABLED](#) and [NIFGEN_ATTR_OSP_FIR_FILTER_ENABLED](#) attributes must have the same enable/disable setting.

NIFGEN_ATTR_OSP_FIR_FILTER_INTERPOLATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

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 [NIFGEN_ATTR_OMP_IQ_RATE](#) attribute.

NIFGEN_ATTR_OMP_OVERFLOW_ERROR_REPC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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 [NIFGEN_ATTR_OSP_OVERFLOW_STATUS](#) attribute to query for overflow conditions whether the [NIFGEN_ATTR_OSP_OVERFLOW_ERROR_REPORTING](#) attribute is enabled. The device continues to generate after an overflow, regardless of the value of the [NIFGEN_ATTR_OSP_OVERFLOW_ERROR_REPORTING](#) attribute is enabled.

Defined Values

NIFGEN_VAL_ERROR_REPORTING_ERROR	NI-FGEN returns errors whenever an overflow has occurred in the OSP block.
NIFGEN_VAL_ERROR_REPORTING_DISABLED	NI-FGEN does not return errors when an overflow occurs in the OSP block.

NIFGEN_ATTR_OSP_OVERFLOW_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Returns a bit field of the overflow status in any stage of the OSP block. This attribute is functional regardless of the value for the [NIFGEN_ATTR_OSP_OVERFLOW_ERROR_REPORTING](#) attribute.

Write 0 to this attribute to clear the current [NIFGEN_ATTR_OSP_OVERFLOW_ERROR_REPORTING](#) value.

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

NIFGEN_ATTR_OSP_CARRIER_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enables or disables generation of the carrier.

NIFGEN_ATTR_OSP_CARRIER_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the frequency of the generated carrier.

NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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

Defined Values

NIFGEN_VAL_OSP_REAL	The waveform data points are real numbers (I data).
NIFGEN_VAL_OSP_COMPLEX	The waveform data points are complex numbers (IQ data).

NIFGEN_ATTR_OSP_FIR_FILTER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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

NIFGEN_VAL_OSP_FLAT	Applies a flat filter to the data specified in the NIFGEN_ATTR_OSP_FIR_F attribute.
NIFGEN_VAL_OSP_RAISED_COSINE	Applies a raised cosine filter specified in the NIFGEN_ATTR_OSP_FIR_F attribute.
NIFGEN_VAL_OSP_ROOT_RAISED_COSINE	Applies a root raised cosine filter specified in the NIFGEN_ATTR_OSP_FIR_F attribute.
NIFGEN_VAL_OSP_GAUSSIAN	Applies a Gaussian filter to the data specified in the NIFGEN_ATTR_OSP_FIR_F attribute.
NIFGEN_VAL_OSP_CUSTOM	Applies a custom filter to the data. If this is selected, you must provide the filter coefficients by using the niFgen_ConfigureCustomFIR function.

NIFGEN_ATTR_OSP_FIR_FILTER_FLAT_PASSBA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the passband value to use when calculating the FIR filter coefficients. The FIR filter is designed to be flat to passband x IQ rate. This attribute is only used when the [**NIFGEN_ATTR_OSP_FIR_FILTER_TYPE**](#) attribute is set to NIFGEN_VAL_OSP_FLAT.

NIFGEN_ATTR_OSP_FIR_FILTER_GAUSSIAN_B1

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the BT value to use when calculating the pulse-shaping FIR filter coefficients. Only used when the [NIFGEN_ATTR_OSP_FIR_FILTER_TYPE](#) attribute is set to NIFGEN_VAL_OSP_GAUSSIAN.

NIFGEN_ATTR_OMP_FIR_FILTER_RAISED_COSII

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the alpha value to use when calculating the pulse shaping FIR filter coefficients. Only used when the [NIFGEN_ATTR_OSP_FIR_FILTER_TYPE](#) attribute is set to NIFGEN_VAL_OSP_RAISED_COSINE.

NIFGEN_ATTR_OMP_FIR_FILTER_ROOT_RAISEC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the alpha value to use when calculating the pulse-shaping FIR filter coefficients. Only used when the [NIFGEN_ATTR_OSP_FIR_FILTER_TYPE](#) attribute is set to NIFGEN_VAL_OSP_ROOT_RAISED_COSINE.

NIFGEN_ATTR_OSP_IQ_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the rate at which the user-provided waveform data is generated when the [NIFGEN_ATTR_OSP_ENABLED](#) attribute is set to VI_TRUE. NI-FGEN sets the [NIFGEN_ATTR_ARB_SAMPLE_RATE](#) attribute of the signal generator to the product of the NIFGEN_ATTR_OSP_IQ_RATE, [NIFGEN_ATTR_OSP_FIR_FILTER_INTERPOLATION](#), and [NIFGEN_ATTR_OSP_CIC_FILTER_INTERPOLATION](#) attributes. When the [NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE](#) attribute is set to NIFGEN_VAL_OSP_REAL, the NIFGEN_ATTR_OSP_IQ_RATE attribute is the rate at which the signal generator processes real (I) data. When the [NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE](#) attribute is set to NIFGEN_VAL_OSP_COMPLEX, the NIFGEN_ATTR_OSP_IQ_RATE is the rate at which the signal generator processes complex (IQ) data.

NIFGEN_ATTR_OSP_CARRIER_PHASE_I

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the I carrier phase in degrees at the first point of the generation

NIFGEN_ATTR_OSP_CARRIER_PHASE_Q

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the Q carrier phase in degrees at the first point of the generation. This attribute is only used when the [NIFGEN_ATTR_OSP_DATA_PROCESSING_MODE](#) attribute is set to NIFGEN_VAL_OSP_COMPLEX.

Default Value: -90.0

NIFGEN_ATTR_OMP_PRE_FILTER_GAIN_I

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_OMP_PRE_FILTER_GAIN_Q

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the digital gain to apply to the Q data stream before any filtering by the OSP block. This attribute is only used when the [NIFGEN_ATTR OSP DATA PROCESSING MODE](#) attribute is set to NIFGEN_VAL_OSP_COMPLEX.

Default Value: 1.0

Values: Range is from -2.0 to 2.0

NIFGEN_ATTR_OMP_PRE_FILTER_OFFSET_I

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_OMP_PRE_FILTER_OFFSET_Q

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the digital offset to apply to the Q data stream. This offset is applied after the prefilter gain and before any filtering. This attribute is only used when the [NIFGEN_ATTR_OMP_DATA_PROCESSING_MODE](#) attribute is set to NIFGEN_VAL_OMP_COMPLEX.

Default Value: 0.0

Values: Range is from -1.0 to 1.0

NIFGEN_ATTR_OSP_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_ARB_SAMPLE_RATE

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W N/A None [niFgen_ConfigureSampleRate](#)

Description

Specifies the rate at which the signal generator outputs the points in arbitrary waveforms. Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_ARB or NIFGEN_VAL_OUTPUT_SEQ.

Units: S/s

NIFGEN_ATTR_WAVEFORM_QUANTUM

IviFgenArbWfm Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 RO N/A None [niFgen_QueryArbWfmCapabiliti](#)

Description

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

For example, when this attribute returns a value of 4, all waveform sizes must be a multiple of 4. Typically, this value is constant for the signal generator.

NIFGEN_ATTR_OPERATION_MODE

IviFgenBase Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W Channel None [niFgen_ConfigureOperationMod](#)

Description

Determines how the signal generator produces waveforms. NI signal generators currently support only one value. To control trigger mode, use [NIFGEN_ATTR_TRIGGER_MODE](#) instead.

Defined Value: NIFGEN_VAL_OPERATE_CONTINUOUS

NIFGEN_ATTR_OUTPUT_ENABLED

IviFgenBase Attribute

Data Type Access Applies to Coercion High-Level Functions

ViBoolean R/W Channel None [niFgen_ConfigureOutputEnabled](#)

Description

Controls whether the signal that the signal generator produces appears at the channel output connector.

NIFGEN_ATTR_OUTPUT_IMPEDANCE

IviFgenBase Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureOutputImpedance](#)

Description

Controls the impedance value that the signal generator uses.

Defined Values

50.0 75.0

Default Value: 50.0

NIFGEN_ATTR_OUTPUT_MODE

IviFgenBase Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W N/A None [niFgen_ConfigureOutputMode](#)

Description

Sets which output mode that the signal generator uses. The value that you specify determines which functions and attributes you can use to configure the waveform that the signal generator produces.

Defined Values

NIFGEN_VAL_OUTPUT_FUNC	Standard Function mode —Generates standard function waveforms such as sine, square, triangle, and so on.
NIFGEN_VAL_OUTPUT_FREQ_LIST	Frequency List mode —Generates a standard function using a list of frequencies you define.
NIFGEN_VAL_OUTPUT_ARB	Arbitrary Waveform mode —Generates waveforms from user-created/provided waveform arrays of numeric data.
NIFGEN_VAL_OUTPUT_SEQ	Arbitrary Sequence mode —Generates downloaded waveforms in an order you specify.
NIFGEN_VAL_OUTPUT_SCRIPT	Script mode —Allows you to use scripting to link and loop multiple waveforms in complex combinations.

Default Value: NIFGEN_VAL_OUTPUT_FUNC

Standard Function

When you set this attribute to NIFGEN_VAL_OUTPUT_FUNC, you can use the following functions to configure the waveform:

- [niFgen_ConfigureStandardWaveform](#)
- [niFgen_DefineUserStandardWaveform](#)
- [niFgen_ClearUserStandardWaveform](#)
- [niFgen_ConfigureFrequency](#)

- [niFgen_ConfigureAmplitude](#)

And you can set the following attributes:

- [NIFGEN_ATTR_FUNC_WAVEFORM](#)
- [NIFGEN_ATTR_FUNC_AMPLITUDE](#)
- [NIFGEN_ATTR_FUNC_DC_OFFSET](#)
- [NIFGEN_ATTR_FUNC_FREQUENCY](#)
- [NIFGEN_ATTR_FUNC_START_PHASE](#)
- [NIFGEN_ATTR_FUNC_DUTY_CYCLE_HIGH](#)

Frequency List

When you set this attribute to NIFGEN_VAL_OUTPUT_FREQ_LIST, you can use the following functions to configure the waveform:

- [niFgen_CreateFreqList](#)
- [niFgen_ClearFreqList](#)
- [niFgen_QueryFreqListCapabilities](#)
- [niFgen_ConfigureFreqList](#)
- [niFgen_ConfigureAmplitude](#)

And you can set the following attributes:

- [NIFGEN_ATTR_FREQ_LIST_HANDLE](#)
- [NIFGEN_ATTR_FUNC_AMPLITUDE](#)
- [NIFGEN_ATTR_FUNC_DC_OFFSET](#)
- [NIFGEN_ATTR_FUNC_START_PHASE](#)

Arbitrary Waveform

When you set this attribute to NIFGEN_VAL_OUTPUT_ARB, you can use the following functions to configure the waveform:

- [niFgen_QueryArbWfmCapabilities](#)
- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)
- [niFgen_ClearArbWaveform](#)
- [niFgen_ClearArbMemory](#)

- [niFgen_ConfigureSampleRate](#)
- [niFgen_ConfigureGain](#)
- [niFgen_ConfigureArbWaveform](#)
- [niFgen_ConfigureClockMode](#)

And you can set the following attributes:

- [NIFGEN_ATTR_ARB_WAVEFORM_HANDLE](#)
- [NIFGEN_ATTR_ARB_GAIN](#)
- [NIFGEN_ATTR_ARB_OFFSET](#)
- [NIFGEN_ATTR_ARB_SAMPLE_RATE](#)
- [NIFGEN_ATTR_CLOCK_MODE](#)

Arbitrary Sequence

When you set this parameter to NIFGEN_VAL_OUTPUT_SEQ, you can use the following functions to configure the sequence:

- [niFgen_QueryArbWfmCapabilities](#)
- [niFgen_CreateWaveformF64](#)
- [niFgen_CreateWaveformI16](#)
- [niFgen_CreateWaveformFromFileI16](#)
- [niFgen_CreateWaveformFromFileF64](#)
- [niFgen_CreateWaveformFromFileHWS](#)
- [niFgen_ClearArbWaveform](#)
- [niFgen_ClearArbMemory](#)
- [niFgen_QueryArbSeqCapabilities](#)
- [niFgen_CreateArbSequence](#)
- [niFgen_CreateAdvancedArbSequence](#)
- [niFgen_ClearArbSequence](#)
- [niFgen_ConfigureSampleRate](#)
- [niFgen_ConfigureClockMode](#)
- [niFgen_ConfigureArbSequence](#)
- [niFgen_ConfigureGain](#)

And you can set the following attributes:

- [NIFGEN_ATTR_ARB_SEQUENCE_HANDLE](#)
- [NIFGEN_ATTR_ARB_GAIN](#)

- [NIFGEN_ATTR_ARB_OFFSET](#)
- [NIFGEN_ATTR_ARB_SAMPLE_RATE](#)
- [NIFGEN_ATTR_CLOCK_MODE](#)

Script

When you set this parameter to NIFGEN_VAL_OUTPUT_SCRIPT, you can use the following functions to configure the script:

- [niFgen_AllocateNamedWaveform](#)
- [niFgen_SetNamedWaveformNextWritePosition](#)
- [niFgen_WriteNamedWaveformF64](#)
- [niFgen_WriteNamedWaveformI16](#)
- [niFgen_WriteNamedWaveformComplexF64](#)
- [niFgen_WriteNamedWaveformI16](#)
- [niFgen_DeleteNamedWaveform](#)
- [niFgen_WriteScript](#)
- [niFgen_DeleteScript](#)

And you can set the following attribute:

- [NIFGEN_ATTR_SCRIPT_TO_GENERATE](#)

NIFGEN_ATTR_DAQMX_TASK

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

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

NIFGEN_ATTR_EXTERNAL_CLOCK_DELAY_BIN

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViInt32	R/W	N/A	None	None

Description

Binary value of the external clock delay.

NIFGEN_ATTR_OSCILLATOR_PHASE_DAC_VAL

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the value of the oscillator phase DAC.

NIFGEN_ATTR_CAL_ADC_INPUT

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViInt32	R/W	N/A	None	None

Description

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.

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.

NIFGEN_ATTR_FLATNESS_CORRECTION_ENAB

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enables flatness correction. When flatness correction is enabled, 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 attribute should be set to VI_FALSE when performing Flatness Calibration.

Defined Values

VI_TRUE	Enables flatness correction.
VI_FALSE	Disables flatness correction.

NIFGEN_ATTR_GAIN_DAC_VALUE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_OFFSET_DAC_VALUE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_OSCILLATOR_FREQ_DAC_VALU

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViInt32	R/W	N/A	None	None

Description

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

NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of post-amplifier attenuation that should be applied to the signal.

Units: decibels

NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of pre-amplifier attenuation that should be applied to the signal.

Units: decibels

NIFGEN_ATTR_ACTUAL_ARB_SAMPLE_RATE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	RO	N/A	None	None

Description

The actual sample rate value of the signal generator.

NIFGEN_ATTR_SAMPLE_CLOCK_ABSOLUTE_DELAY

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the absolute delay adjustment of the sample clock. The sample clock delay adjustment is expressed in seconds.



Note For the NI 5421, absolute delay can only be applied when an external sample clock is used.

NIFGEN_ATTR_CLOCK_MODE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Controls which clock mode is used for the signal generator.

For signal generators that support it, this attribute allows switching the sample clock to High-Resolution mode. When in Divide-Down 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.

Defined Values

NIFGEN_VAL_DIVIDE_DOWN	Divide down sampling —Sample rates are generated by dividing the source frequency.
NIFGEN_VAL_HIGH_RESOLUTION	High resolution sampling —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 clocking modes.

Default Value: Depends on the device

NIFGEN_ATTR_EXPORTED_ONBOARD_REF_CLK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_REF_CLOCK_FREQUENCY

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the frequency of the reference clock that is used in the phase-locked 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 [niFgen Abort Generation](#) function or wait for the generation to complete.

NIFGEN_ATTR_EXPORTED_REF_CLOCK_OUTPI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_REFERENCE_CLOCK_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the reference clock source used by the signal generator. For example, when you set this attribute to "ClkIn", the signal generator uses the signal it receives at the Clk In front panel connector as its 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 attribute. To change the device configuration, call the [niFgen_AbortGeneration](#) function 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
"PXI_Clk10"	PXI Clock 10 line
"ClkIn"	Front panel connector
"OnboardRefClk"	Onboard reference clock
"RTSI7"	RTSI line 7
"RefIn"	Ref In

Default Value: "None"

NIFGEN_ATTR_EXPORTED_SAMPLE_CLOCK_TI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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 attribute. To change the device configuration, call the [niFgen_AbortGeneration](#) function or wait for the generation to complete.

NIFGEN_ATTR_EXPORTED_SAMPLE_CLOCK_O

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_SAMPLE_CLOCK_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the Sample Clock source.



Note The signal generator must not be in the Generating state when you change this attribute. To change the device configuration, call the [niFgen_AbortGeneration](#) function 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"	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"

NIFGEN_ATTR_FREQ_LIST_DURATION_QUANTL

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Returns the quantum of which all durations must be a multiple in a frequency list.

NIFGEN_ATTR_FREQ_LIST_HANDLE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Sets which frequency list the signal generator produces. Create a frequency list using the [niFgen_CreateFreqList](#) function. The niFgen_CreateFreqList function returns a handle that you can use to identify the list.

NIFGEN_ATTR_MAX_FREQ_LIST_DURATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	RO	N/A	None	None

Description

The maximum duration of any one step in the frequency list.

NIFGEN_ATTR_MAX_FREQ_LIST_LENGTH

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

The maximum number of steps that can be in a frequency list.

NIFGEN_ATTR_MAX_NUM_FREQ_LISTS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

The maximum number of frequency lists the signal generator allows.

NIFGEN_ATTR_MIN_FREQ_LIST_DURATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	RO	N/A	None	None

Description

The minimum number of steps that can be in a frequency list.

NIFGEN_ATTR_MIN_FREQ_LIST_LENGTH

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

The minimum number of frequency lists that the signal generator allows.

NIFGEN_ATTR_LOGICAL_NAME

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Contains the logical name that you specified when opening the current IVI session.

You may pass a logical name to the [niFgen_init](#) function or the [niFgen_InitWithOptions](#) function. 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.

NIFGEN_ATTR_RESOURCE_DESCRIPTOR

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Indicates the resource descriptor that NI-FGEN uses to identify the physical device.

If you initialize NI-FGEN with a logical name, this attribute contains the resource descriptor that corresponds to the entry in the IVI Configuration Utility.

If you initialize NI-FGEN with the resource descriptor, this attribute contains that value.

NIFGEN_ATTR_CHANNEL_COUNT

IviFgenBase Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

Indicates the number of channels that the specific instrument driver supports.

For each attribute for which IVI_VAL_MULTI_CHANNEL is set, the IVI Engine maintains a separate cache value for each channel.

NIFGEN_ATTR_GROUP_CAPABILITIES

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

A string that contains a comma-separated list of class-extension groups that NI-FGEN implements.

NIFGEN_ATTR_SUPPORTED_INSTRUMENT_MOI

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Contains a model code of the device. For NI-FGEN versions that support more than one device, this attribute contains a comma-separated list of supported device models.

NIFGEN_ATTR_SPECIFIC_DRIVER_CLASS_SPEC

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

Returns the major version number of the class specification with which NI-FGEN is compliant.

NIFGEN_ATTR_SPECIFIC_DRIVER_CLASS_SPEC

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

Returns the minor version number of the class specification with which NI-FGEN is compliant.

NIFGEN_ATTR_SPECIFIC_DRIVER_DESCRIPTION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains a brief description of NI-FGEN.

NIFGEN_ATTR_SPECIFIC_DRIVER_VENDOR

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains the name of the vendor that supplies NI-FGEN.

NIFGEN_ATTR_SPECIFIC_DRIVER_REVISION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

Returns a string that contains additional version information about NI-FGEN.

NIFGEN_ATTR_INSTRUMENT_FIRMWARE_REVIS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the firmware revision information for the device that you are currently using.

NIFGEN_ATTR_INSTRUMENT_MANUFACTURER

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the name of the device manufacturer you are currently using.

NIFGEN_ATTR_INSTRUMENT_MODEL

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the model number or name of the device that you are currently using.

NIFGEN_ATTR_CACHE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to cache the value of attributes. When caching is enabled, NI-FGEN keeps track of the current device settings and avoids sending redundant commands to the device. Thus, you can significantly increase execution speed.

NI-FGEN can choose to always cache or to never cache particular attributes regardless of the setting of this attribute. Use the [niFgen_InitWithOptions](#) function to override the default value.

Default Value: VI_TRUE

NIFGEN_ATTR_INTERCHANGE_CHECK

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to perform interchangeability checking and retrieve interchangeability warnings when you call the [niFgen_InitiateGeneration](#) function.

Interchangeability warnings indicate that using your application with a different device might cause different behavior. Call the [niFgen_GetNextInterchangeWarning](#) function to extract interchange warnings. Call the [niFgen_ClearInterchangeWarnings](#) function to clear the list of interchangeability warnings without reading them.

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 device and you have not set that attribute, or the attribute has been invalidated since you set it.

Default Value: VI_FALSE

NIFGEN_ATTR_QUERY_INSTRUMENT_STATUS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether NI-FGEN queries the device status after each operation. Querying the device status is very useful for debugging. After you validate your program, you can set this attribute to VI_FALSE to disable status checking and maximize performance.

NI-FGEN can choose to ignore status checking for particular attributes regardless of the setting of this attribute.

Default Value: VI_TRUE

Use the [niFgen_InitWithOptions](#) function to override the default value.

NIFGEN_ATTR_RANGE_CHECK

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to validate attribute values and function parameters. If enabled, NI-FGEN validates the parameter values that you pass to the functions. Range-checking parameters is very useful for debugging. After you validate your program, you can set this attribute to VI_FALSE to disable range checking and maximize performance.

Default Value: VI_TRUE

Use the [niFgen_InitWithOptions](#) function to override the default value.

NIFGEN_ATTR_RECORD_COERCIONS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether the IVI Engine keeps a list of the value coercions it makes for ViInt32 and ViReal64 attributes. Call the [niFgen_GetNextCoercionRecord](#) function to extract and delete the oldest coercion record from the list.

Use the [niFgen_InitWithOptions](#) function to override default value.

Default Value: VI_FALSE

NIFGEN_ATTR_SIMULATE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to simulate NI-FGEN I/O operations. If simulation is enabled, NI-FGEN functions perform range checking and call Get Attribute and Set Attribute functions, but they do not perform device I/O. For output parameters that represent device data, NI-FGEN functions return calculated values.

Use the [niFgen_InitWithOptions](#) function to override default value.

Default Value: VI_FALSE

NIFGEN_ATTR_BUS_TYPE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

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

NIFGEN_ATTR_MEMORY_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

The total amount of memory in bytes on the signal generator.

NIFGEN_ATTR_SERIAL_NUMBER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns the signal generator's serial number.

NIFGEN_ATTR_VIDEO_WAVEFORM_TYPE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Selects which waveform type that the NI 5431 generates. Setting this attribute ensures that the crystal is set to the proper frequency.

NIFGEN_ATTR_SYNC_OUT_OUTPUT_TERMINAL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the terminal to export the Sync Out signal. For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX. This attribute is not supported for all devices.

NIFGEN_ATTR_REF_CLOCK_SOURCE

IviFgenBase Attribute

Data Type Access Applies to Coercion High-Level Functions

ViString R/W N/A None [niFgen_ConfigureRefClockSour](#)

Description

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 attribute. To change the device configuration, call the [niFgen Abort Generation](#) function 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
"RefIn"	External signal on the Ref In front panel connector

Default Value: "None"

NIFGEN_ATTR_CYCLE_COUNT

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

The number of cycles that you want the signal generator to produce after it receives a trigger.

For standard and arbitrary waveforms, a cycle is one period of the waveform.

An arbitrary sequence consists of multiple arbitrary waveforms in a sequence. Each waveform can be repeated a discrete number of times before the next waveform is produced. For arbitrary sequences, a cycle is one complete progression through the generation of all iterations of all waveforms in the sequence.



Note The NI 5411/5421 arbitrary waveform generators only support continuous generation.

Defined Value

NIFGEN_VAL_GENERATE_CONTINUOUS

NIFGEN_ATTR_ERROR_ELABORATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViString	R/W	N/A	None	None

Description

An optional string that contains additional information concerning the primary error condition.

NIFGEN_ATTR_IO_SESSION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViSession	RO	N/A	None	None

Description

Specifies the I/O session that NI-FGEN uses to communicate with the instrument.

NIFGEN_ATTR_NUM_CHANNELS

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	RO	N/A	None	None

Description

Indicates the number of channels that the specific instrument driver supports.

For each attribute for which IVI_VAL_MULTI_CHANNEL is set, the IVI Engine maintains a separate cache value for each channel.

NIFGEN_ATTR_PRIMARY_ERROR

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Provides a code that describes the first error that occurred since the last call to the [niFgen_GetError](#) function on the session. The value follows the VXI *plug&play* completion code conventions. A negative value 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.

NIFGEN_ATTR_UPDATE_CLOCK_SOURCE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the update clock source.

Defined Values

NIFGEN_VAL_INTERNAL	Internal clock source
NIFGEN_VAL_EXTERNAL	External clock source
NIFGEN_VAL_CLK_IN	CLK IN front panel connector
NIFGEN_VAL_DDC_CLK_IN	Digital Data & Control clock input
NIFGEN_VAL_PXI_STAR	PXI Star
NIFGEN_VAL_RTSI_0	RTSI 0 or PXI_Trig 0
NIFGEN_VAL_RTSI_1	RTSI 1 or PXI_Trig 1
NIFGEN_VAL_RTSI_2	RTSI 2 or PXI_Trig 2
NIFGEN_VAL_RTSI_3	RTSI 3 or PXI_Trig 3
NIFGEN_VAL_RTSI_4	RTSI 4 or PXI_Trig 4
NIFGEN_VAL_RTSI_5	RTSI 5 or PXI_Trig 5
NIFGEN_VAL_RTSI_6	RTSI 6 or PXI_Trig 6
NIFGEN_VAL_RTSI_7	RTSI 7 or PXI_Trig 7

Default Value: NIFGEN_VAL_INTERNAL

NIFGEN_ATTR_SECONDARY_ERROR

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_ANALOG_FILTER_ENABLED

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Controls whether the signal generator applies an analog filter to the output signal. This attribute is valid in arbitrary waveform, arbitrary sequence, and script modes. This attribute can also be used in standard function and frequency list modes for user-defined waveforms.

Defined Values

VI_TRUE	Applies an analog filter.
VI_FALSE	Does not apply an analog filter.

NIFGEN_ATTR_ANALOG_PATH

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the analog signal path that should be used. The main path allows you to configure gain, offset, analog filter status, output impedance, and output enable. The main path has two amplifier options, high-gain and low-gain.

The direct path presents a much smaller gain range, and you cannot adjust offset or the filter status. The direct path also provides a smaller output range but also lower distortion. NI-FGEN normally chooses the amplifier based on the user-specified gain.

Defined Values

NIFGEN_VAL_MAIN_ANALOG_PATH	Sets the Main path
NIFGEN_VAL_DIRECT_ANALOG_PATH	Sets the Direct path
NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	Sets the Main path with low-gain amplifier
NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	Sets the Main path with high-gain amplifier

NIFGEN_ATTR_DATAMARKEREVENTS_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

Specifies the number of Data Marker events supported by the device.

NIFGEN_ATTR_ANALOG_DATA_MASK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the mask to apply to the analog output. The masked data is replaced with the data in the [NIFGEN_ATTR_ANALOG_STATIC_VALUE](#) attribute.

NIFGEN_ATTR_ANALOG_STATIC_VALUE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the static value that replaces data masked by the [NIFGEN_ATTR_ANALOG_DATA_MASK](#) attribute.

NIFGEN_ATTR_DIGITAL_DATA_MASK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

This attribute contains the mask to apply to the output on the digital connector. The masked data is replaced with the data in the [NIFGEN_ATTR_DIGITAL_STATIC_VALUE](#) attribute.

NIFGEN_ATTR_DIGITAL_STATIC_VALUE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the static value that replaces data masked by the [NIFGEN_ATTR_DIGITAL_DATA_MASK](#) attribute.

NIFGEN_ATTR_DIGITAL_FILTER_ENABLED

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Controls whether the signal generator applies a digital filter to the output signal. This attribute is valid in Arbitrary Waveform, Arbitrary Sequence, and Script modes. This attribute can also be used in Standard Function and Frequency List modes for user-defined waveforms.

NIFGEN_ATTR_DIGITAL_FILTER_INTERPOLATIO

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

This attribute only affects the device when the [NIFGEN_ATTR_DIGITAL_FILTER_ENABLED](#) attribute is set to VI_TRUE. If you do not set this attribute directly, NI-FGEN automatically selects the maximum interpolation factor allowed for the current sample rate. Legal values are 2, 4, and 8.

NIFGEN_ATTR_DIGITAL_GAIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the digital gain to apply to the processed data.

Default Value: 1.0

NIFGEN_ATTR_DIGITAL_PATTERN_ENABLED

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViBoolean	R/W	N/A	None	None

Description

Controls whether the signal generator generates a digital pattern of the output signal.

NIFGEN_ATTR_FILTER_CORRECTION_FREQUENCY

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Controls the filter correction frequency of the analog filter. This attribute corrects 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 attribute to 0 Hz.

Units: Hz

Default Value: 0

NIFGEN_ATTR_LOAD_IMPEDANCE

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

This channel-based attribute specifies the load impedance connected to the analog output of the channel. If you set this attribute to NI-FGEN_VAL_MATCHED_LOAD_IMPEDANCE (-1.0), NI-FGEN assumes that the load impedance matches the output impedance. NI-FGEN compensates to give the desired peak-to-peak voltage amplitude or arbitrary gain (relative to 1 V).

NIFGEN_ATTR_SYNC_DUTY_CYCLE_HIGH

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	N/A	None	None

Description

Controls the duty cycle of the square wave the signal generator produces on the SYNC out line. Specify this attribute 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%

NIFGEN_ATTR_DATA_MARKER_EVENT_DATA_E

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the bit number to assign to the Data Marker event.

NIFGEN_ATTR_DATA_MARKER_EVENT_LEVEL_

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Data Marker event. Refer to [Data Marker Events](#) topic for more information on Data Marker event polarity.

Defined Values

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Data Marker event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Data Marker event level is high.

NIFGEN_ATTR_DATA_MARKER_EVENT_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination terminal for the Data Marker event. For a list of the destination terminals available on your device, refer to the **Device Routes** tab in MAX.

NIFGEN_ATTR_DONE_EVENT_LATCHED_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Returns the latched status of the specified Done event.

NIFGEN_ATTR_DONE_EVENT_DELAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of delay applied to a Done Event with respect to the analog output of the signal generator. A positive delay value indicates that the Done event follows the analog data, while a negative delay value indicates that the Done event precedes the analog data. The default value is zero, which aligns the Done event with the analog output signal.

You can specify the units of the delay value by setting the [NIFGEN_ATTR_DONE_EVENT_DELAY](#) attribute.

NIFGEN_ATTR_DONE_EVENT_DELAY_UNITS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the units applied to the value of the [NIFGEN_ATTR_DONE_EVENT_DELAY](#) attribute.

Defined Values

NIFGEN_VAL_SECONDS	Delay is specified in seconds and then coerced up by NI-FGEN to the nearest Sample clock period.
NIFGEN_VAL_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

NIFGEN_VAL_SECONDS

NIFGEN_ATTR_DONE_EVENT_LEVEL_ACTIVE_L

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Done event.

Defined Values:

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Done event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Done event level is high.

NIFGEN_ATTR_DONE_EVENT_OUTPUT_BEHAVI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output behavior for the Done event.

Defined Values

NIFGEN_VAL_PULSE	Triggers a pulse for a specified period of time.
NIFGEN_VAL_LEVEL	Shifts high or low while the event is active, depending on the active state you specify.

NIFGEN_ATTR_DONE_EVENT_OUTPUT_TERMIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination terminal for the Done event. For a list of the destination terminals available on your device, refer to the **Device Routes** tab in MAX.

NIFGEN_ATTR_DONE_EVENT_PULSE_POLARITY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Done event.

Defined Values

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Done event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Done event level is high.

NIFGEN_ATTR_DONE_EVENT_PULSE_WIDTH_U

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the pulse width units for the Done event.

Defined Values

NIFGEN_VAL_SAMPLE_CLOCK_PERIODS	Specifies the pulse width in Sample clock periods.
NIFGEN_VAL_SECONDS	Specifies the pulse width in seconds.

NIFGEN_ATTR_DONE_EVENT_PULSE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the pulse width for the Done event.

NIFGEN_ATTR_EXPORTED_SAMPLE_CLOCK_DIR

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the factor by which to divide the update (sample) clock before it is exported. To export the sample clock, use the [niFgen_RouteSignalOut](#) function.

NIFGEN_ATTR_EXPORTED_SAMPLE_CLOCK_TI

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViInt32	R/W	N/A	None	None

Description

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 [niFgen Export Signal](#) function.

Default Value: 2

Values: Range is from 2 to 4,194,304.

NIFGEN_ATTR_ALL_MARKER_EVENTS_LATCH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Returns a bit field of the latched status of all Marker events.

Write 0 to this attribute to clear the latched status of all Marker events.

Defined Values:

Marker 0	0x0
Marker 1	0x1
Marker 2	0x2
Marker 3	0x4

NIFGEN_ATTR_ALL_MARKER_EVENTS_LIVE_S1

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

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

NIFGEN_ATTR_MARKER_EVENT_LATCHED_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies the latched status of the specified Marker event.

NIFGEN_ATTR_MARKER_EVENT_LIVE_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Returns the live status of the specified Marker event.

NIFGEN_ATTR_MARKER_EVENT_DELAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of delay applied to a Marker event with respect to the analog output of the signal generator. A positive delay value indicates that the Marker event follows the analog data, while a negative delay value indicates that the Marker event will precede the analog data. The default value is zero, which aligns the Marker event with the analog output signal.

You can specify the units of the delay value by setting the [NIFGEN_ATTR_MARKER_EVENT_DELAY](#) attribute.

NIFGEN_ATTR_MARKER_EVENT_DELAY_UNITS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the units applied to the value of the [NIFGEN_ATTR_MARKER_EVENT_DELAY](#) attribute.

Defined Values

NIFGEN_VAL_SECONDS	Delay is specified in seconds and then coerced up by NI-FGEN to the nearest Sample clock period.
NIFGEN_VAL_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

NIFGEN_VAL_SECONDS

NIFGEN_ATTR_MARKER_EVENT_OUTPUT_BEH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output behavior for the Marker event.

Defined Values

NIFGEN_VAL_PULSE	Triggers a pulse for a specified period of time.
NIFGEN_VAL_LEVEL	Shifts high or low while the event is active, depending on the active state you specify.

NIFGEN_ATTR_MARKER_EVENT_OUTPUT_TERI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination terminal for the Marker event.

For a list of the terminals available on your device, refer to the **Device Routes** tab in MAX.

NIFGEN_ATTR_MARKER_EVENT_PULSE_POLAI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Marker event.

Defined Values

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Marker event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Marker event level is high.

NIFGEN_ATTR_MARKER_EVENT_PULSE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the pulse width units for the Marker event.

Defined Values

NIFGEN_VAL_SAMPLE_CLOCK_PERIODS	Specifies the pulse width in Sample clock periods.
NIFGEN_VAL_SECONDS	Specifies the pulse width in seconds.

NIFGEN_ATTR_MARKER_EVENT_PULSE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the pulse width for the Marker event.

NIFGEN_ATTR_MARKER_EVENT_TOGGLE_INITI

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Toggles the Initial state of the Marker event.

NIFGEN_ATTR_READY_FOR_START_EVENT_LIV

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Returns the live status of the specified Ready For Start event.

NIFGEN_ATTR_READY_FOR_START_EVENT_LE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Ready for Start event.

Defined Values:

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Ready for Start event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Ready for Start event is high.

NIFGEN_ATTR_READY_FOR_START_EVENT_OUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_STARTED_EVENT_LATCHED_ST

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Specifies the latched status of the Started event.

NIFGEN_ATTR_STARTED_EVENT_DELAY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of delay applied to a Started event with respect to the analog output of the signal generator. A positive delay value indicates that the Started event precedes the analog data, while a negative delay value indicates that the Started event follows the analog data. The default value is zero, which aligns the Started event with the analog output.

You can specify the units of the delay value by setting the [NIFGEN_ATTR_STARTED_EVENT_DELAY](#) attribute.

NIFGEN_ATTR_STARTED_EVENT_DELAY_UNITS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the units applied to the value of the [NIFGEN_ATTR_STARTED_EVENT_DELAY](#) attribute.

Defined Values

NIFGEN_VAL_SECONDS	Delay is specified in seconds and then coerced up by NI-FGEN to the nearest Sample clock period.
NIFGEN_VAL_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

NIFGEN_VAL_SECONDS

NIFGEN_ATTR_STARTED_EVENT_LEVEL_ACTIV

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Started event.

Defined Values:

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Started event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Started event level is high.

NIFGEN_ATTR_STARTED_EVENT_OUTPUT_BEH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output behavior for the Started event.

Defined Values

NIFGEN_VAL_PULSE	Triggers a pulse for a specified period of time.
NIFGEN_VAL_LEVEL	Shifts high or low while the event is active, depending on the active state you specify.

NIFGEN_ATTR_STARTED_EVENT_OUTPUT_TER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_STARTED_EVENT_PULSE_POLA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the output polarity of the Started event.

Defined Values

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Started event level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Started event level is high.

NIFGEN_ATTR_STARTED_EVENT_PULSE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the pulse width units for the Started event.

Defined Values:

NIFGEN_VAL_SAMPLE_CLOCK_PERIODS	Specifies the pulse width in Sample clock periods.
NIFGEN_VAL_SECONDS	Specifies the pulse width in seconds.

NIFGEN_ATTR_STARTED_EVENT_PULSE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the pulse width for the Started event.

NIFGEN_ATTR_MARKEREVENTS_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

Returns the number of markers supported by the device. Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_SCRIPT.

NIFGEN_ATTR_SCRIPT_TO_GENERATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Yes	None	None

Description

This attribute identifies which script the generator produces. You can create multiple scripts using the [niFgen_WriteScript](#) function. To configure the generator to run a particular script, set this attribute to the name of the script.

Use this property only when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_SCRIPT.



Note You cannot change this attribute while the device is generating a waveform. If you want to change the device configuration, call the [niFgen_AbortGeneration](#) function or wait for the generation to complete.

NIFGEN_ATTR_SCRIPTTRIGGERS_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	RO	N/A	None	None

Description

Specifies the number of script triggers supported by the device. Use this attribute when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_SCRIPT.

NIFGEN_ATTR_STREAMING_WAVEFORM_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the name of the waveform used to continuously stream data during generation. This attribute defaults to "" (empty string) when no streaming waveform is specified.

Use in conjunction with the

[NIFGEN_ATTR_STREAMING_SPACE_AVAILABLE_IN_WAVEFORM](#)
attribute.

NIFGEN_ATTR_FUNC_AMPLITUDE

IviFgenStdFunc Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureStandardWave](#)

Description

Controls the amplitude of the standard waveform that the signal generator produces. 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 the amplitude to 10.00 V.



Note This parameter does not affect signal generator behavior when you set the **waveform** parameter of the [niFgen_ConfigureStandardWaveform](#) function to NIFGEN_VAL_WFM_DC.

Units: V_{pk-pk}

Default Value: None

NIFGEN_ATTR_FUNC_BUFFER_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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.



Note Refer to the [Standard Function Mode](#) topic for more information on the implementation of Standard Function Mode on your device.

NIFGEN_ATTR_FUNC_DC_OFFSET

IviFgenStdFunc Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureStandardWave](#)

Description

Controls the DC offset of the standard waveform that the signal generator produces. This value is the offset at the output terminal. The value is the offset from ground to the center of the waveform that you specify with the **waveform** parameter of the [niFgen_ConfigureStandardWaveform](#) function.

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 to 5.00 V.

Units: volts

Default Value: None

NIFGEN_ATTR_FUNC_DUTY_CYCLE_HIGH

IviFgenStdFunc Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
ViReal64	R/W	Channel	None	None

Description

Controls the duty cycle of the square wave the signal generator produces. Specify this attribute as a percentage of the time the square wave is high in a cycle.



Note This parameter only affects signal generator behavior when you set the **waveform** parameter of the [niGen_ConfigureStandardWaveform](#) function to NIFGEN_VAL_WFM_SQUARE.

Units: Percentage of time the waveform is high

Default Value: 50%

NIFGEN_ATTR_FUNC_FREQUENCY

IviFgenStdFunc Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureStandardWave](#)

Description

Controls the frequency of the standard waveform that the signal generator produces.



Notes This parameter does not affect signal generator behavior when you set the **waveform** parameter of the [niFgen_ConfigureStandardWaveform](#) function to NIFGEN_VAL_WFM_DC.

For NIFGEN_VAL_WFM_SINE, the range is between 0 MHz and 16 MHz, but is between 0 MHz and 1 MHz for all other waveforms.

Units: hertz

Default Value: None

NIFGEN_ATTR_FUNC_MAX_BUFFER_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

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 read-only for all other devices.



Note Refer to the [Standard Function Mode](#) topic for more information on the implementation of Standard Function Mode on your device.

NIFGEN_ATTR_FUNC_START_PHASE

IviFgenStdFunc Attribute

Data Type Access Applies to Coercion High-Level Functions

ViReal64 R/W Channel None [niFgen_ConfigureStandardWave](#)

Description

Controls horizontal offset of the standard waveform the signal generator produces. Specify this attribute 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.



Note This parameter does not affect signal generator behavior when you set the **waveform** parameter of the [niFgen_ConfigureStandardWaveform](#) function to NIFGEN_VAL_WFM_DC.

Units: Degrees of one cycle

Default Value: None

NIFGEN_ATTR_FUNC_WAVEFORM

IviFgenStdFunc Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W Channel None [niFgen_ConfigureStandardWave](#)

Description

This channel-based attribute specifies which standard waveform the signal generator produces.

Use this attribute only when the [NIFGEN_ATTR_OUTPUT_MODE](#) attribute is set to NIFGEN_VAL_OUTPUT_FUNC.

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 as defined with the niFgen_DefineUserStandardWaveform function

Default Value: NIFGEN_VAL_WFM_SINE

NIFGEN_ATTR_DIGITAL_EDGE_SCRIPT_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the active edge for the Script trigger. This attribute is used when the [NIFGEN_ATTR_SCRIPT_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_EDGE.

Defined Values

NIFGEN_VAL_RISING_EDGE	Occurs when the signal transitions from low level to high level.
NIFGEN_VAL_FALLING_EDGE	Occurs when the signal transitions from high level to low level.

NIFGEN_ATTR_DIGITAL_EDGE_SCRIPT_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source terminal for the Script trigger. This attribute is used when the [NIFGEN_ATTR_SCRIPT_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_EDGE.

NIFGEN_ATTR_DIGITAL_LEVEL_SCRIPT_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source terminal for the Script trigger. This attribute is used when the [NIFGEN_ATTR_SCRIPT_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_LEVEL.

NIFGEN_ATTR_DIGITAL_LEVEL_SCRIPT_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the active level for the Script trigger. This attribute is used when the [NIFGEN_ATTR_SCRIPT_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_LEVEL.

Defined Values

NIFGEN_VAL_ACTIVE_HIGH	When the signal level is high, the Script trigger level is high.
NIFGEN_VAL_ACTIVE_LOW	When the signal level is low, the Script trigger level is high.

NIFGEN_ATTR_EXPORTED_SCRIPT_TRIGGER_C

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the output terminal for the exported Script trigger. Setting this attribute 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.

NIFGEN_ATTR_SCRIPT_TRIGGER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the Script Trigger type. Depending upon the value of this attribute, additional attributes may need to be configured to fully configure the trigger.

Defined Values:

NIFGEN_VAL_TRIG_NONE	No trigger is configured. Signal generation starts immediately.
NIFGEN_VAL_DIGITAL_EDGE	Operation begins when a digital edge is detected.
NIFGEN_VAL_DIGITAL_LEVEL	Operation begins when a digital level is detected.
NIFGEN_VAL_SOFTWARE_EDGE	Operation begins when a software edge is detected.

NIFGEN_ATTR_DIGITAL_EDGE_START_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the active edge for the Start trigger. This attribute is used only when the [NIFGEN_ATTR_START_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_EDGE.

Defined Values

NIFGEN_VAL_RISING_EDGE	Occurs when the signal transitions from low level to high level.
NIFGEN_VAL_FALLING_EDGE	Occurs when the signal transitions from high level to low level.

NIFGEN_ATTR_DIGITAL_EDGE_START_TRIGGER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source terminal for the Start trigger. This attribute is used only when the [NIFGEN_ATTR_START_TRIGGER_TYPE](#) attribute is set to NIFGEN_VAL_DIGITAL_EDGE.

NIFGEN_ATTR_EXPORTED_START_TRIGGER_O

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

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.

NIFGEN_ATTR_START_TRIGGER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies whether you want the Start trigger to be a Digital Edge or Software trigger. You can also choose NIFGEN_VAL_TRIG_NONE as the value for this attribute.

Defined Values:

NIFGEN_VAL_TRIG_NONE	No trigger is configured. Signal generation starts immediately.
NIFGEN_VAL_DIGITAL_EDGE	Operation begins when a digital edge is detected.
NIFGEN_VAL_SOFTWARE_EDGE	Operation begins when a software edge is detected.

NIFGEN_ATTR_SYNCHRONIZATION

Specific Attribute

Data Type	Access	Applies to	Coercion	High-Level Functions
VlInt32	R/W	N/A	None	None

Description

Specifies the source of the synchronization signal that you want to use.

Defined Values

NIFGEN_VAL_NONE	No synchronization source
NIFGEN_VAL_RTSI_0	RTSI 0 or PXI_Trig 0
NIFGEN_VAL_RTSI_1	RTSI 1 or PXI_Trig 1
NIFGEN_VAL_RTSI_2	RTSI 2 or PXI_Trig 2
NIFGEN_VAL_RTSI_3	RTSI 3 or PXI_Trig 3
NIFGEN_VAL_RTSI_4	RTSI 4 or PXI_Trig 4
NIFGEN_VAL_RTSI_5	RTSI 5 or PXI_Trig 5
NIFGEN_VAL_RTSI_6	RTSI 6 or PXI_Trig 6
NIFGEN_VAL_RTSI_7	RTSI 7 or PXI_Trig 7

NIFGEN_ATTR_TRIGGER_MODE

Specific Attribute

Data Type Access Applies to Coercion High-Level Functions

ViInt32 R/W Channel None [niFgen_ConfigureTriggerMode](#)

Description

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.

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

Default Value: NIFGEN_VAL_CONTINUOUS

NIFGEN_ATTR_TRIGGER_SOURCE

IviFgenTrigger Attribute

Data Type Access Applies to Coercion High-Level Functions

VlInt32 R/W Channel None [niFgen_ConfigureTriggerSource](#)

Description

Controls which trigger source the signal generator uses.

After you call the [niFgen_InitiateGeneration](#) function, the signal generator waits for the trigger that you specify in the **triggerSource** parameter in the niFgen_ConfigureTriggerSource function. After the signal generator receives a trigger, it produces the number of cycles that you specify in the [NIFGEN_ATTR_CYCLE_COUNT](#) function.

This attribute is also the source for the trigger in the other trigger modes as specified by the [NIFGEN_ATTR_TRIGGER_MODE](#) attribute.

Defined Values

NIFGEN_VAL_IMMEDIATE	Immediate—The signal generator does not wait for a trigger.
NIFGEN_VAL_EXTERNAL	External—The signal generator waits for a trigger on the external trigger input.
NIFGEN_VAL_PFI_0	PFI 0
NIFGEN_VAL_PFI_1	PFI 1
NIFGEN_VAL_PFI_2	PFI 2
NIFGEN_VAL_PFI_3	PFI 3
NIFGEN_VAL_SOFTWARE_TRIG	Software Trigger—The signal generator waits until you call the niFgen_SendSoftwareEdgeTrigger function.
NIFGEN_VAL_RTSI_0	RTSI 0 or PXI_Trig 0
NIFGEN_VAL_RTSI_1	RTSI 1 or PXI_Trig 1
NIFGEN_VAL_RTSI_2	RTSI 2 or PXI_Trig 2
NIFGEN_VAL_RTSI_3	RTSI 3 or PXI_Trig 3
NIFGEN_VAL_RTSI_4	RTSI 4 or PXI_Trig 4
NIFGEN_VAL_RTSI_5	RTSI 5 or PXI_Trig 5
NIFGEN_VAL_RTSI_6	RTSI 6 or PXI_Trig 6
NIFGEN_VAL_RTSI_7	RTSI 7 or PXI_Trig 7

NIFGEN_VAL_PXI_STAR

PXI Star

Default Value: NIFGEN_VAL_IMMEDIATE

NI-FGEN Examples for LabWindows/CVI

NI-FGEN ships with several examples for use with LabWindows/CVI. 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

If you are using LabWindows/CVI 7.0 or later, you can locate these examples 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 [NI-FGEN Instrument Driver Readme](#).

The NI Example Finder allows you to browse and search installed examples and examples on the [NI Developer Zone](#).

Using the Examples

To use an example in LabWindows/CVI, open the project file for the example you want to run. The source code for these examples is documented, as well as all of the input and output values, to make changing the code to perform different types of generation easier.

Help for Examples

For more information about individual functions and attributes, select **Instrument»National Instruments Function Generator** to launch the NI-FGEN function panel.



Note If NI-FGEN does not appear in the Instrument menu, select **Load** and navigate to the location of the example files specified in the [NI-FGEN Instrument Driver Readme](#). Select niFgen.fp to open.

NI-FGEN

Examples for Microsoft Visual C/C++ (C examples)

NI-FGEN ships with several examples for use with Microsoft Visual C/C++ (MSVC). These examples can help you develop software, and they illustrate how to perform the most common operations with the NI signal generators.

The MSVC examples were built and tested using Microsoft Visual C.

Note Use the NMAKE utility in MSVC to build the examples. From an MS-DOS session, navigate to the location of the example files specified in the [NI-FGEN Instrument Driver Readme](#).

Use the `msvc vcvars32.bat` to set up the variables necessary for building examples. To build an example, run `nmake /f examplename.mak`. The executable is built to the debug subdirectory.

The source code for these examples as well as all of the input and output values, is documented, making it easy to change the code to perform different types of generation.