CALIBRATION PROCEDURE

NI 5402/5406

This document contains instructions for writing a calibration procedure for the NI PCI/PXI-5402/5406 arbitrary function generators. This calibration procedure is intended for metrology labs.

Refer to `ni.com/calibration` for additional information about calibration solutions from National Instruments.

Contents

Conventions 2
Software Requirements 3
Documentation Requirements 4
Password 4
Calibration Interval 4
Test Equipment 5
Test Conditions 5
Calibration Procedures 6
- Initial Setup 6
- Self-Calibration 7
  - MAX 7
  - FGEN Soft Front Panel 7
  - NI-FGEN 8
- Verification 9
  - Verifying the Oscillator Frequency Accuracy 11
  - Verifying the Sine Wave Flatness Correction Accuracy 14
  - Verifying the Main Path Gain and Offset Accuracy 21
  - Verifying Square Wave Gain and Offset Accuracy 30
- Updating the Calibration Date and Temperature 38
- Adjustment 39
  - Adjusting the Calibration ADC 41
  - Adjusting the Analog Output 54
  - Adjusting the Sine Wave Flatness Correction 81
  - Adjusting the Oscillator Frequency 90

Appendix A: Calibration Procedure Options..95
External Calibration...95
Complete Calibration..96
Optional Calibration...97
Appendix B: Calibration Utilities...99
MAX..99
FGEN SFP..99
NI-FGEN..100
Where to Go for Support...101

Conventions

The following conventions are used in this manual:

» — The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash. When this symbol is marked on a product, refer to the *Safety and Radio-Frequency Interference Read Me First* for information about precautions to take.

bold — Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic — Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace` — Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Platform — Text in this font denotes a specific platform and indicates that the text following it applies only to that platform.

Software Requirements

Calibrating the NI 5402/5406 requires installing NI-FGEN version 2.4 or later on the calibration system. You can download NI-FGEN from the Instrument Driver Network website at `ni.com/idnet`. NI-FGEN supports programming a *Self-Calibration* and an *External Calibration* in the LabVIEW, LabWindows™/CVI™, and C application development environments (ADEs). When you install NI-FGEN, you only need to install support for the ADE that you intend to use.

LabVIEW support is in the `niFgen.llb` file, and all calibration functions appear in the NI-FGEN calibration palette. For LabWindows/CVI, the NI-FGEN function panel (`niFgen.fp`) provides access to the available functions.

Calibration functions are LabVIEW VIs or C function calls in NI-FGEN. In this document, the LabVIEW VI or NI-FGEN LabVIEW Property Node is listed in the instructions. Tables for each step show both the configuration of the VI or Property Node and the C function. The C function calls are valid for any compiler capable of calling a 32-bit DLL. Many of the functions use constants defined in the `niFgen.h` file. To use these constants in C, include `niFgen.h` in your code when you write the calibration procedure.

For the locations of files you may need to calibrate your device, refer to the *NI-FGEN Instrument Driver Readme*, which is available on the NI-FGEN CD.

Note After you install NI-FGEN, you can access the *NI-FGEN Instrument Driver Readme* at **Start»All Programs»National Instruments»NI-FGEN»Documentation**.

Documentation Requirements

For information about NI-FGEN and the NI 5402/5406, you may find the following documents helpful:

- *NI Signal Generators Getting Started Guide*—provides instructions for installing and configuring NI signal generators.
- *NI Signal Generators Help*—includes detailed information about the NI 5402/5406 and the NI-FGEN VIs and functions.
- *NI 5402/5406 Specifications*—provides the published specification values for the NI 5402/5406

These documents are installed with NI-FGEN. You can also find the latest versions of the documentation at `ni.com/manuals`.

Password

The default password for password-protected operations on your device is "`NI`." This password is required to open an external calibration session.

Calibration Interval

A calibration is required only once every two years; however, the measurement accuracy demands of your application determine how often you should perform external calibration. For more information on designing a calibration procedure to suit your needs, refer to *Appendix A: Calibration Procedure Options*.

Test Equipment

External calibration requires different equipment for each applicable specification. Refer to Table 1 for information about the required equipment.

Table 1. Equipment Required for Calibrating the NI 5402/5406

Equipment	Recommended Models	Requirements
Digital multimeter (DMM)	NI PXI-4070/4071 Agilent/HP 34401A Keithley 2000	DC accuracy ≤ ±50 ppm Resolution ≤ 1 μV
Frequency Counter	Agilent/HP 53131A	Ability to measure 10 MHz or higher sine waves Frequency accuracy of ± 500 ppb
Power Meter	Rohde & Schwarz NRVS power meter with Rohde & Schwarz NRV-Z53 sensor Agilent E4419B Power Meter with Agilent E9304A power sensor	Power accuracy ± 0.10 dB For flatness measurements from 50 kHz to 50 MHz
Male banana to female BNC adapter	—	—
Male BNC to female SMB cable	—	50 Ω, RG-223

Test Conditions

Follow these guidelines to optimize the connections and the environment during calibration:

- Keep connections to the NI 5402/5406 short. Long cables and wires act as antennae, picking up noise that can affect measurements.
- Keep relative humidity between 10% and 90% non-condensing.
- Maintain a temperature between 18 °C and 28 °C.
- Allow a warm-up time of at least 15 minutes after NI-FGEN initiates a session to the board. The warm-up time ensures that the measurement circuitry of the NI 5402/5406 is at a stable operating temperature.

Note Unless manually disabled, NI-FGEN automatically loads with the operating system and enables the device.

- **(PXI)** Ensure that the PXI chassis fan speed is set to HI, that the filter fans are clean, and that the empty slots contain filler panels.
- **(PXI)** Plug the PXI chassis and the calibrator into the same power strip to avoid ground loops.

Calibration Procedures

The calibration process has the five following steps.

1. *Initial Setup*—Configure your device in Measurement & Automation Explorer (MAX).
2. *Self-Calibration*—Adjust the self-calibration constants of the device.
3. *Verification*—Verify the existing operation of the device. This step allows you to confirm that the device was operating within its specified range prior to calibration.
4. *Adjustment*—Adjust the device calibration constants with respect to a known voltage source and update the calibration date and temperature.
5. *Reverification*—Repeat the verification process to ensure that the device is operating within its specifications after adjustment.

These steps are described in detail in the following sections.

Note In some cases, the complete calibration procedure may not be required. Refer to *Appendix A: Calibration Procedure Options* for more information about developing a calibration procedure that fits your needs.

Initial Setup

The device must be configured in Measurement & Automation Explorer (MAX) to communicate with NI-FGEN.

Complete the following steps to configure a device in MAX.

1. Install the NI-FGEN driver software.
2. Power off the computer or chassis that will hold the device and install the device in an available slot.
3. Power on the computer or chassis and launch MAX.
4. Configure the device identifier and select Self-Test to ensure that the device is working properly.

Note When a device is configured with MAX, it is assigned a device identifier. This device identifier is used to open an NI-FGEN session.

Note For more information about configuring and testing your device in MAX, refer to the *NI Signal Generators Getting Started Guide*.

Self-Calibration

The NI 5402/5406 is capable of performing self-calibration, which adjusts the square waveform analog path and the gain and offset voltage of the main analog path. Self-calibration uses an onboard ADC to measure the output voltage.

Note Square wave gain and offset are calibrated through self-calibration only.

Note The analog output path, sine wave flatness correction, oscillator frequency, and the calibration ADC can be calibrated by using an adjustment procedure.

You can initiate self-calibration interactively from Measurement & Automation Explorer (MAX) or from the FGEN Soft Front Panel (SFP). Alternatively, you can initiate self-calibration programmatically using NI-FGEN. The following sections include information on performing a self-calibration within each of these environments.

MAX

To initiate self-calibration from MAX, complete the following steps:

1. Launch MAX.
2. Select **My System»Devices and Interfaces»NI-DAQmx Devices**.
3. Select the device that you want to calibrate.
4. Initiate self-calibration in one of the following ways:
   - Click **Self-Calibrate** in the upper right corner of the window.
   - Right-click the device name under Devices and Interfaces, and select **Self-Calibrate** from the drop-down menu.

FGEN Soft Front Panel

To initiate self-calibration from the FGEN SFP, complete the following steps:

1. Launch the FGEN Soft Front Panel.
2. Select **Edit»Device Configuration** to open the Device Configuration dialog box.

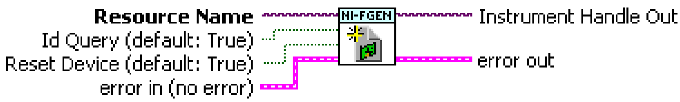
3. Select the device that you want to calibrate from the Device drop-down listbox. Click **Okay** when finished.
4. Select **Utility»Calibration** to open the Calibration dialog box.
5. Click **Perform self-calibration**.

NI-FGEN

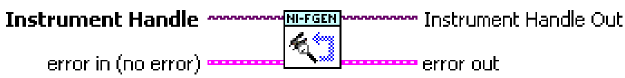
Complete the following steps to programatically perform a self-calibration on the NI 5402/5406 using NI-FGEN:

1. Open an instrument driver session, initialize the device for operation, and return a session handle that will be used to identify the device in future NI-FGEN calls by calling the niFgen Initialize VI.

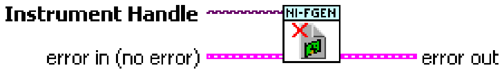
Note Throughout the procedure, refer to C/C++ function call parameters for the LabVIEW input values.

LabVIEW Block Diagram	C/C++ Function Call
 	Call `niFgen_init` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **IDQuery**: `VI_TRUE` **resetDevice**: `VI_TRUE`

2. Initiate self-calibration by calling the niFgen Self Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
 	Call `niFgen_SelfCal` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

3. Close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses by calling the niFgen Close VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_close` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

Verification

This section provides instructions for verifying the NI 5402/5406 specifications. Refer to Table 1 for recommendations on choosing an instrument to use for each test.

Note Always self-calibrate the NI 5402/5406 after warm-up and before beginning a verification procedure.

You can verify the following NI 5402/5406 specifications:

- Oscillator frequency accuracy
- Sine wave flatness accuracy
- Main path gain and offset accuracy
- Square wave gain and offset accuracy

Note If any of these tests fail immediately after you perform an external adjustment, verify that you have met the required test conditions (refer to the *Test Conditions* section) before you return the NI 5402/5406 to NI for repair.

Refer to Figure 1 for the names and locations of the NI 5402/5406 front panel connectors.

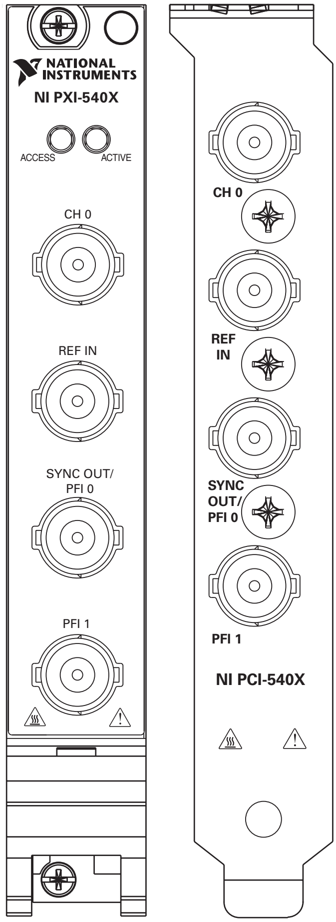
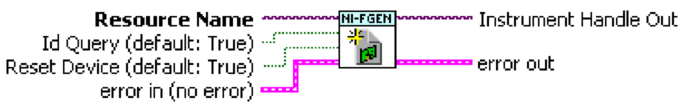


Figure 1. NI PXI-5402/5406 and NI PCI-5402/5406 Front Panel Connectors

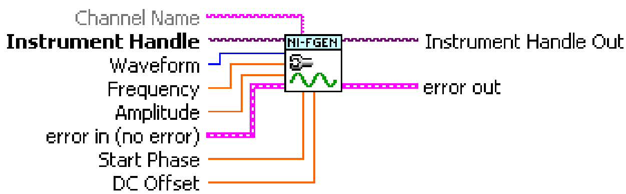
Verifying the Oscillator Frequency Accuracy

Complete this test to verify the frequency accuracy of the oscillator on the NI 5402/5406. In this test, you generate a 10 MHz sine wave with the NI 5402/5406 and measure the sine wave frequency with a frequency counter.

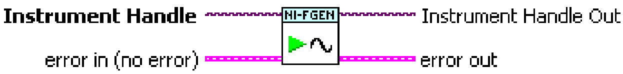
1. Connect the NI 5402/5406 CH 0 front panel connector to the frequency counter.
2. Open an instrument driver session, initialize the device for operation, and return a session handle that will be used to identify the device in future NI-FGEN calls by calling the niFgen Initialize VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_init` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **IDQuery**: `VI_TRUE` **resetDevice**: `VI_TRUE`

3. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ConfigureStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **waveform**: `NIFGEN_VAL_WFM_SINE` **frequency**: `10,000,000` **amplitude**: `1` **startPhase**: `0` **dcOffset**: `0`

4. Initiate waveform generation by calling the niFgen Initiate Generation VI.

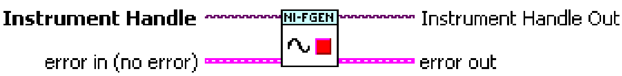
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitiateGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

5. Use the frequency counter to measure the frequency output at the CH 0 front panel connector of the NI 5402/5406. This value is the measured frequency of the generated sine wave.
6. Compare the measured frequency of the generated sine wave to the calibration test limits and published specifications shown in Table 2.

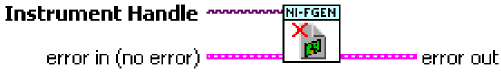
Table 2. Frequency Ranges for Verifying Oscillator Frequency Accuracy

Calibration Test Limits ±4.5 ppm		Published Specifications ±25 ppm	
Low	**High**	**Low**	**High**
9,999,955 Hz	10,000,045 Hz	9,999,750 Hz	10,000,250 Hz

7. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_AbortGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

8. Close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses by calling the niFgen Close VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_close` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

9. If any of the errors are greater than the *calibration test limits*, perform an external *Adjustment*.

Verifying the Sine Wave Flatness Correction Accuracy

This test verifies the Sine Wave Flatness Correction Accuracy of the NI 5402/5406. In this procedure, you will compare the signal generator's output power level during the generation of sine waves of various frequencies to the output power level measured during the generation of a 50 kHz sine wave. The following figure shows the programming flow for this test:

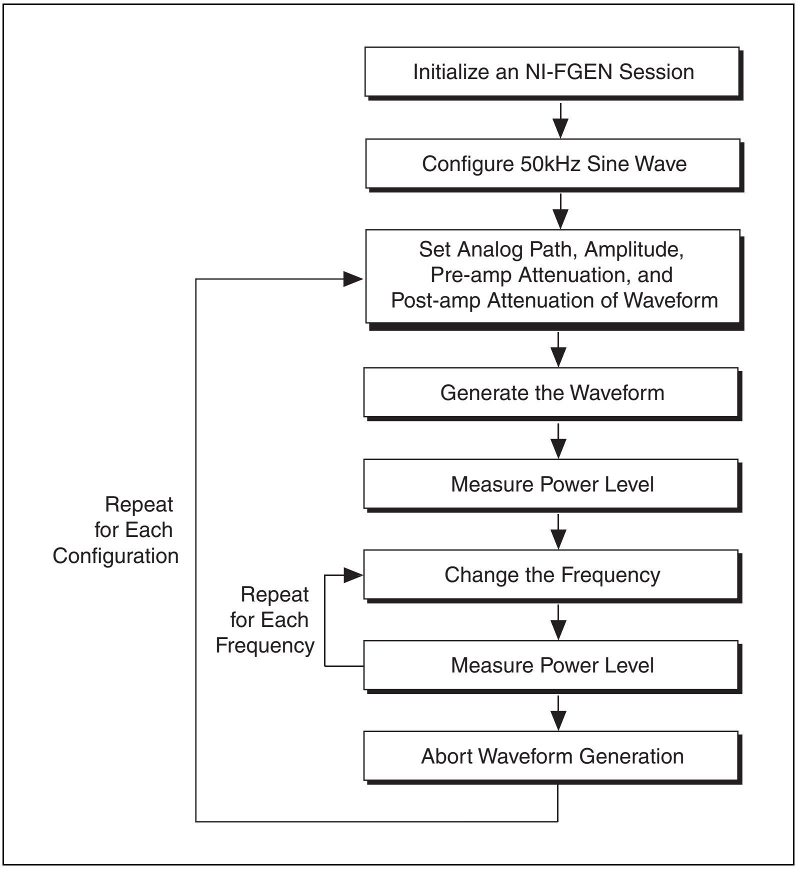
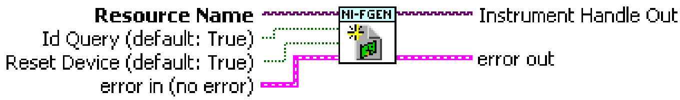
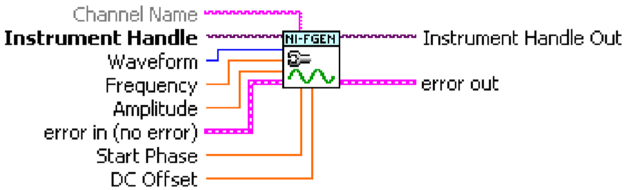


Figure 2. Programming Flow for the Verification of the Sine Wave Flatness Correction Accuracy

1. Connect the NI 5402/5406 CH 0 front panel connector to the power meter.
2. Open an instrument driver session, initialize the device for operation, and return a session handle that will be used to identify the device in future NI-FGEN calls by calling the niFgen Initialize VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_init` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **IDQuery**: `VI_TRUE` **resetDevice**: `VI_TRUE`

3. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

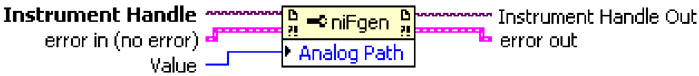
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ConfigureStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **waveform**: `NIFGEN_VAL_WFM_SINE` **frequency**: `50,000` **amplitude**: `1.667` **startPhase**: `0` **dcOffset**: `0`

Repeat steps 4 through 11 for each of the iterations listed in Table 3.

Table 3. Values for Verifying the Sine Wave Flatness Accuracy

Iteration	Analog Path	Pre-Amplifier Attenuation (dB)	Post-Amplifier Attenuation (dB)	Amplitude (V_{pp})	Calibration Test Limits (dB)	Published Specification (dB)
1	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	0.00	0.00	1.667000	±0.13	±0.4
2	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	3.00	0.00	1.179400	±0.13	±0.4
3	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	6.00	0.00	0.835000	±0.13	±0.4
4	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	9.00	0.00	0.591100	±0.13	±0.4
5	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	12.00	0.00	0.418500	±0.13	±0.4
6	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	0.00	12.00	2.500000	±0.13	±0.4
7	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	3.00	12.00	1.780000	±0.13	±0.4
8	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	6.00	12.00	1.260000	±0.13	±0.4
9	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	9.00	12.00	0.890000	±0.13	±0.4
10	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	12.00	12.00	0.630000	±0.13	±0.4

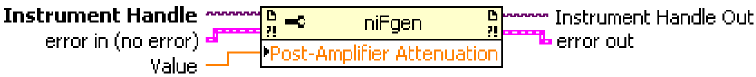
4. Set the analog path by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: The *Analog Path* value listed in Table 3 for the current iteration

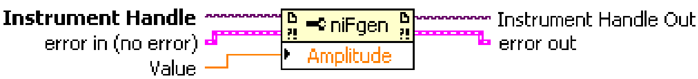
5. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: The *Pre-Amplifier Attenuation* value listed in Table 3 for the current iteration

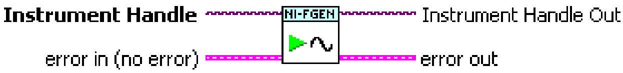
6. Set the post-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Post-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION` **value**: The *Post-Amplifier Attenuation* value listed in Table 3 for the current iteration

7. Set the amplitude by calling the niFgen Property Node and selecting **Standard Function Output»Amplitude**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_AMPLITUDE` **value**: The *Amplitude* value listed in Table 3 for the current iteration

8. Initiate waveform generation by calling the niFgen Initiate Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Initiate Generation` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

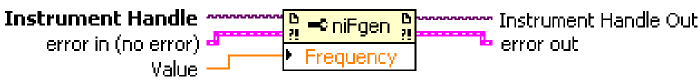
9. Use the power meter to measure the output power level of the NI 5402/5406. This value is the measured power of the 50 kHz sine wave.

10. In this step you generate a sine wave configured for the current iteration of Table 3 at each of the frequencies listed in Table 4.

Table 4. Frequencies for Verifying Sine Wave Flatness Correction Accuracy

Frequencies for the NI 5402 (MHz)	Frequencies for the NI 5406 (MHz)
1.00000	1.00000
5.00000	5.00000
10.00000	10.00000
15.00000	15.00000
20.00000	20.00000
25.00000	25.00000
—	30.00000
—	40.00000

a. Set the frequency by calling the niFgen Property Node and selecting **Standard Function Output»Frequency**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_FREQUENCY` **value**: The current *Frequency* value from Table 4

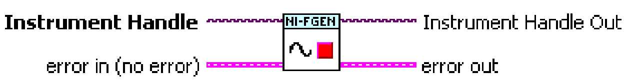
b. Use the power meter to measure the power level output of the NI 5402/5406. This value is the measured power of the generated sine wave (dB).

c. Check to see whether the sine wave flatness error falls between the high and low flatness error calibration test limits. Flatness error can be determined by using the following calculation:

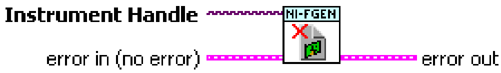
$$\textit{Sine Wave Flatness Error} = (\textit{Measured Power of Generated Sine Wave}) - (\textit{Measured Power of 50kHz Sine Wave})$$

d. Repeat steps 10a through 10c for each frequency in Table 4.

11. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_AbortGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

12. Close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses by calling the niFgen Close VI.

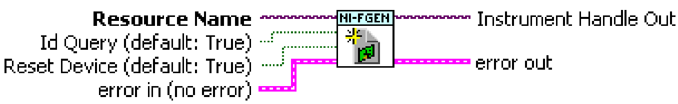
LabVIEW Block Diagram	C/C++ Function Call
 	Call `niFgen_close` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

13. If any of the errors are greater than the *calibration test limits*, perform an *Adjustment*.

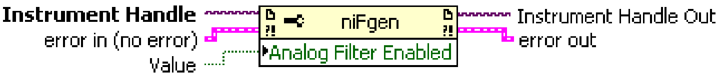
Verifying the Main Path Gain and Offset Accuracy

This test verifies the main path gain and offset accuracy of the NI 5402/5406 into a high-impedance load. In this procedure you generate a number of DC voltages and offsets, measure the voltage with a DMM, and compare the results to the error limits.

1. Connect the NI 5402/5406 CH 0 front panel connector to the DMM.
2. Open an instrument driver session, initialize the device for operation, and return a session handle that will be used to identify the device in future NI-FGEN calls by calling the niFgen Initialize VI.

LabVIEW Block Diagram	C/C++ Function Call
 	Call `niFgen_init` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **IDQuery**: `VI_TRUE` **resetDevice**: `VI_TRUE`

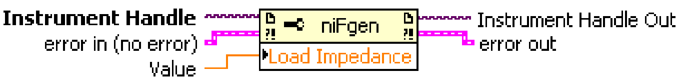
3. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>


</td><td>Call <code>niFgen_ SetAttribute ViBoolean</code> using the following parameters:

vi: The session handle returned from <code>niFgen_init</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_ANALOG_ FILTER_ENABLED</code>
value: <code>VI_FALSE</code></td></tr>
</table>

4. Set the load impedance by calling the niFgen Property Node and selecting **Output Attributes»Load Impedance**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>


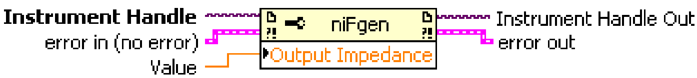
</td><td>Call <code>niFgen_Set AttributeViReal64</code> using the following parameters:

vi: The session handle returned from <code>niFgen_init</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ ATTR_LOAD_IMPEDANCE</code>
value: <code>10,000,000,000</code></td></tr>
</table>

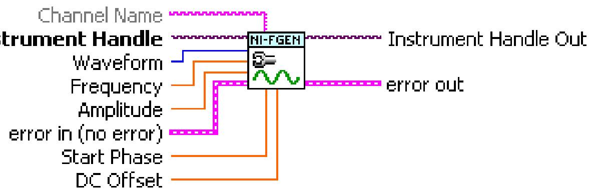
5. Set the analog path by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: `NIFGEN_VAL_MAIN_ANALOG_PATH`

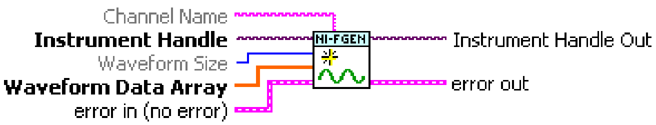
6. Set the output impedance by calling the niFgen Property Node and selecting **Basic Operation»Output Impedance**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE` **value**: `50.00`

7. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ConfigureStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **waveform**: `NIFGEN_VAL_WFM_USER` **frequency**: `1,000,000` **amplitude**: `1` **startPhase**: `0` **dcOffset**: `0`

8. Create an array of waveform samples for the positive full-scale DC waveform. This array should contain 16,384 samples with each sample having the value `1.0` (representation: double).
9. Define the waveform by calling the niFgen Define User Standard Waveform VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_DefineUserStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **waveformSize**: `16,384` **waveformDataArray**: The array of waveform samples created in step 8

Repeat steps 10 through 18 for each of the iterations listed in Table 5, changing the *Digital Gain*, *Amplitude*, and *DC Offset* values for each iteration.

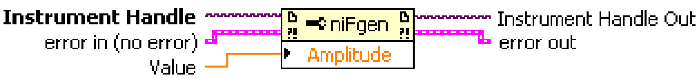
Table 5. Values for Verifying DC Gain and Offset Accuracy

Iteration	Digital Gain	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limit (V)	Published Specification (V)
1	1	18	1	10	±0.038	±0.097
2	–1	18	–1	–10	±0.038	±0.097
3	0	0.07	9.965	9.965	±0.015587	±0.052175
4	0	0.07	–9.965	–9.965	±0.015587	±0.052175
5	0	0.07	8.465	8.465	±0.013337	±0.044675
6	0	0.07	–8.465	–8.465	±0.013337	±0.044675
7	0	0.07	6.965	6.965	±0.011088	±0.037175
8	0	0.07	–6.965	–6.965	±0.011088	±0.037175
9	0	0.07	4.965	4.965	±0.008087	±0.027175
10	0	0.07	–4.965	–4.965	±0.008087	±0.027175
11	0	0.07	3.465	3.465	±0.005837	±0.019675
12	0	0.07	–3.465	–3.465	±0.005837	±0.019675
13	0	0.07	2.365	2.365	±0.004188	±0.014175
14	0	0.07	–2.365	–2.365	±0.004188	±0.014175
15	0	0.012	1.994	1.994	±0.003515	±0.01203
16	0	0.012	–1.994	–1.994	±0.003515	±0.01203
17	0	0.012	1.494	1.494	±0.002765	±0.00953
18	0	0.012	–1.494	–1.494	±0.002765	±0.00953
19	0	0.012	1.094	1.094	±0.002165	±0.00753
20	0	0.012	–1.094	–1.094	±0.002165	±0.00753
21	0	0.012	0.794	0.794	±0.001715	±0.00603
22	0	0.012	–0.794	–0.794	±0.001715	±0.00603
23	0	0.012	0.594	0.594	±0.001415	±0.00503
24	0	0.012	–0.594	–0.594	±0.001415	±0.00503
25	0	0.012	0.394	0.394	±0.001115	±0.00403
26	0	0.012	–0.394	–0.394	±0.001115	±0.00403
27	0	0.012	0.269	0.269	±0.000928	±0.003405

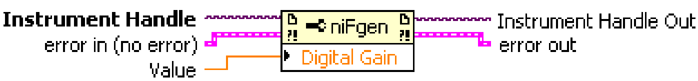
Table 5. Values for Verifying DC Gain and Offset Accuracy (Continued)

Iteration	Digital Gain	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limit (V)	Published Specification (V)
28	0	0.012	–0.269	–0.269	±0.000928	±0.003405
29	0	0.012	0.194	0.194	±0.000815	±0.00303
30	0	0.012	–0.194	–0.194	±0.000815	±0.00303
31	0	0.012	0.144	0.144	±0.00074	±0.00278
32	0	0.012	–0.144	–0.144	±0.00074	±0.00278
33	0	0.012	0.094	0.094	±0.000665	±0.00253
34	0	0.012	–0.094	–0.094	±0.000665	±0.00253
35	0	0.012	0.064	0.064	±0.00062	±0.00238
36	0	0.012	–0.064	–0.064	±0.00062	±0.00238
37	0	0.012	0.044	0.044	±0.00059	±0.00228
38	0	0.012	–0.044	–0.044	±0.00059	±0.00228
39	0	0.012	0.029	0.029	±0.000567	±0.002205
40	0	0.012	–0.029	–0.029	±0.000567	±0.002205
41	0	0.012	0.019	0.019	±0.000553	±0.002155
42	0	0.012	–0.019	–0.019	±0.000553	±0.002155
43	0	0.012	0.014	0.014	±0.000545	±0.00213
44	0	0.012	–0.014	–0.014	±0.000545	±0.00213
45	0	0.012	0.008	0.008	±0.000536	±0.0021
46	0	0.012	–0.008	–0.008	±0.000536	±0.0021
47	0	0.012	0.004	0.004	±0.00053	±0.00208
48	0	0.012	–0.004	–0.004	±0.00053	±0.00208
49	0	0.012	0.002	0.002	±0.000527	±0.00207
50	0	0.012	–0.002	–0.002	±0.000527	±0.00207

10. Set the amplitude by calling the niFgen Property Node and selecting **Standard Function Output»Amplitude**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_AMPLITUDE` **value**: The *Amplitude* value for the current iteration from Table 5

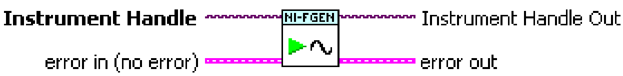
11. Set the digital gain by calling the niFgen Property Node and selecting **Output Attributes»Digital Gain**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_DIGITAL_GAIN` **value**: The *Digital Gain* value for the current iteration from Table 5

12. Set the DC Offset by calling the niFgen Property Node and selecting **Standard Function Output»DC Offset**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_DC_OFFSET` **value**: The *DC Offset* value for the current iteration from Table 5

13. Initiate waveform generation by calling the niFgen Initiate Generation VI.

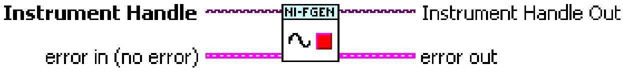
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitiateGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

14. Wait 200 ms for the output to settle.
15. Use the DMM to measure the DC voltage output of the NI 5402/5406. This value is the *measured DC output* value.
16. Determine the error for positive full scale using the following formula:

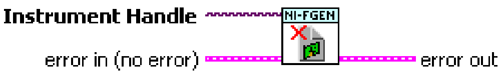
$$\text{DC Error} = (\text{Measured DC Output} - (\text{Ideal DC Output})$$

17. Compare the DC Error to the *calibration test limits* and the *published specifications*.

18. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_AbortGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

19. Close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses by calling the niFgen Close VI.

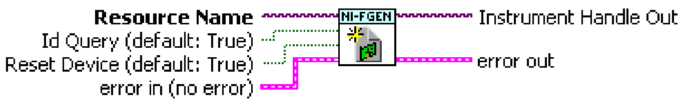
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_close` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

20. If any of the errors are greater than the *calibration test limits*, perform an *Adjustment*.

Verifying Square Wave Gain and Offset Accuracy

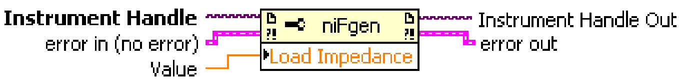
This test verifies the square wave gain and offset accuracy of the NI 5402/5406 into a high-impedance load. In this procedure you generate a square wave using a number of amplitudes and DC offsets, measure the voltage with a DMM, and compare the results to the error limits.

1. Connect the NI 5402/5406 CH 0 front panel connector to the DMM.
2. Open an instrument driver session, initialize the device for operation, and return a session handle that will be used to identify the device in future NI-FGEN calls by calling the niFgen Initialize VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_init</code> using the following parameters:

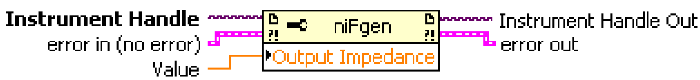
resourceName: The name of the device that you want to verify. This name is the device identifier assigned in MAX.
IDQuery: <code>VI_TRUE</code>
resetDevice: <code>VI_TRUE</code></td></tr>
</table>

3. Set the load impedance by calling the niFgen Property Node and selecting **Output Attributes»Load Impedance**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_SetAttributeViReal64</code> using the following parameters:

vi: The session handle returned from <code>niFgen_init</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_LOAD_IMPEDANCE</code>
value: <code>10,000,000,000</code></td></tr>
</table>

4. Set the output impedance by calling the niFgen Property Node and selecting **Basic Operation»Output Impedance**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE` **value**: `50.00`

Repeat steps 5 through 13 for each of the iterations listed in Table 6, changing the Analog Static Value, Amplitude, and DC Offset values for each iteration.

Table 6. Values for Verifying Square Wave Gain and Offset Accuracy

Iteration	Analog Static Value	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limits (V)	Published Specification (V)
1	7FFF	20	0	10	±0.021	±0.102
2	FFFF8000	20	0	–10	±0.021	±0.102
3	7FFF	17	0	8.5	±0.018	±0.087
4	FFFF8000	17	0	–8.5	±0.018	±0.087
5	7FFF	14	0	7	±0.015	±0.072
6	FFFF8000	14	0	–7	±0.015	±0.072
7	7FFF	10	0	5	±0.011	±0.052
8	FFFF8000	10	0	–5	±0.011	±0.052
9	7FFF	7	0	3.5	±0.008	±0.037
10	FFFF8000	7	0	–3.5	±0.008	±0.037
11	7FFF	5	0	2.5	±0.006	±0.027
12	FFFF8000	5	0	–2.5	±0.006	±0.027
13	7FFF	4	0	2	±0.005	±0.022

Table 6. Values for Verifying Square Wave Gain and Offset Accuracy (Continued)

Iteration	Analog Static Value	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limits (V)	Published Specification (V)
14	FFFF8000	4	0	–2	±0.005	±0.022
15	7FFF	3	0	1.5	±0.004	±0.017
16	FFFF8000	3	0	–1.5	±0.004	±0.017
17	7FFF	2.8	0	1.4	±0.0038	±0.016
18	FFFF8000	2.8	0	–1.4	±0.0038	±0.016
19	7FFF	2	0	1	±0.003	±0.012
20	FFFF8000	2	0	–1	±0.003	±0.012
21	7FFF	1.4	0	0.7	±0.0024	±0.009
22	FFFF8000	1.4	0	–0.7	±0.0024	±0.009
23	7FFF	1	0	0.5	±0.002	±0.007
24	FFFF8000	1	0	–0.5	±0.002	±0.007
25	7FFF	0.7	0	0.35	±0.0017	±0.0055
26	FFFF8000	0.7	0	–0.35	±0.0017	±0.0055
27	7FFF	0.5	0	0.25	±0.0015	±0.0045
28	FFFF8000	0.5	0	–0.25	±0.0015	±0.0045
29	7FFF	0.3	0	0.15	±0.0013	±0.0035
30	FFFF8000	0.3	0	–0.15	±0.0013	±0.0035
31	7FFF	0.2	0	0.1	±0.0012	±0.003
32	FFFF8000	0.2	0	–0.1	±0.0012	±0.003
33	7FFF	0.14	0	0.07	±0.00114	±0.0027
34	FFFF8000	0.14	0	–0.07	±0.00114	±0.0027
35	7FFF	0.1	0	0.05	±0.0011	±0.0025
36	FFFF8000	0.1	0	–0.05	±0.0011	±0.0025
37	7FFF	0.07	0	0.035	±0.00107	±0.00235
38	FFFF8000	0.07	0	–0.035	±0.00107	±0.00235
39	7FFF	0.05	0	0.025	±0.00105	±0.00225
40	FFFF8000	0.05	0	–0.025	±0.00105	±0.00225
41	7FFF	0.04	0	0.02	±0.00104	±0.0022
42	FFFF8000	0.04	0	–0.02	±0.00104	±0.0022

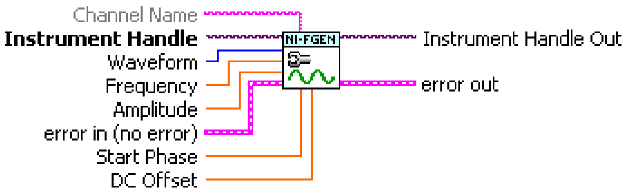
Table 6. Values for Verifying Square Wave Gain and Offset Accuracy (Continued)

Iteration	Analog Static Value	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limits (V)	Published Specification (V)
43	7FFF	0.03	0	0.015	±0.00103	±0.00215
44	FFFF8000	0.03	0	–0.015	±0.00103	±0.00215
45	7FFF	0.02	0	0.01	±0.00102	±0.0021
46	FFFF8000	0.02	0	–0.01	±0.00102	±0.0021
47	7FFF	0.012	0	0.006	±0.001012	±0.00206
48	FFFF8000	0.012	0	–0.006	±0.001012	±0.00206
49	7FFF	10	5	10	±0.016	±0.077
50	FFFF8000	10	–5	–10	±0.016	±0.077
51	7FFF	8.5	4.25	8.5	±0.01375	±0.06575
52	FFFF8000	8.5	–4.25	–8.5	±0.01375	±0.06575
53	7FFF	7	3.5	7	±0.0115	±0.0545
54	FFFF8000	7	–3.5	–7	±0.0115	±0.0545
55	7FFF	5	2.5	5	±0.0085	±0.0395
56	FFFF8000	5	–2.5	–5	±0.0085	±0.0395
57	7FFF	3.5	1.75	3.5	±0.00625	±0.02825
58	FFFF8000	3.5	–1.75	–3.5	±0.00625	±0.02825
59	7FFF	2.5	1.25	2.5	±0.00475	±0.02075
60	FFFF8000	2.5	–1.25	–2.5	±0.00475	±0.02075
61	7FFF	2	1	2	±0.004	±0.017
62	FFFF8000	2	–1	–2	±0.004	±0.017
63	7FFF	1.5	0.75	1.5	±0.00325	±0.01325
64	FFFF8000	1.5	–0.75	–1.5	±0.00325	±0.01325
65	7FFF	1.4	0.7	1.4	±0.0031	±0.0125
66	FFFF8000	1.4	–0.7	–1.4	±0.0031	±0.0125
67	7FFF	1	0.5	1	±0.0025	±0.0095
68	FFFF8000	1	–0.5	–1	±0.0025	±0.0095
69	7FFF	0.7	0.35	0.7	±0.00205	±0.00725
70	FFFF8000	0.7	–0.35	–0.7	±0.00205	±0.00725
71	7FFF	0.5	0.25	0.5	±0.00175	±0.00575

Table 6. Values for Verifying Square Wave Gain and Offset Accuracy (Continued)

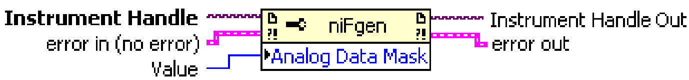
Iteration	Analog Static Value	Amplitude (V_{pp})	DC Offset (V)	Ideal DC Output (V)	Calibration Test Limits (V)	Published Specification (V)
72	FFFF8000	0.5	–0.25	–0.5	±0.00175	±0.00575
73	7FFF	0.35	0.175	0.35	±0.001525	±0.004625
74	FFFF8000	0.35	–0.175	–0.35	±0.001525	±0.004625
75	7FFF	0.25	0.125	0.25	±0.001375	±0.003875
76	FFFF8000	0.25	–0.125	–0.25	±0.001375	±0.003875
77	7FFF	0.15	0.075	0.15	±0.001225	±0.003125
78	FFFF8000	0.15	–0.075	–0.15	±0.001225	±0.003125
79	7FFF	0.1	0.05	0.1	±0.00115	±0.00275
80	FFFF8000	0.1	–0.05	–0.1	±0.00115	±0.00275
81	7FFF	0.07	0.035	0.07	±0.001105	±0.002525
82	FFFF8000	0.07	–0.035	–0.07	±0.001105	±0.002525
83	7FFF	0.05	0.025	0.05	±0.001075	±0.002375
84	FFFF8000	0.05	–0.025	–0.05	±0.001075	±0.002375
85	7FFF	0.035	0.0175	0.035	±0.001053	±0.0022625
86	FFFF8000	0.035	–0.0175	–0.035	±0.001053	±0.0022625
87	7FFF	0.025	0.0125	0.025	±0.001038	±0.0021875
88	FFFF8000	0.025	–0.0125	–0.025	±0.001038	±0.0021875
89	7FFF	0.02	0.01	0.02	±0.00103	±0.00215
90	FFFF8000	0.02	–0.01	–0.02	±0.00103	±0.00215
91	7FFF	0.015	0.0075	0.015	±0.001022	±0.0021125
92	FFFF8000	0.015	–0.0075	–0.015	±0.001022	±0.0021125
93	7FFF	0.012	0.005	0.011	±0.001017	±0.002085
94	FFFF8000	0.012	–0.005	–0.011	±0.001017	±0.002085
95	7FFF	0.012	0.002	0.008	±0.001014	±0.00207
96	FFFF8000	0.012	–0.002	–0.008	±0.001014	±0.00207

5. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_ConfigureStandardWaveform</code> using the following parameters:

vi: The session handle returned from <code>niFgen_init</code>.
channelName: <code>"0"</code>
waveform: <code>NIFGEN_VAL_WFM_SQUARE</code>
frequency: <code>1,000,000</code>
amplitude: The Amplitude value for the current iteration from Table 6
startPhase: <code>0</code>
dcOffset: The DC Offset value for the current iteration from Table 6</td></tr>
</table>

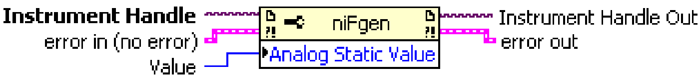
6. Set the analog data mask by calling the niFgen Property Node and selecting **Output Attributes»Data Mask»Analog Data Mask**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_SetAttributeViInt32</code> using the following parameter:

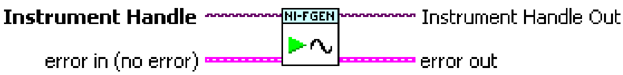
vi: The session handle returned from <code>niFgen_init</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_ANALOG_DATA_MASK</code>
value: <code>x0</code></td></tr>
</table>

Note The data mask is used to mask bits of data to be replaced with the corresponding static value bits. For example, a mask of 0xFF00 and a static value of 0xAAAA applied to a DC waveform with a value of 0x1111 produces a DC waveform with a value of 0x11AA.

7. Set the analog static value by calling the niFgen Property Node and selecting **Output Attributes»Data Mask»Analog Static Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameter: **vi**: The session handle returned from `niFgen_init`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_STATIC_VALUE` **value**: The current *Analog Static Value* from Table 6

8. Initiate waveform generation by calling the niFgen Initiate Generation VI.

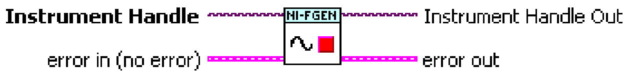
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitiateGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

9. Wait 200 ms for the output to settle.
10. Use the DMM to measure the DC voltage output of the NI 5402/5406. This is the *measured DC output* value.

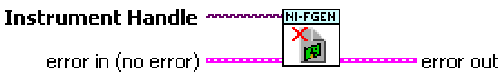
11. Determine the error for each measurement using the following formula:

$$\textit{DC Error} = (\textit{Measured DC Output Value}) - (\textit{Ideal DC Output Value})$$

12. Compare the DC error to the *calibration test limits* and *published specifications*.
13. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Abort Generation` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

14. Close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses by calling the niFgen Close VI.

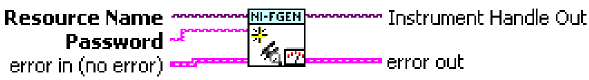
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_close` using the following parameter: **vi**: The session handle returned from `niFgen_init`.

15. If any of the errors are greater than the *calibration test limits*, perform a self-calibration procedure, followed by a reverification procedure.

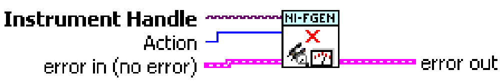
Updating the Calibration Date and Temperature

If the NI 5402/5406 passed verification within the calibration test limits and you do not want to perform an adjustment, you can update the calibration date and onboard calibration temperature without making any adjustments by completing the following steps:

1. Open an NI-FGEN external calibration session by calling the niFgen Init Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Init ExtCal` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **password**: `"NI"`

2. Close the instrument driver session and save the calibration date and temperature to the onboard EEPROM by calling the niFgen Close Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Close ExtCal` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal` **action**: `NIFGEN_VAL_ EXT_CAL_COMMIT`. This stores the date and temperature of the system at the time of calibration.

Adjustment

If the NI 5402/5406 successfully passes all verification within the calibration test limits, NI recommends adjustment to guarantee its published specifications for the next two years. If you choose not to adjust your device, refer to the *Updating the Calibration Date and Temperature* section for instructions on updating the calibration date and temperature without performing an adjustment. If the NI 5402/5406 is not within the calibration test limits for each verification procedure, perform the adjustment procedure to improve the accuracy of the NI 5402/5406. Refer to *Appendix A: Calibration Procedure Options* for more information on which procedures to perform.

The external calibration procedure adjusts the onboard calibration A/D converter (ADC), the oscillator frequency, the analog output, and the flatness correction. *Calibration ADC adjustment* characterizes the onboard ADC gain and offset so that self-calibration results in an accurately calibrated device. *Oscillator frequency adjustment* characterizes the onboard oscillator to ensure frequency accuracy. *Analog output adjustment* characterizes the DC gains and the offsets of the main and square wave analog path to ensure the analog output voltage accuracy. *Flatness Correction adjustment* characterizes the sine wave flatness correction applied to generated sine waves.

To perform an adjustment, create an external calibration session by calling the niFgen Init Ext Cal VI. Along with the standard NI-FGEN attributes, the external calibration session uses a set of calibration constants that are determined during the calibration procedure and stored in the device onboard memory when the session is closed. NI-FGEN uses these calibration constants during a standard NI-FGEN session to ensure that the device operates within its specifications. You must close an external calibration session by calling the niFgen Close Ext Cal VI.

The following figure shows the programming flow for an external calibration procedure.

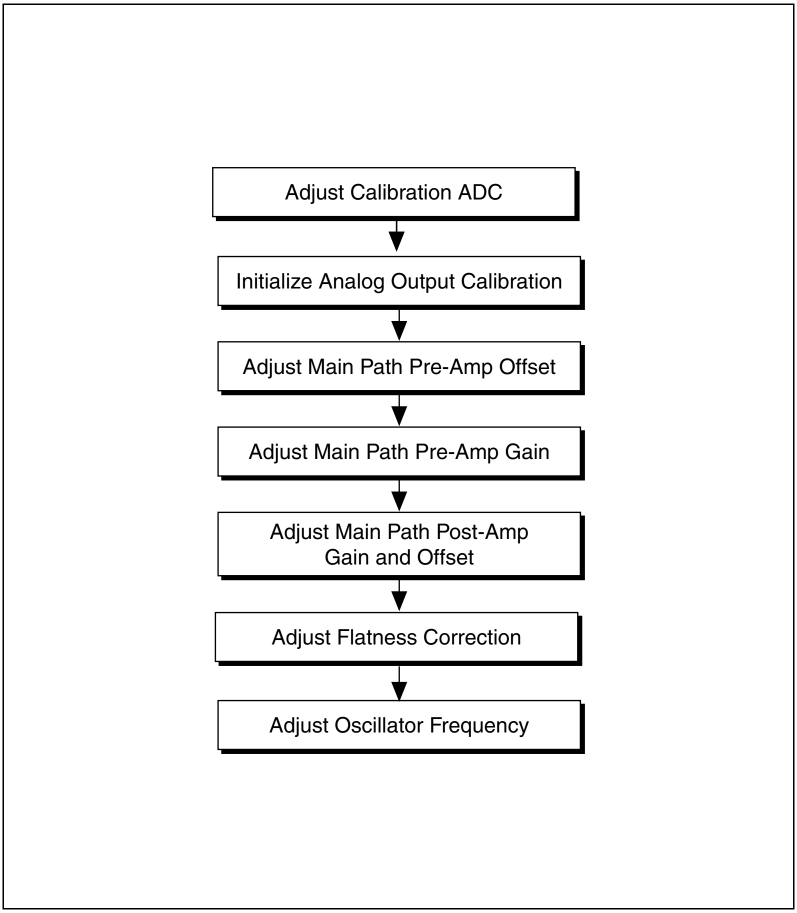
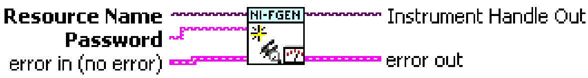


Figure 3. NI 5402/5406 External Calibration Procedure

Adjusting the Calibration ADC

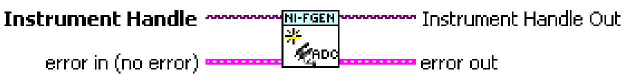
Complete this procedure to adjust the onboard calibration ADC of the NI 5402/5406 that is used during self-calibration and external calibration. You will characterize the gain and offset associated with this ADC.

1. Connect the NI 5402/5406 CH 0 front panel connector to a DMM.
2. Open an NI-FGEN external calibration session by calling the niFgen Init Ext Cal VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_Init ExtCal</code> using the following parameters:

resourceName: The name of the device that you want to verify. This name is the device identifier assigned in MAX.
password: <code>"NI"</code></td></tr>
</table>

3. Initialize ADC calibration by calling niFgen Initialize Cal ADC Calibration VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_Initialize CalADCCalibration</code> using the following parameter:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.</td></tr>
</table>

4. Set the analog path by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: `NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`

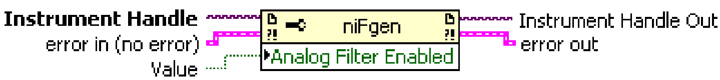
5. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE` **value**: `1,700`

6. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `32,767`

7. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED` **value**: `VI_FALSE`

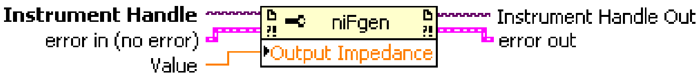
8. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: `0`

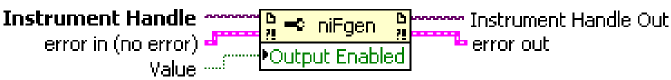
9. Set the post-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Post-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION` **value**: `0`

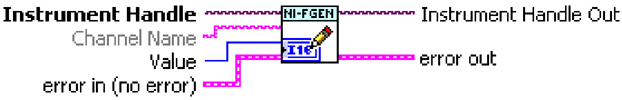
10. Set the output impedance by calling the niFgen Property Node and selecting **Basic Operation»Output Impedance**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE` **value**: `50.00`

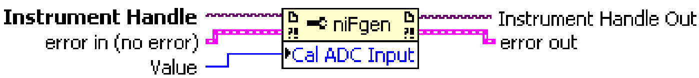
11. Disable the analog output by calling the niFgen Property Node and selecting **Basic Operation»Output Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED` **value**: `VI_FALSE`

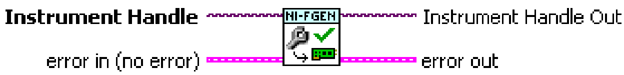
12. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Write Binary16Analog StaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `27,232`

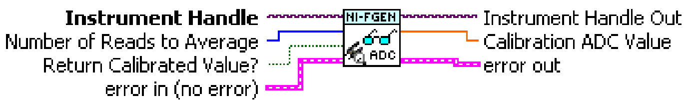
13. Set the calibration ADC input by calling the niFgen Property Node and selecting **Calibration»Cal ADC Input**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Set AttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `""` (empty string) **attributeID**: `NIFGEN_ ATTR_CAL_ADC_INPUT` **value**: `NIFGEN_VAL_ ANALOG_OUTPUT`

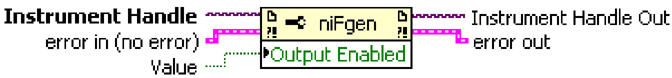
14. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

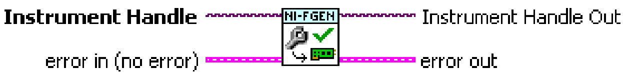
15. Wait 500 ms for the output to settle.
16. Measure the analog output voltage with the onboard calibration ADC by calling the niFgen Read CAL ADC VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Read CalADC` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **numberOfReads ToAverage**: `3` **returnCalibratedValue**: `VI_FALSE` **calADCValue**: Returns a ViReal64 variable. The variable passed by reference through this parameter receives the voltage measured by the onboard ADC. This value is `cal ADC measurement 0`, which is used in step 30.

17. Enable the analog output by calling the niFgen Property Node and selecting **Basic Operation»Output Enabled**.

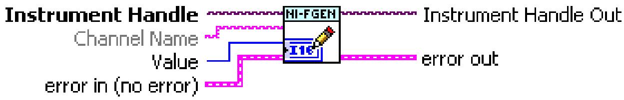
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED` **value**: `VI_TRUE`

18. Commit the attribute values to the device by calling the niFgen Commit VI.

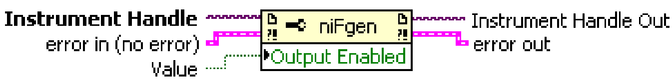
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

19. Wait 500 ms for the output to settle.
20. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `external measurement 0`, which is used in step 30.

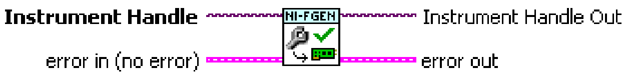
21. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `10,232`

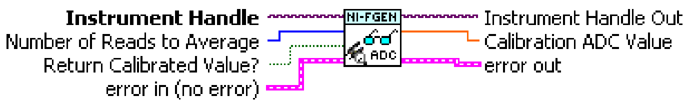
22. Disable the analog output by calling the niFgen Property Node and selecting **Basic Operation»Output Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED` **value**: `VI_FALSE`

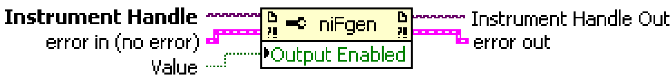
23. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

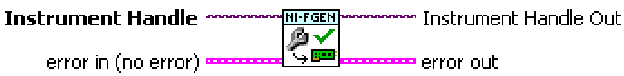
24. Wait 500 ms for the output to settle.
25. Measure the analog output voltage with the onboard calibration ADC by calling the niFgen Read CAL ADC VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Read CalADC` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **numberOfReadsToAverage**: `3` **returnCalibratedValue**: `VI_FALSE` **calADCValue**: Returns a ViReal64 variable. The variable passed by reference through this parameter receives the voltage measured by the onboard ADC. This value is `cal ADC measurement 1`, which is used in step 30.

26. Enable the analog output by calling the niFgen Property Node and selecting **Basic Operation»Output Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED` **value**: `VI_TRUE`

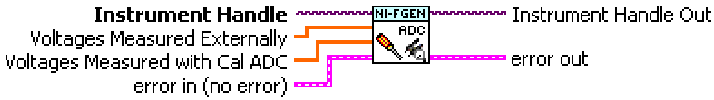
27. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

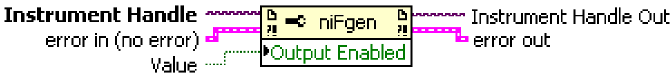
28. Wait 500 ms for the output to settle.

29. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `external measurement 1`, which is used in step 30.

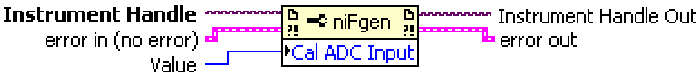
30. Adjust the ADC calibration by calling the niFgen Cal Adjust Cal ADC VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Cal AdjustCalADC` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **voltagesMeasured Externally**: `{external measurement 0, external measurement 1}` **voltagesMeasured WithCalADC**: `{cal ADC measurement 0, cal ADC measurement 1}`

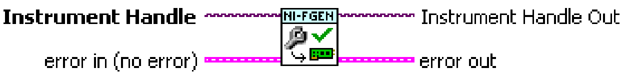
31. Disable the analog output by calling the niFgen Property Node and selecting **Basic Operation»Output Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Set AttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED` **value**: `VI_FALSE`

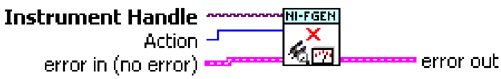
32. Set the calibration ADC input by calling the niFgen Property Node and selecting **Calibration»Cal ADC Input**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `""` (empty string) **attributeID**: `NIFGEN_ATTR_CAL_ADC_INPUT` **value**: `NIFGEN_VAL_GROUND`

33. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

34. Close the instrument driver session and save the calibration date and temperature by calling the niFgen Close Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_CloseExtCal` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **action**: If the external adjustment procedure completed without any errors, use `NIFGEN_VAL_EXT_CAL_COMMIT`. This function stores the new calibration constants, updated calibration dates, updated calibration temperatures in the onboard EEPROM. If any errors occurred during the external adjustment procedure, or if you want to abort the operation, use `NIFGEN_VAL_EXT_CAL_ABORT`. This function then discards the new calibration constants and does not change any of the calibration data stored in the onboard EEPROM.

Adjusting the Analog Output

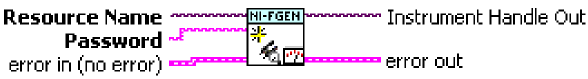
The analog output adjustment procedure has several sub-procedures that adjust the following:

- Main path pre-amplifier offset
- Main path pre-amplifier gain
- Main path post-amplifier gain and offset

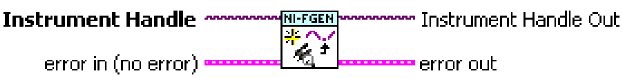
In each of these sub-procedures, you put the device in several configurations and take several output measurements. You then pass these measurements to NI-FGEN, which determines the calibration constants for the device.

Initializing Analog Output Calibration

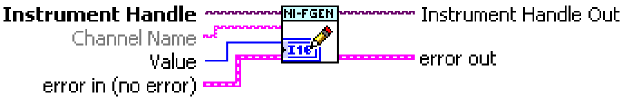
1. Connect the NI 5402/5406 CH 0 front panel connector to the DMM.
2. Open an NI-FGEN external calibration session by calling the niFgen Init Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitExtCal` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **password**: `"NI"`

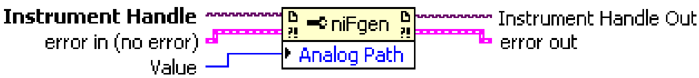
3. Initialize analog output calibration by calling the niFgen Initialize Analog Output Calibration VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitializeAnalogOutputCalibration` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

4. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `0`

5. Set the analog path value by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: `NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`

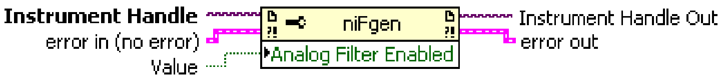
6. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE` **value**: `2,000`

7. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `32,767`

8. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ SetAttribute ViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_ FILTER_ENABLED` **value**: `VI_FALSE`

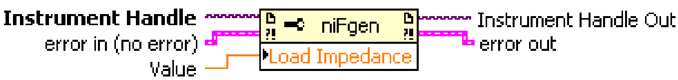
9. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Set AttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal.`. **channelName**: `"0"` **attributeID**: `NIFGEN_ ATTR_PRE_AMPLIFIER_ ATTENUATION` **value**: `0.00`

10. Set the post-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Post-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION` **value**: `0.00`

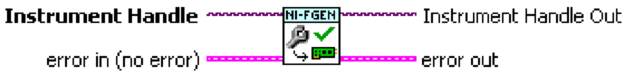
11. Set the load impedance by calling the niFgen Property Node and selecting **Output Attributes»Load Impedance**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_LOAD_IMPEDANCE` **value**: `10,000,000,000`

12. Set the waveform to user-defined by calling the niFgen Property Node and selecting **Standard Function Output»Waveform**.

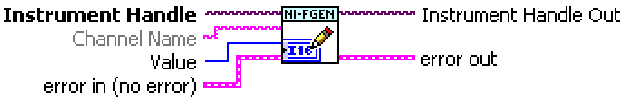
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_WAVEFORM` **value**: `NIFGEN_VAL_WVFM_USER`

13. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

Adjusting the Main Path Pre-Amplifier Offset

1. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

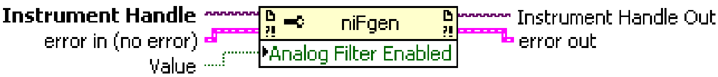
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `0`

Repeat steps 2 through 5 for each of the 10 iterations listed in Table 7, changing the *Analog Filter Enable*, *Pre-Amplifier Attenuation*, and *Current Configuration* values for each iteration.

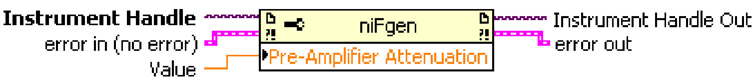
Table 7. Attributes and Values for Main Path Pre-Amplifier Offset

Iteration	Analog Filter Enable	Pre-Amplifier Attenuation (dB)	Current Configuration
1	VI_FALSE	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_0DB
2	VI_FALSE	3	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_3DB
3	VI_FALSE	6	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_6DB
4	VI_FALSE	9	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_9DB
5	VI_FALSE	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_12DB
6	VI_TRUE	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_0DB
7	VI_TRUE	3	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_3DB
8	VI_TRUE	6	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_6DB
9	VI_TRUE	9	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_9DB
10	VI_TRUE	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_12DB

2. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED` **value**: The *Analog Filter Enable* value for the current iteration from Table 7

3. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: The *Pre-Amplifier Attenuation* value for the current iteration from Table 7

4. Take the following voltage measurements at the NI 5402/5406 CH 0 front panel connector into a high-impedance load:

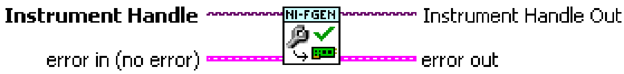
   a. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE` **value**: `2,000`

   b. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `38,000`

c. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

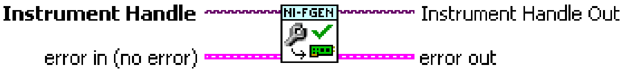
d. Wait 500 ms for the output to settle.

e. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 0`, which is used in step 5.

f. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE` **value**: `1,000`

g. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`

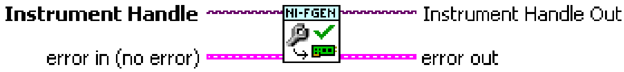
h. Wait 500 ms for the output to settle.

i. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 1`, which is used in step 5.

j. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Set AttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_ DAC_VALUE` **value**: `26,000`

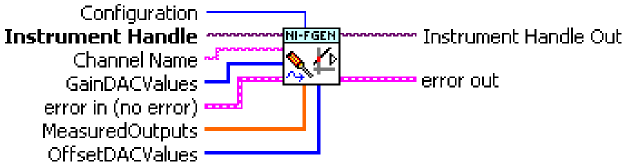
k. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

l. Wait 500 ms for the output to settle.

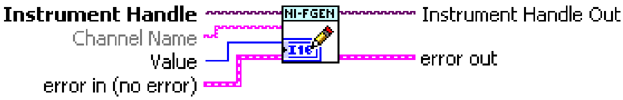
m. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 2`, which is used in step 5.

5. Adjust the pre-amplifier main path offset by calling the niFgen Cal Adjust Main Path Pre Amp Offset VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_CalAdjustMainPathPreAmpOffset` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **configuration**: The *Current Configuration* value for the current iteration from Table 7 **gainDACValues**: `{2000,1000}` **offsetDACValues**: `{38000,26000}` **measuredOutputs**: `{measurement 0, measurement 1, measurement 2}`

Adjusting the Main Path Pre-Amplifier Gain

1. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `0`

2. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

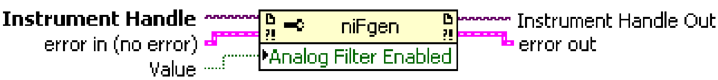
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `32,000`

Repeat steps 3 through 6 for each of the 10 iterations listed in Table 8, changing the *Analog Filter Enable*, *Pre-Amplifier Attenuation*, and *Current Configuration* values for each iteration.

Table 8. Attributes and Values for Adjusting the Main Path Pre-Amplifier Gain

Iteration	Analog Filter Enable Value	Pre-Amplifier Attenuation (dB)	Current Configuration
1	VI_FALSE	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_0DB
2	VI_FALSE	3	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_3DB
3	VI_FALSE	6	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_6DB
4	VI_FALSE	9	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_9DB
5	VI_FALSE	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_OFF_12DB
6	VI_TRUE	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_0DB
7	VI_TRUE	3	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_3DB
8	VI_TRUE	6	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_6DB
9	VI_TRUE	9	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_9DB
10	VI_TRUE	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_FILTER_ON_12DB

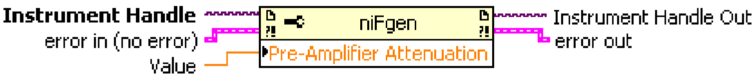
3. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++ Function Call</th></tr>
<tr><td>


</td><td>Call <code>niFgen_SetAttributeViBoolean</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_ANALOG_FILTER_ENABLED</code>
value: The Analog Filter Enable value for the current iteration from Table 8</td></tr>
</table>

4. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

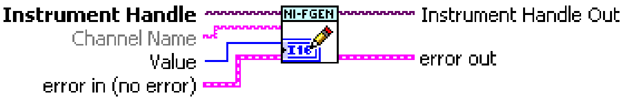
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: The *Pre-Amplifier Attenuation* value for the current iteration from Table 8

5. Take the following voltage measurements at the NI 5402/5406 CH 0 front panel connector into a high-impedance load:

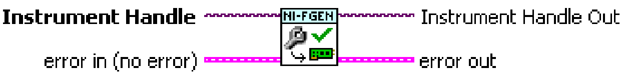
   a. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE` **value**: `1,500`

b. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `25,233`

c. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`

d. Wait 500 ms for the output to settle.

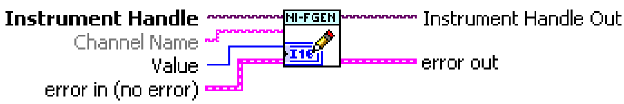
e. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 0`, which is used in step 6.

f. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_SetAttributeViInt32</code> using the following parameters:

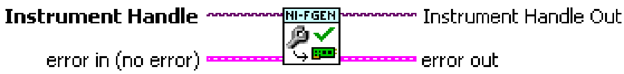
vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_GAIN_DAC_VALUE</code>
value: <code>2,000</code></td></tr>
</table>

g. Set the main DAC value by calling the niFgen Write Binary 16 Analog Static Value VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_WriteBinary16AnalogStaticValue</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
value: <code>–29,232</code></td></tr>
</table>

h. Commit the attribute values to the device by calling the niFgen Commit VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>


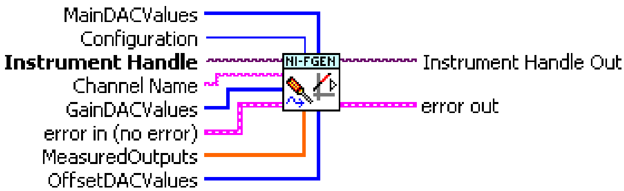
</td><td>
Call <code>niFgen_Commit</code> using the following parameter:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
</td></tr>
</table>

i. Wait 500 ms for the output to settle.

j. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 1`, which is used in step 6.

6. Adjust the pre-amplifier main path gain and offset by calling the niFgen Cal Adjust Main Path Pre Amp Gain VI.

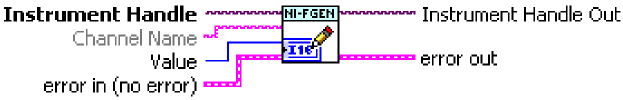
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>


</td><td>
Call <code>niFgen_Cal AdjustMainPath PreAmpGain</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
configuration: The Current Configuration value for the current iteration from Table 8
mainDACValues: <code>{25233, –29232}</code>
gainDACValues: <code>{1500, 2000}</code>
offsetDACValues: <code>{32000}</code>
measuredOutputs: <code>{measurement 0, measurement 1}</code>
</td></tr>
</table>

Adjusting the Main Path Post-Amplifier Gain and Offset

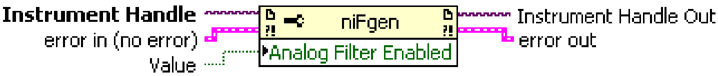
1. Set the main DAC value by calling niFgen Write Binary 16 Analog Static Value VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_WriteBinary16AnalogStaticValue` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **value**: `0`

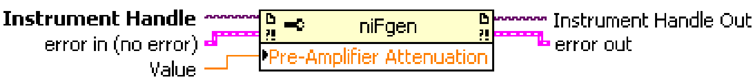
2. Set the waveform to user-defined by calling the niFgen Property Node and selecting **Standard Function Output»Waveform**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_WAVEFORM` **value**: `NIFGEN_VAL_WVFM_USER`

3. Set the analog filter state by calling the niFgen Property Node and selecting **Output Attributes»Analog Filter Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED` **value**: `VI_FALSE`

4. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: `0`

5. Set the gain DAC value by calling the niFgen Property Node and selecting **Calibration»Gain DAC Value**.

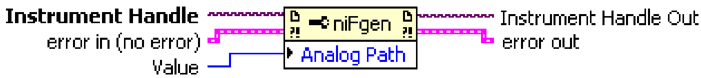
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_SetAttributeViInt32</code> using the following parameters:
vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_GAIN_DAC_VALUE</code>
value: <code>2,000</code></td></tr>
</table>

Repeat steps 6 through 9 for each of the eight iterations listed in Table 9, changing the *Analog Path*, *Post-Amplifier Attenuation*, and *Current Configuration* values for each iteration.

Table 9. Values for Adjusting the Main Path Post-Amplifier Gain and Offset

Iteration	Analog Path	Post-Amp Attenuation (dB)	Current Configuration
1	NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_0DB
2	NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_12DB
3	NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	24	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_24DB
4	NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH	36	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_36DB
5	NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_HIGH_GAIN_0DB
6	NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_HIGH_GAIN_12DB
7	NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	24	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_HIGH_GAIN_24DB
8	NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH	36	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_HIGH_GAIN_36DB

6. Set the analog path by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: The *Analog Path* value for the current iteration from Table 9

7. Set the post-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Post-Amplifier Attenuation**.

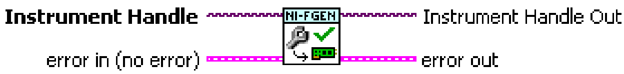
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION` **value**: The *Post-Amplifier Attenuation* value for the current iteration from Table 9

8. Take the following voltage measurements at the NI 5402/5406 CH 0 front panel connector into a high-impedance load:

   a. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `38,000`

   b. Commit the attribute values to the device by calling the niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

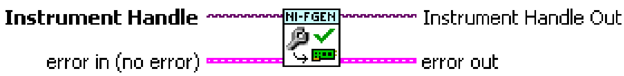
   c. Wait 500 ms for the output to settle.

   d. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 0`, which is used in step 9.

e. Set the offset DAC value by calling the niFgen Property Node and selecting **Calibration»Offset DAC Value**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE` **value**: `27,000`

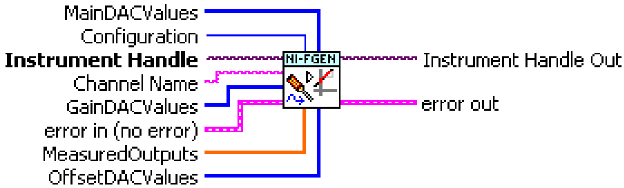
f. Commit the attribute values to the device by calling niFgen Commit VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Commit` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

g. Wait 500 ms for the output to settle.

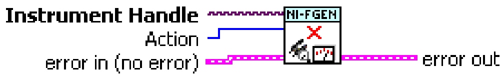
h. Use the DMM to measure the NI 5402/5406 output voltage. This measurement is `measurement 1`, which is used in step 9.

9. Adjust the post-amplifier main path gain and offset by calling the niFgen Cal Adjust Main Path Post Amp Gain And Offset VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_Cal AdjustMainPath PostAmpGainAnd Offset</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
configuration: The Current Configuration value for the current iteration from Table 9
mainDACValues: <code>{0, 0}</code>
gainDACValues: <code>{2000}</code>
offsetDACValues: <code>{38000, 27000}</code>
measuredOutputs: <code>{measurement 0, measurement 1}</code></td></tr>
</table>

10. Close the instrument driver session and save the calibration date and temperature by calling the niFgen Close Ext Cal VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td></td><td>Call <code>niFgen_CloseExtCal</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.

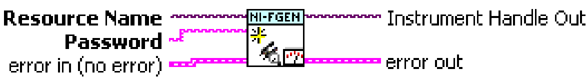
action: If the external adjustment procedure completed without any errors, use <code>NIFGEN_VAL_EXT_CAL_COMMIT</code>. This function then stores the new calibration constants, updated calibration dates, updated calibration temperatures in the onboard EEPROM.

If any errors occurred during the external adjustment procedure, or if you want to abort the operation, use <code>NIFGEN_VAL_EXT_CAL_ABORT</code>. This function then discards the new calibration constants and does not change any of the calibration data stored in the onboard EEPROM.</td></tr>
</table>

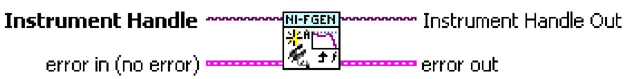
Adjusting the Sine Wave Flatness Correction

Complete this procedure to adjust the sine wave flatness correction of the NI 5402/5406. In this procedure, you compare the signal generator's power level output during the generation of sine waves of various frequencies to determine the flatness of the sine wave.

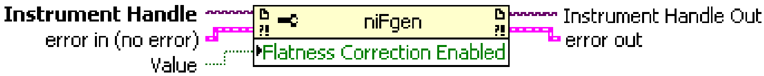
1. Connect the NI 5402/5406 CH 0 front panel connector to a power meter.
2. Open an NI-FGEN external calibration session by calling the niFgen Init Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitExtCal` using the following parameters: **resourceName**: The name of the device that you want to verify. This name is the device identifier assigned in MAX. **password**: `"NI"`

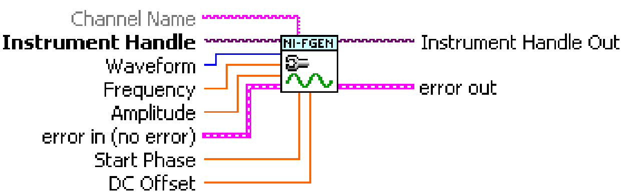
3. Initialize flatness calibration by calling the niFgen Initialize Flatness Calibration VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitializeFlatnessCalibration` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

4. Set the flatness correction state by calling the niFgen Property Node and selecting **Calibration»Flatness Correction Enabled**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViBoolean` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal` **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FLATNESS_CORRECTION_ENABLED` **value**: `VI_FALSE`

5. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

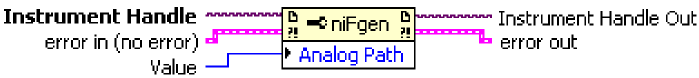
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ConfigureStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **waveform**: `NIFGEN_VAL_WFM_SINE` **frequency**: `50,000` **amplitude**: `2.00` **startPhase**: `0` **dcOffset**: `0`

Repeat steps 6 through 13 for each of the 10 iterations listed in Table 10, changing the *Path*, *Pre-Amplifier Attenuation*, *Post-Amplifier Attenuation*, and *Amplitude* values for each iteration.

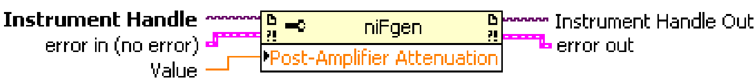
Table 10. Values for Adjusting Flatness Correction

Iteration	Analog Path	Pre-Amplifier Attenuation (dB)	Post-Amplifier Attenuation (dB)	Amplitude (V_{pp})	Current Configuration
1	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	0.00	0.00	1.700000	`NIFGEN_VAL_CAL_CONFIG_LOW_GAIN_PATH_PRE_AMP_0DB`
2	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	3.00	0.00	1.200000	`NIFGEN_VAL_CAL_CONFIG_LOW_GAIN_PATH_PRE_AMP_3DB`
3	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	6.00	0.00	0.850000	`NIFGEN_VAL_CAL_CONFIG_LOW_GAIN_PATH_PRE_AMP_6DB`
4	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	9.00	0.00	0.600000	`NIFGEN_VAL_CAL_CONFIG_LOW_GAIN_PATH_PRE_AMP_9DB`
5	`NIFGEN_VAL_FIXED_LOW_GAIN_ANALOG_PATH`	12.00	0.00	0.420000	`NIFGEN_VAL_CAL_CONFIG_LOW_GAIN_PATH_PRE_AMP_12DB`
6	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	0.00	12.00	2.500000	`NIFGEN_VAL_CAL_CONFIG_HIGH_GAIN_PATH_PRE_AMP_0DB`
7	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	3.00	12.00	1.780000	`NIFGEN_VAL_CAL_CONFIG_HIGH_GAIN_PATH_PRE_AMP_3DB`
8	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	6.00	12.00	1.260000	`NIFGEN_VAL_CAL_CONFIG_HIGH_GAIN_PATH_PRE_AMP_6DB`
9	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	9.00	12.00	0.890000	`NIFGEN_VAL_CAL_CONFIG_HIGH_GAIN_PATH_PRE_AMP_9DB`
10	`NIFGEN_VAL_FIXED_HIGH_GAIN_ANALOG_PATH`	12.00	12.00	0.630000	`NIFGEN_VAL_CAL_CONFIG_HIGH_GAIN_PATH_PRE_AMP_12DB`

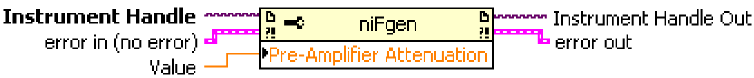
6. Set the analog path by calling the niFgen Property Node and selecting **Output Attributes»Analog Path**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViInt32` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_ANALOG_PATH` **value**: The *Analog Path* value for the current iteration from Table 10

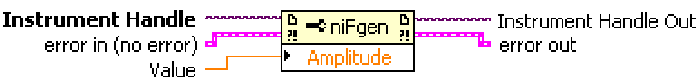
7. Set the post-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Post-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION` **value**: The *Post-Amplifier Attenuation* value for the current iteration from Table 10

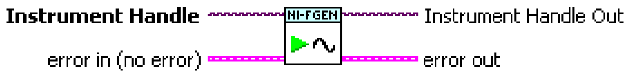
8. Set the pre-amplifier attenuation by calling the niFgen Property Node and selecting **Calibration»Pre-Amplifier Attenuation**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION` **value**: The *Pre-Amplifier Attenuation* value for the current iteration from Table 10

9. Set the amplitude by calling the niFgen Property Node and selecting **Standard Function Output»Amplitude**.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_SetAttributeViReal64` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **attributeID**: `NIFGEN_ATTR_FUNC_AMPLITUDE` **value**: The *Amplitude* value for the current iteration from Table 10

10. Initiate waveform generation by calling the niFgen Initiate Generation VI.

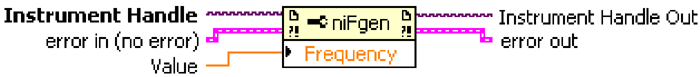
LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Initiate Generation` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

11. Generate a sine wave configured for the current iteration of Table 10 at each of the frequencies listed in Table 11. These steps will be repeated for each iteration in Table 10.

Table 11. Frequencies for Adjusting Sine Wave Flatness Correction

Frequencies for the NI 5402 (Hz)	Frequencies for the NI 5406 (Hz)
100,000	100,000
1,000,000	1,000,000
5,000,000	5,000,000
10,000,000	10,000,000
15,000,000	15,000,000
20,000,000	20,000,000
—	25,000,000
—	30,000,000
—	40,000,000

a. Set the frequency by calling the niFgen Property Node and selecting **Standard Function Output»Frequency**.

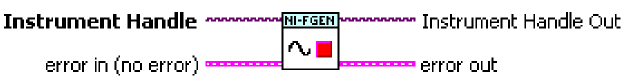
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_SetAttributeViReal64</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.
channelName: <code>"0"</code>
attributeID: <code>NIFGEN_ATTR_FUNC_FREQUENCY</code>
value: The current Frequency value from Table 11</td></tr>
</table>

b. Use the power meter to measure the power generated by the NI 5402/5406 during the generation of the sine wave.

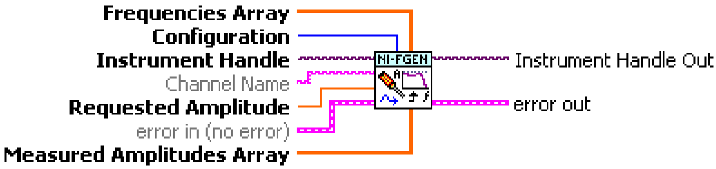
c. Repeat steps 11a through 11b for each frequency in Table 11.

12. Abort waveform generation by calling the niFgen Abort Generation VI.

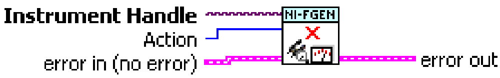
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_AbortGeneration</code> using the following parameter:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.</td></tr>
</table>

13. Configure the flatness by calling the niFgen Cal Adjust Flatness VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Cal AdjustFlatness` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **frequenciesArray**: An array of the frequencies from Table 11 **configuration**: The *Current Configuration* value for the current iteration from Table 11 **requestedAmplitude AtCalibration**: The requested amplitude value for the current configuration from Table 10 **measuredAmplitudes**: An array of the amplitudes measured by the power meter during the current iteration

14. Close the instrument driver session and save the calibration date and temperature by calling the niFgen Close Ext Cal VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_CloseExtCal</code> using the following parameters:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.

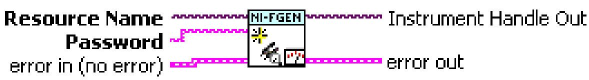
action: If the external adjustment procedure completed without any errors, use <code>NIFGEN_VAL_EXT_CAL_COMMIT</code>. This function then stores the new calibration constants, updated calibration dates, updated calibration temperatures in the onboard EEPROM.

If any errors occurred during the external adjustment procedure, or if you want to abort the operation, use <code>NIFGEN_VAL_EXT_CAL_ABORT</code>. This function then discards the new calibration constants and does not change any of the calibration data stored in the onboard EEPROM.</td></tr>
</table>

Adjusting the Oscillator Frequency

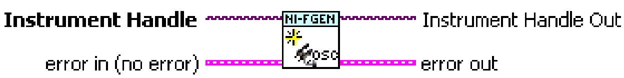
Complete this procedure to adjust the oscillator frequency of the NI 5402/5406. In this procedure, you generate a 10 MHz sine wave, iteratively measure the sine wave frequency with a frequency counter, and adjust the oscillator frequency until the measured frequencies fall within the desired tolerance 4.5 ppm.

1. Connect the NI 5402/5406 CH 0 front panel connector to the frequency counter.
2. Open an NI-FGEN external calibration session by calling the niFgen Init Ext Cal VI.

<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_Init ExtCal</code> using the following parameters:

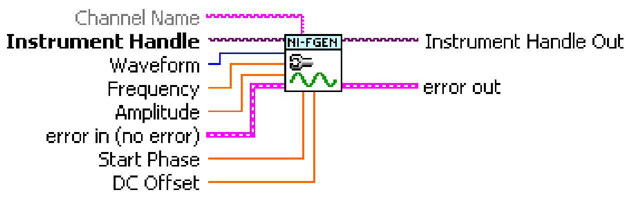
resourceName: The name of the device that you want to verify. This name is the device identifier assigned in MAX.
password: <code>"NI"</code></td></tr>
</table>

3. Initialize oscillator frequency calibration by calling the niFgen Initialize Oscillator Frequency Calibration VI.

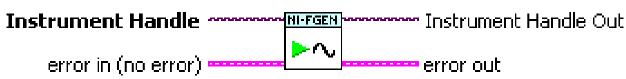
<table>
<tr><th>LabVIEW Block Diagram</th><th>C/C++
Function Call</th></tr>
<tr><td>
</td><td>Call <code>niFgen_ InitializeOscillatorFrequency Calibration</code> using the following parameter:

vi: The session handle returned from <code>niFgen_InitExtCal</code>.</td></tr>
</table>

4. Configure the waveform by calling the niFgen Configure Standard Waveform VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_ConfigureStandardWaveform` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **channelName**: `"0"` **waveform**: `NIFGEN_VAL_WFM_SINE` **frequency**: `10,000,000` **amplitude**: `1` **startPhase**: `0` **dcOffset**: `0`

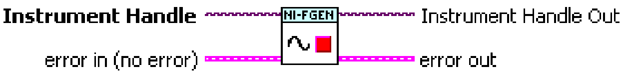
5. Initiate waveform generation by calling the niFgen Initiate Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_InitiateGeneration` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

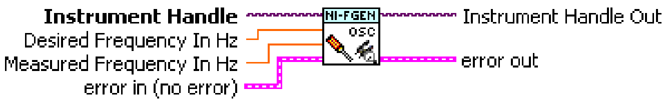
6. Use the frequency counter to measure the frequency of the generated waveform. This value is the first measured frequency that is used in step 7b.

7. The measured frequency should converge on the desired frequency. If the measured frequency does not converge on the desired frequency within 16 iterations, a problem may exist with your measurement device or the NI 5402/5406.

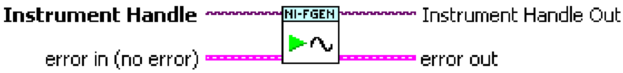
   a. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Abort Generation` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

   b. Adjust the oscillator frequency by calling the niFgen Cal Adjust Oscillator Frequency VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_CalAdjust OscillatorFrequency` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal`. **desiredFrequencyInHz**: `10000000` **measuredFrequencyIn Hz**: The measured frequency value (in Hz)

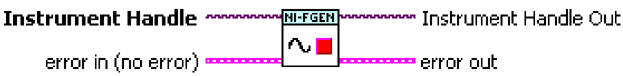
c. Initiate waveform generation by calling the niFgen Initiate Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Initiate Generation` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

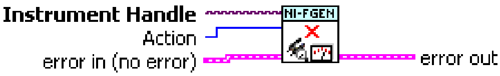
d. Use the frequency counter to measure the frequency of the generated waveform. This value is the new measured frequency.

e. Repeat steps 7a through 7d until the difference between the measured frequency and the desired frequency (10 MHz) is less than or equal to the tolerance (4.5 ppm).

8. Abort waveform generation by calling the niFgen Abort Generation VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_Abort Generation` using the following parameter: **vi**: The session handle returned from `niFgen_InitExtCal`.

9. Close the instrument driver session and save the calibration date and temperature by calling the niFgen Close Ext Cal VI.

LabVIEW Block Diagram	C/C++ Function Call
	Call `niFgen_CloseExtCal` using the following parameters: **vi**: The session handle returned from `niFgen_InitExtCal` **action**: If the external adjustment procedure completed without any errors, use `NIFGEN_VAL_EXT_CAL_COMMIT`. This function stores the new calibration constants, updated calibration dates, updated calibration temperatures in the onboard EEPROM. If any errors occurred during the external adjustment procedure, or if you want to abort the operation, use `NIFGEN_VAL_EXT_CAL_ABORT`. This function discards the new calibration constants and does not change any of the calibration data stored in the onboard EEPROM.

Appendix A: Calibration Procedure Options

External Calibration

External calibration involves both verification and adjustment. Verification is the process of testing the device to ensure that the output accuracy is within certain specifications. Adjustment is the process of measuring and compensating for device performance to improve the output accuracy. The device is guaranteed to meet or exceed its published specifications for the duration of the calibration interval.

You can use the two sets of test limits provided in this document (the calibration test limits and the published specifications) to perform a verification to determine whether an adjustment process needs to be performed or, if an adjustment has already been performed, to ensure that it was successful.

If all the output errors determined during verification fall within the calibration test limits, the device is guaranteed to meet or exceed its published specifications for a full calibration interval (two years). For this reason, you must verify against the calibration test limits when performing verification after adjustment.

Published specification values are less restrictive than the calibration test limits. If all of the output errors determined during verification fall within the published specifications, but not within the calibration test limits, the device currently meets its published specifications. The device will meet published specifications for the rest of the current calibration interval, but may not remain within these specifications for another two years. In this case, you can perform an adjustment if you want to improve the output accuracy or reset the calibration interval. However, if some output errors determined during verification do not fall within the published specifications, you must perform an adjustment to restore the device operation to its published specifications.

The *Complete Calibration* section describes the recommended calibration procedure. The *Optional Calibration* section describes alternative procedures that allow you to skip adjustment if the device already meets its calibration test limits or published specifications.

Complete Calibration

Performing a complete calibration is the recommended method of calibration, as it guarantees that the NI 5402/5406 meets or exceeds its published specifications for a two-year calibration interval. At the end of the complete calibration procedure, you verify that the output error falls within the calibration test limits. Figure 4 shows the programming flow for complete calibration.

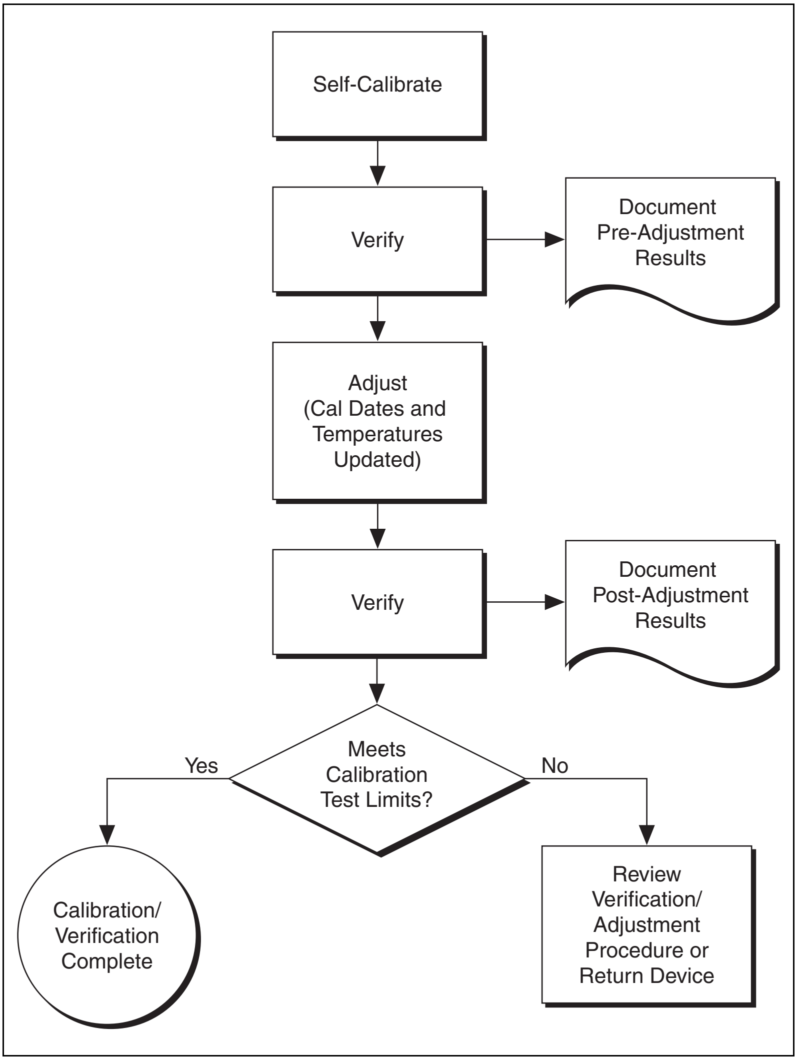


Figure 4. Complete Calibration Programming Flow

Optional Calibration

You can choose to skip the adjustment steps of the calibration procedure if the output error is within the calibration test limits or the published specifications during the first verification. If all of the output errors determined during the first verification fall within the calibration test limits, the device is guaranteed to meet or exceed its published specifications for a full calibration interval. In this case, you can update the calibration date, effectively resetting the calibration interval, without actually performing an adjustment. Refer to the *Updating the Calibration Date and Temperature* section for more information about this process.

If all of the output errors determined during the first verification fall within the published specifications, but not within the calibration test limits, adjustment is also optional. However, you cannot update the calibration date because the device will not necessarily operate within the published specifications for an additional two years.

Note Regardless of the results of the first verification, if you choose to perform an adjustment you must verify that the output error falls within the calibration test limits at the end of the calibration procedure.

Figure 5 shows the programming flow for the optional calibration.

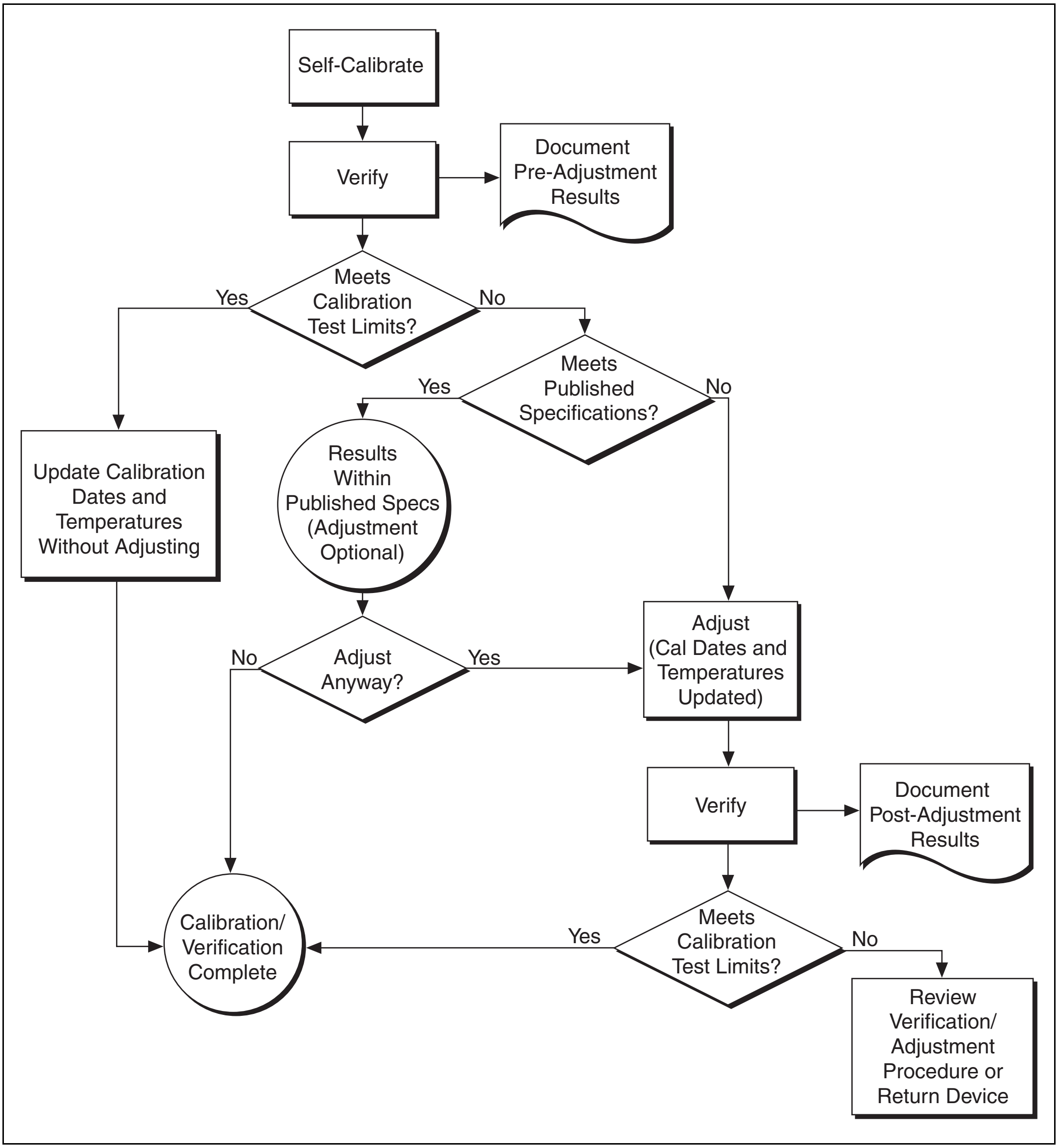


Figure 5. Optional Calibration Programming Flow

Appendix B: Calibration Utilities

NI-FGEN supports several calibration utilities that allow you to retrieve information about adjustments performed on the NI 5402/5406 arbitrary function generator, restore an external calibration, change the external calibration password, and store small amounts of information in the onboard EEPROM. You can retrieve some data using MAX or the FGEN SFP; however, you can retrieve all the data using NI-FGEN.

MAX

To retrieve data using MAX, complete the following steps:

1. Launch MAX.
2. Navigate to **My System»Devices and Interfaces»NI-DAQmx Devices** and select the device from which you want to retrieve information.
3. Select the **Calibration Tab** on the lower right corner. You should see information about the last calibration dates and temperature for both external and self-calibration.

FGEN SFP

To retrieve data using the FGEN SFP, complete the following steps:

1. Launch the FGEN SFP.
2. Navigate to **Edit»Device Configuration** and select the device from which you want to retrieve information using the Device Configuration dialog box.
3. Navigate to **Edit»Device ConfigurationUtility»Calibration** to open the Calibration dialog box. You should see information about the last calibration dates for both external and self-calibration.

NI-FGEN

NI-FGEN provides a full complement of calibration utility VIs and functions. Refer to the *NI Signal Generators Help* for the complete VI and function references. The following are the niFgen utility VIs:

- niFgen Get Self Cal Supported
- niFgen Restore Last Ext Cal Constants
- niFgen Get Ext Cal Recommended Interval
- niFgen Get Self Cal Last Date and Time
- niFgen Get Self Cal Last Temp
- niFgen Read Current Temp
- niFgen Get Ext Cal Last Date and Time
- niFgen Get Ext Cal Last Temp
- niFgen Get Cal User Defined Info
- niFgen Set Cal User Defined Info
- niFgen Change Ext Cal Password

The following are the niFgen utility functions:

- `niFgen_GetSelfCalSupported`
- `niFgen_GetSelfCalLastDateAndTime`
- `niFgen_GetExtCalLastDateAndTime`
- `niFgen_GetSelfCalLastTemp`
- `niFgen_GetExtCalLastTemp`
- `niFgen_GetExtCalRecommendedInterval`
- `niFgen_ChangeExtCalPassword`
- `niFgen_SetCalUserDefinedInfo`
- `niFgen_GetCalUserDefinedInfo`
- `niFgen_GetCalUserDefinedInfoMaxSize`
- `niFgen_ReadCurrentTemperature`
- `niFgen_RestoreLastExtCalConstants`

Where to Go for Support

The National Instruments Web site is your complete resource for technical support. At `ni.com/support` you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world to help address your support needs. For telephone support in the United States, create your service request at `ni.com/support` and follow the calling instructions or dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 1800 300 800, Austria 43 662 457990-0,
Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599,
Canada 800 433 3488, China 86 21 5050 9800,
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00,
Finland 385 (0) 9 725 72511, France 01 57 66 24 24,
Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737,
Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400,
Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710,
Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466,
New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60,
Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851,
Singapore 1800 226 5886, Slovenia 386 3 425 42 00,
South Africa 27 0 11 805 8197, Spain 34 91 640 0085,
Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151,
Taiwan 886 02 2377 2222, Thailand 662 278 6777,
Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

372135A-01 Jun07