

CALIBRATION PROCEDURE

NI PXIe-5442

This document contains instructions for calibrating the National Instruments PXIe-5442 (NI 5442) arbitrary waveform generator. This calibration procedure is intended for metrology labs. It describes specific programming steps for writing an external calibration procedure for the NI 5442.

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

Contents

Software Requirements.....	2
Documentation Requirements	2
Password.....	3
Calibration Interval.....	3
Test Equipment.....	3
Test Conditions.....	4
Calibration Procedures	4
Initial Setup.....	4
Self-Calibration	5
MAX.....	5
FGEN Soft Front Panel.....	6
NI-FGEN	6
Verification.....	6
Verifying the Oscillator Frequency Accuracy.....	8
Verifying the DC Gain and Offset Accuracy	10
Verifying the AC Voltage Amplitude Absolute Accuracy.....	20
Verifying Frequency Response (Flatness).....	23
Updating the Calibration Date and Temperature.....	30
Adjustment.....	30
Initializing the External Calibration Session	31
Adjusting the Analog Output.....	31
Adjusting the Oscillator Frequency.....	45
Adjusting the Calibration ADC	48
Closing the External Adjustment Session	52
Appendix A: Calibration Procedure Options.....	53
External Calibration.....	53
Complete Calibration.....	54
Optional Calibration	54

Appendix B: Calibration Utilities	56
MAX	56
FGEN SFP	56
NI-FGEN	57
Where to Go for Support	58

Software Requirements

Calibrating the NI 5442 requires installing NI-FGEN version 2.5 or later on the calibration system. You can download the NI-FGEN instrument driver from the Instrument Driver Network Web site at ni.com/idnet. NI-FGEN supports programming a [Self-Calibration](#) and an [External Calibration](#) in the LabVIEW, LabWindows™/CVI™, and C or 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 users, 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 is listed in the instructions. Tables for each step show both the configuration of the VI or property and the C function. The C function calls are valid for supported C or C++ compilers. Many of the functions use constants defined in the `niFgen.h` file. To use these constants in C or 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* and other signal generators documentation at **Start»All Programs»National Instruments»NI-FGEN»Documentation**.

Documentation Requirements

For information about NI-FGEN and the NI 5442, you can consult the following documents:

- *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 5442 and the NI-FGEN VIs and functions.
- *NI 5442 Specifications*—provides the published specification values for the NI 5442.

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

Password

"NI" is the default password for password-protected operations on your device. 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 about 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 a list of equipment.

Table 1. Equipment Required for Calibrating the NI 5442

Instrument	Applicable Specification	Minimum Specifications	Recommended Instruments
Digital multimeter (DMM)	DC gain and offset	DC accuracy $\leq \pm 50$ ppm Resolution $\leq 1 \mu\text{V}$	NI PXI-4070 Agilent/HP 34401A Keithley 2000
Banana(m)-to-BNC(f) adapter		—	—
BNC(m)-to-SMB(f) cable		50 Ω , RG-223	—
Spectrum analyzer or frequency meter	Frequency accuracy	Ability to measure 10 MHz or higher sine waves Frequency accuracy to ± 500 ppb	NI PXI-5660 Agilent/HP 8560E Agilent/HP 53131A or HP 53132A with timebase option 001, 010, or 012
Power meter/sensor	Frequency response (flatness)	VSWR: (50 kHz to 120 MHz) ≤ 1.11 Relative power accuracy: ≤ 0.022 dB	R&S NRP-Z91
Type N(f)-to-SMB plug adapter		VSWR: 1.3	Pasternak PE9316

Test Conditions

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

- Keep connections to the NI 5442 short. Long cables and wires act as antennae, picking up noise that can affect measurements.
- Keep relative humidity between 10 and 90% noncondensing.
- Maintain a temperature between 18 and 28 °C.
- Allow a warm-up time of at least 15 minutes after powering on all hardware, loading the operating system, and, if necessary, enabling the device. Unless manually disabled, the NI-FGEN driver automatically loads with the operating system and enables the device. The warm-up time brings the measurement circuitry of the NI 5442 to a stable operating temperature.
- Ensure that the PXI Express chassis fan speed is set to HI, that the fan filters are clean, and that the empty slots contain filler panels.
- Plug the PXI Express chassis and the calibrator into the same power strip to avoid ground loops.

Calibration Procedures

The calibration process has the following steps:

1. *Initial Setup*—Configure the 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 adjustment.
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.



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 MAX to communicate with NI-FGEN.

Complete the following steps to configure a device in MAX.

1. Install NI-FGEN.
2. Power off the computer or chassis that will hold the device and install the device in an available slot. Refer to the *NI Signal Generators Getting Started Guide* for information about installing the hardware.

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 5442 can perform self-calibration, which adjusts the gain and offset voltage of the analog path. Self-calibration uses an onboard analog-to-digital converter (ADC) to measure the output voltage.



Note You can calibrate the oscillator frequency and the calibration ADC only through an external adjustment procedure.

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



Note Running self-calibration overwrites the analog output constants measured in external calibration, using the onboard ADC to adjust for changes in temperature and other conditions that may have occurred since the external calibration. Calling the `niFgen Restore Last Ext Cal Constants VI` or the `niFgen_RestoreLastExtCalConstants` function reverts the device to the analog-output constants from the external calibration.

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 NI 5442 to self-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 listbox.

FGEN Soft Front Panel

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

1. Launch the FGEN SFP.
2. Select **Edit»Device Configuration** to launch the Device Configuration dialog box.
3. Select the device that you want to calibrate from the Device drop-down listbox. Click **OK** when finished.
4. Select **Utility»Calibration** to launch the Calibration dialog box.
5. Click **Perform self-calibration**.

NI-FGEN

Complete the following steps to programmatically perform a self-calibration on the NI 5442 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 Call `niFgen_init` (niFgen Initialize VI) to open 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. Use 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`



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

2. Call `niFgen_SelfCal` (niFgen Self Cal VI) to initiate self-calibration using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
3. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses. Use the following parameter:
 - **vi**: The session handle returned from `niFgen_init`

Verification

This section provides instructions for verifying the NI 5442 specifications. Refer to Table 1 for recommendations about choosing an instrument for each test.



Note Always self-calibrate the NI 5442 before beginning a verification procedure.

The steps in the verification procedures describe the code that you use to generate the appropriate signals, as well as the NI-FGEN function calls that you make to verify specifications.

You can verify the following specifications for the NI 5442:

- Oscillator frequency accuracy
- DC gain and offset accuracy
- AC amplitude
- Flatness

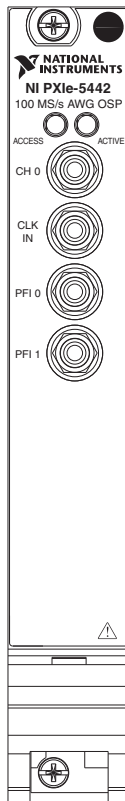
The verification procedure for each of these specifications includes setting up, programming, and closing the application.



Note If any of these tests fail immediately after you perform an external adjustment, verify that you have met the required test conditions before you return the NI 5442 to NI for repair.

Refer to Figure 1 for the names and locations of the NI PXIe-5442 front panel connectors. You can find information about the functions of these connectors in the *NI Signal Generators Getting Started Guide*.

Figure 1. NI PXIe-5442 Front Panel Connectors



Verifying the Oscillator Frequency Accuracy

This test verifies the frequency accuracy of the oscillator on the NI 5442. The verification involves generating a 10 MHz sine wave with the NI 5442 and measuring the sine wave frequency with one of the instruments from Table 1.

To verify the frequency accuracy of the oscillator on the NI 5442, complete the following steps:

1. Connect the NI 5442 CH 0 front panel connector to the instrument measuring the frequency accuracy with a male BNC-to-female SMB cable.
2. Call `niFgen_init` (niFgen Initialize VI) 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`
 - **vi**: A pointer to a ViSession. The variable passed by reference through this parameter receives the value that identifies the session created by this function. This value acts as the session handle and is passed as the first parameter to all subsequent NI-FGEN functions.
3. Call `niFgen_SetAttributeViReal64` to set the sample rate (niFgen Set Sample Set Sample Rate VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: `"0"`
 - **attributeId**: `NIFGEN_ATTR_ARB_SAMPLE_RATE`
 - **value**: `100000000`
4. Call `niFgen_SetAttributeViReal64` to set the gain (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Gain**) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: `"0"`
 - **attributeId**: `NIFGEN_ATTR_ARB_GAIN`
 - **value**: `1`



Note You can adjust the gain value based on which measuring device you use.

5. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Offset**) to set the offset using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: `"0"`
 - **attributeId**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: `0`



Note You can adjust the offset value based on which measuring device you use.

6. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to set the analog filter state using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_TRUE`
7. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Digital Filter Enabled**) to enable the digital filter state using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
 - **value**: `VI_TRUE`
8. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Output Attributes»Digital Filter Interpolation Factor**) to set the digital filter interpolation factor using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_DIGITAL_FILTER_INTERPOLATION_FACTOR`
 - **value**: 4
9. Generate an array of waveform samples. The waveform should have 10 samples per cycle with a total of 500 samples and 50 sine wave cycles. The 100 MS/s sample rate with 10 samples per cycle results in a 10 MHz sine wave waveform.



Note The sample values in this waveform must fall between -1.0 and 1.0 (representation: double).

10. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) instance of the niFgen Create Waveform (poly) VI) to create an arbitrary waveform using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **numberOfSamples**: The size in samples (500) of the waveform you created in step 9
 - **wfmData[]**: The array of waveform samples that you created in step 9
11. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`

12. Measure the frequency of the NI 5442 output signal.

A frequency error of 45 Hz for a 10 MHz signal corresponds to an error of 4.5 ppm. This limit accounts for the initial accuracy and the frequency deviation caused by temperature and aging. Refer to Table 2 for frequency ranges.

Table 2. Frequency Ranges

Calibration Test Limit		Published Specifications ± 25 ppm	
Low (Hz)	High (Hz)	Low (Hz)	High (Hz)
9,999,955	10,000,045	9,999,750	10,000,250

13. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:

- **vi**: The session handle returned from `niFgen_init`

14. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses. Use the following parameter:

- **vi**: The session handle returned from `niFgen_init`

Verifying the DC Gain and Offset Accuracy

This test verifies the DC gain and offset accuracy of the NI 5442 into a high-impedance load by generating a number of DC voltages and offsets, measuring the voltages with a DMM, and comparing the NI 5442 results to the error limits.

The DC gain and offset accuracy verification procedure has three subprocedures that verify the following specifications:

- Main analog path gain
- Main analog path offset
- Direct path gain

Verifying the Main Analog Path Gain

To verify the gain of the NI 5442 Main analog path, complete the following steps:

1. Connect the NI 5442 CH 0 front panel connector to a DMM to measure DC gain and offset accuracy.
2. Call `niFgen_init` (niFgen Initialize VI) to 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. Use 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. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes» Analog Filter Enabled**) to disable the analog filter state using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_FALSE`
4. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Output Attributes» Load Impedance**) to set the load impedance using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_LOAD_IMPEDANCE`
 - **value**: `10000000000`
5. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes» Analog Path**) to set the main analog path 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. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Basic Operation» Output Impedance**) to set the output impedance using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
 - **value**: `50.00`
7. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to enable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`
8. Create an array of waveform samples for the positive full-scale DC waveform. This array should contain 500 samples with each sample having the value 1.0 (representation: double).
9. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) instance of the Create Arbitrary Waveform (poly) VI) to create an arbitrary waveform using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"

- **numberOfSamples**: The size in samples (500) of the waveform you created in step 8
 - **wfmData[]**: The array of waveform samples that you created in step 8
10. Create an array of waveform samples for the negative full-scale DC waveform. This array should contain 500 samples with each sample having the value -1.0 (representation: double).
 11. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) instance of the Create Arbitrary Waveform (poly) VI) to create an arbitrary waveform using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **numberOfSamples**: The size in samples (500) of the waveform that you created in step 10
 - **wfmData[]**: The array of waveform samples that you created in step 10
 12. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output**»**Arbitrary Waveform Offset**) to set the arbitrary waveform offset using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: 0

Steps 13 through 23 use the values listed in Table 3. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 3. Values for Verifying the Gain of the Main Analog Path

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	Calibration Test Limit (Volts)	Published Specification (Volts)
1	2.000000	2.000000	-2.000000	± 0.003700	± 0.008500
2	1.650000	1.650000	-1.650000	± 0.003140	± 0.007100
3	1.250000	1.250000	-1.250000	± 0.002500	± 0.005500
4	0.850000	0.850000	-0.850000	± 0.001860	± 0.003900
5	0.600000	0.600000	-0.600000	± 0.001460	± 0.002900
6	0.415000	0.415000	-0.415000	± 0.001164	± 0.002160
7	0.300000	0.300000	-0.300000	± 0.000980	± 0.001700
8	0.205000	0.205000	-0.205000	± 0.000828	± 0.001320
9	0.150000	0.150000	-0.150000	± 0.000740	± 0.001100

Table 3. Values for Verifying the Gain of the Main Analog Path (Continued)

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	Calibration Test Limit (Volts)	Published Specification (Volts)
10	0.105000	0.105000	-0.105000	±0.000668	±0.000920
11	0.075000	0.075000	-0.075000	±0.000620	±0.00080
12	0.055000	0.055000	-0.055000	±0.000588	±0.000720
13	0.037500	0.037500	-0.037500	±0.000560	±0.000650
14	0.026000	0.026000	-0.026000	±0.000542	±0.000604
15	0.018500	0.018500	-0.018500	±0.000530	±0.000574
16	0.013000	0.013000	-0.013000	±0.000521	±0.000552
17	0.009000	0.009000	-0.009000	±0.000514	±0.000536
18	0.006500	0.006500	-0.006500	±0.000510	±0.000526

Notes: *Error Positive Full-Scale Value = (Measured Positive Full-Scale Value) - (Ideal Positive Full-Scale Value)*
Error Negative Full-Scale Value = (Measured Negative Full-Scale Value) - (Ideal Negative Full-Scale Value)

13. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Gain**) to set the arbitrary waveform gain using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_GAIN`
 - **value:** The *Gain* value listed in Table 3 for the current iteration
14. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value:** The **waveformHandle** from step 9 (positive full-scale handle)
15. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
16. Measure the NI 5442 DC output voltage. This voltage is the *Measured Positive Full-Scale Value*.

17. Determine the error for positive full scale using the following formula:

$$\text{Error Positive Full-Scale} = (\text{Measured Positive Full-Scale Value}) - (\text{Ideal Positive Full-Scale Value})$$

Compare this error to the *Published Specification* or the *Calibration Test Limit* for the current iteration from Table 3. Refer to [Appendix A: Calibration Procedure Options](#) for information about the uses of the published specifications and the calibration test limits.

18. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
19. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value**: The **waveformHandle** from step 11 (negative full-scale handle)
20. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
21. Measure the NI 5442 DC output voltage. This voltage is the *Measured Negative Full-Scale Value*.
22. Determine the error for negative full scale using the following formula:

$$\text{Error Negative Full-Scale} = (\text{Measured Negative Full-Scale Value}) - (\text{Ideal Negative Full-Scale Value})$$

Compare this error to the *Published Specification* or the *Calibration Test Limit* for the current iteration from Table 3. Refer to [Appendix A: Calibration Procedure Options](#) for information about the uses of the published specifications and the calibration test limits.

23. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
24. If any of the errors are greater than the *Calibration Test Limit*, perform an external adjustment.

Verifying the Main Analog Path Offset

To continue the verification of the DC Gain and Offset accuracy, verify the offset of the NI 5442 Main analog path by completing the following steps:

1. Create an array of waveform samples for the mid-scale DC waveform (0 VDC). This array should contain 500 samples with each sample having the value 0.0 (representation: double).

2. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) instance of the niFgen Create Waveform (poly) VI) to create an arbitrary waveform using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **numberOfSamples**: The size in samples (500) of the waveform that you created in step 1
 - **wfmData[]**: The array of waveform samples that you created in step 1
3. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value**: The **waveformHandle** from step 2 (mid-scale handle)

Steps 4 through 14 use the values listed in Table 4. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 4. Values for Verifying the Offset of the Main Analog Path

Iteration	Gain	Ideal Positive Offset (Volts)	Ideal Negative Offset (Volts)	Calibration Test Limit (Volts)	Published Specification (Volts)
1	2.000000	1.000000	-1.000000	±0.004000	±0.009000
2	1.650000	0.825000	-0.825000	±0.003388	±0.007513
3	1.250000	0.625000	-0.625000	±0.002688	±0.005813
4	0.850000	0.425000	-0.425000	±0.001988	±0.004113
5	0.600000	0.300000	-0.300000	±0.001550	±0.003050
6	0.415000	0.207500	-0.207500	±0.001226	±0.002264
7	0.300000	0.150000	-0.150000	±0.001025	±0.001775
8	0.205000	0.102500	-0.102500	±0.000859	±0.001371
9	0.150000	0.075000	-0.075000	±0.000763	±0.001138
10	0.105000	0.052500	-0.052500	±0.000684	±0.000946
11	0.075000	0.037500	-0.037500	±0.000631	±0.000819
12	0.055000	0.027500	-0.027500	±0.000596	±0.000734
13	0.037500	0.018750	-0.018750	±0.000566	±0.000659

Table 4. Values for Verifying the Offset of the Main Analog Path (Continued)

Iteration	Gain	Ideal Positive Offset (Volts)	Ideal Negative Offset (Volts)	Calibration Test Limit (Volts)	Published Specification (Volts)
14	0.026000	0.013000	-0.013000	±0.000546	±0.000611
15	0.018500	0.009250	-0.009250	±0.000532	±0.000579
16	0.013000	0.006500	-0.006500	±0.000523	±0.000555
17	0.009000	0.004500	-0.004500	±0.000516	±0.000538
18	0.006500	0.003250	-0.003250	±0.000511	±0.000528

Notes: *Error Positive Offset Value = (Measured Positive Offset Value) - (Ideal Positive Offset Value)*
Error Negative Offset Value = (Measured Negative Offset Value) - (Ideal Negative Offset Value)

4. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Offset**) to set the arbitrary waveform offset using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_OFFSET`
 - **value:** The *Ideal Positive Offset* value listed in Table 4 for the current iteration
5. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Gain**) to set the arbitrary waveform gain using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_GAIN`
 - **value:** The *Gain* value listed in Table 4 for the current iteration
6. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
7. Measure the positive NI 5442 DC output voltage. This value is the *Measured Positive Offset Value*.
8. Determine the error for positive offset using the following formula:

$$\text{Error Positive Offset} = (\text{Measured Positive Offset Value}) - (\text{Ideal Positive Offset Value})$$

Compare this error to the *Published Specification* or the *Calibration Test Limit* for the current iteration from Table 4. Refer to [Appendix A: Calibration Procedure Options](#) for information about the uses of the published specifications and the calibration test limits.

9. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - The session handle returned from `niFgen_init`
10. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Offset**) to set the arbitrary waveform offset using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: The *Ideal Negative Offset* value listed in Table 4 for the current iteration
11. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
12. Measure the negative NI 5442 DC output voltage. This value is the *Measured Negative Offset Value*.
13. Determine the error for negative offset using the following formula:

$$\text{Error Negative Offset} = (\text{Measured Negative Offset Value}) - (\text{Ideal Negative Offset Value})$$

Compare this error to the *Published Specification* or the *Calibration Test Limit* for the current iteration from Table 4. Refer to [Appendix A: Calibration Procedure Options](#) for information about the uses of the published specifications and the calibration test limits.

14. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
15. If any of the errors are greater than the *Calibration Test Limit*, perform an external adjustment.

Verifying the Gain of the Direct Path

To continue the verification of the DC Gain and Offset accuracy, verify the gain of the NI 5442 Direct path by completing the following steps:

1. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output>Arbitrary Waveform Offset**) to set the arbitrary waveform offset using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: 0

2. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the direct analog path using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ANALOG_PATH`
 - **value**: `NIFGEN_VAL_DIRECT_ANALOG_PATH`

Steps 3 through 14 use the values listed in Table 5. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 5. Values for Verifying the Gain of the Direct Analog Path

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	Offset Limit (Volts)	Calibration Test Limit (Volts)	Published Specification (Volts)
1	1.000000	1.000000	-1.000000	±0.025000	±0.001600	±0.004000
2	0.950000	0.950000	-0.950000	±0.025000	±0.001520	±0.003800
3	0.900000	0.900000	-0.900000	±0.025000	±0.001440	±0.003600
4	0.850000	0.850000	-0.850000	±0.025000	±0.001360	±0.003400
5	0.800000	0.800000	-0.800000	±0.025000	±0.001280	±0.003200
6	0.750000	0.750000	-0.750000	±0.025000	±0.001200	±0.003000
7	0.710000	0.710000	-0.710000	±0.025000	±0.001136	±0.002840

Notes: $Offset = ((Measured\ Positive\ Full-Scale\ Value) + (Measured\ Negative\ Full-Scale\ Value))/2$
 $Error\ Positive\ Full-Scale\ Value = (Measured\ Positive\ Full-Scale\ Value) - Offset - (Ideal\ Positive\ Full-Scale\ Value)$
 $Error\ Negative\ Full-Scale\ Value = (Measured\ Negative\ Full-Scale\ Value) - Offset - (Ideal\ Negative\ Full-Scale\ Value)$

3. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Gain**) to set the arbitrary waveform gain using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_GAIN`
 - **value**: The *Gain* value listed in Table 5 for the current iteration
4. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"

- **attributeId:** NIFGEN_ATTR_ARB_WAVEFORM_HANDLE
 - **value:** The **waveformHandle** from step 9 of the *Verifying the Main Analog Path Gain* section (positive full-scale handle)
5. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
 6. Measure the NI 5442 positive DC output voltage. This value is the *Measured Positive Full-Scale Value*.
 7. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
 8. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_ARB_WAVEFORM_HANDLE
 - **value:** The **waveformHandle** from step 11 of the *Verifying the Main Analog Path Gain* section (negative full-scale handle)
 9. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
 10. Measure the NI 5442 negative DC output voltage. This value is the *Measured Negative Full-Scale Value*.
 11. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
 12. Average the *Measured Positive Full-Scale Value* and *Measured Negative Full-Scale Value* to calculate the *Offset*.
 13. Verify that the *Offset* is less than or equal to the *Offset Limit* listed in Table 5 for the current iteration.
 14. Subtract the *Offset* and the *Ideal Full-Scale Value* from the *Measured Full-Scale Value* to get the *Error Full-Scale Value* for both the positive and negative settings, respectively.
 15. If any of the errors are greater than the *Calibration Test Limit* listed in Table 5, perform an external adjustment.
 16. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources NI-FGEN uses. Use the following parameter:
 - **vi:** The session handle returned from `niFgen_init`



Note The offset is not adjustable for the Direct path.

Verifying the AC Voltage Amplitude Absolute Accuracy

This test verifies the AC voltage amplitude absolute accuracy of the NI 5442 using a DMM. To verify the AC accuracy of the NI 5442, complete the following steps:

1. Connect the NI 5442 CH 0 front panel connector to the DMM. Connect positive terminal to the center pin of the NI 5442 SMB connector, and connect the negative terminal to the shield.
2. Call `niFgen_init` (niFgen Initialize VI) using the following parameters:
 - **resourceName**: The name of the device that you want to verify. You can find this name under Devices and Interfaces in MAX.
 - **IDQuery**: `VI_TRUE`
 - **resetDevice**: `VI_TRUE`
 - **vi**: A pointer to a ViSession. The variable passed by reference through this parameter receives the value that identifies the session created by this function. This value acts as the session handle and is passed as the first parameter to all subsequent NI-FGEN functions.
3. Call `niFgen_ConfigureChannels` (niFgen Configure Channels VI) using the following parameters:
 - **Channels**: `"0"`
 - **vi**: The session handle returned from `niFgen_init`
4. Call `niFgen_ConfigureSampleRate` (niFgen Set Sample Rate VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **Sample Rate**: `100 Ms/s (100000000)`
5. Call `niFgen_SetAttributeViReal64` to set the load impedance (NI-FGEN Load Impedance property) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: `"0"`
 - **attributeID**: `NIFGEN_ATTR_LOAD_IMPEDANCE`
 - **value**: `1000000`
6. Call `niFgen_ConfigureOutputMode` (niFgen Configure Output Mode VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **Output Mode**: `NIFGEN_VAL_OUTPUT_ARB` (Arbitrary Waveform)
7. Create an array of waveform samples. The waveform should have one sine wave cycle with an amplitude of 1 and a length of 2,000 samples.
8. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: `"0"`
 - **wfmSize**: The size in samples (`2000`) of the waveform.

- **wfmData**: The array of waveform samples (double representation).
 - **wfmHandle**: A pointer to a waveform. The variable passed by reference through this parameter acts as a handle to the waveform and can be used for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.
9. **(LabVIEW only)** You must call the niFgen Util Create Waveform Data VI to generate a single cycle sine wave with 2,000 samples and an amplitude of 1 for step 7. Wire the output of the niFgen Util Create Waveform Data VI to the **Waveform Data Array** input of the niFgen Create Waveform (DBL) VI in step 8.
10. Configure the DMM using the following settings:
- Function: AC voltage
 - Range: Refer to Table 6
 - Input impedance: 1 M Ω
 - Average readings: 4
 - Digits: 6.5



Note These values assume you are using an NI 4070 DMM. For other DMMs, use the range closest to the values listed in step 10. The input impedance should be equal to or greater than the values indicated in Table 1, [Equipment Required for Calibrating the NI 5442](#).

11. Repeat steps 12 through 18 for each of the 24 iterations listed in Table 6, changing the *Gain* and *DMM Range (VRMS)* values for each iteration.

Table 6. Values for Verifying the AC Voltage Amplitude Absolute Accuracy

Iteration	Gain	DMM Range (VRMS)	Expected Amplitude (VRMS)	Test Limit (-VRMS)	Test Limit (+VRMS)
1	2.000000	5	1.414214	-0.014849	0.028991
2	1.700000	5	1.202020	-0.012728	0.024749
3	1.400000	5	0.989949	-0.010607	0.020506
4	1.150000	5	0.813173	-0.008839	0.016971
5	1.000000	5	0.707107	-0.007778	0.014849
6	0.800000	0.5	0.565685	-0.006364	0.012021
7	0.600000	0.5	0.424264	-0.004950	0.009192
8	0.350000	0.5	0.247487	-0.003182	0.005657
9	0.300000	0.5	0.212132	-0.002828	0.004950
10	0.215000	0.5	0.152028	-0.002227	0.003748

Table 6. Values for Verifying the AC Voltage Amplitude Absolute Accuracy (Continued)

Iteration	Gain	DMM Range (VRMS)	Expected Amplitude (VRMS)	Test Limit (-VRMS)	Test Limit (+VRMS)
11	0.150000	0.5	0.106066	-0.001768	0.002828
12	0.100000	0.5	0.070711	-0.001414	0.002121
13	0.070000	0.5	0.049497	-0.001202	0.001697
14	0.050000	0.5	0.353550	-0.001061	0.001414
15	0.035000	0.5	0.024749	-0.000955	0.001202
16	0.025000	0.5	0.017678	-0.000884	0.001061
17	0.015000	0.5	0.010607	-0.000813	0.000919
18	0.010000	0.5	0.007071	-0.000778	0.000849
19	0.009000	0.5	0.006364	-0.000771	0.000834
20	0.006000	0.5	0.004243	-0.000750	0.000792

12. Call `niFgen_SetAttributeViReal64` (NI-FGEN Gain property) to set the gain using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ARB_GAIN`
 - **value**: The *Gain* value listed in Table 6 for the current iteration
13. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
14. Wait 5 seconds for the output of the NI 5442 to settle.
15. Measure and record the output voltage amplitude with the DMM. This value is the measured amplitude, *measuredVRMS*.
16. Calculate the peak amplitude error using the following equation:
$$expectedVRMS - measuredVRMS = error$$
17. Compare the output error to the test limits in Table 6 for the current iteration.
18. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the current generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
19. Return to step 12 until all iterations are completed.

20. If any of the errors are greater than the test limits, perform an external adjustment.
21. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources that NI-FGEN uses. Use the following parameter:
 - **vi**: The session handle returned from `niFgen_init`

Verifying Frequency Response (Flatness)

This test verifies the frequency response (flatness) of the NI 5442 using a power meter. The flatness verification has two subprocedures that verify the following:

- Main analog path flatness
- Direct analog path flatness

Verifying the Main Analog Path Flatness

To verify the main analog path flatness of the NI 5442, complete the following steps:

1. Connect the NI 5442 CH 0 front panel connector to the power meter using the required adapter.
2. Call `niFgen_init` (niFgen Initialize VI) using the following parameters:
 - **resourceName**: The name of the device that you want to verify. You can find this name under Devices and Interfaces in MAX.
 - **IDQuery**: `VI_TRUE`
 - **resetDevice**: `VI_TRUE`
 - **vi**: A pointer to a `ViSession`. The variable passed by reference through this parameter receives the value that identifies the session created by this function. This value acts as the session handle and is passed as the first parameter to all subsequent NI-FGEN functions.
3. Call `niFgen_SetAttributeViBoolean` (NI-FGEN Output Enabled property) to disable the NI 5442 output. Use the following parameters:
 - **channelName**: `"0"`
 - **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_FALSE`
 - **vi**: The session handle returned from `niFgen_init`
4. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
5. Null the power meter according to the power meter documentation.
6. Configure the power meter using the following settings:
 - Average: 16
 - Measure: Watts

7. Call `niFgen_SetAttributeViReal64` to set the offset (NI-FGEN Offset property) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: 0
8. Call `niFgen_SetAttributeViInt32` to set the main analog path (NI-FGEN Analog Path property) 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`
9. Call `niFgen_SetAttributeViReal64` to set the gain using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ARB_GAIN`
 - **value**: 1
10. Call `niFgen_SetAttributeViBoolean` (`niFgen` property node: **Output Attributes»Analog Filter Enabled**) to set the analog filter state using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_TRUE`
11. Call `niFgen_SetAttributeViBoolean` to set the digital filter state (`niFgen` property node: **Output Attributes»Digital Filter Enabled**) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
 - **value**: `VI_TRUE`
12. Call `niFgenSetAttributeViBoolean` (NI-FGEN Output Enabled property) to enable the NI 5442 output. Use the following parameters:
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`
 - **vi**: The session handle returned from `niFgen_init`

13. Call `niFgen_ConfigureSampleRate` (`niFgen Set Sample Rate VI`) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **Sample Rate**: 100 MS/s (100000000)
14. Repeat steps 15 through 24 for each iteration in Table 7, changing the *Number of Samples* and *Number of Cycles* for each iteration.

Table 7. NI 5442 Setup for Main Analog Path Flatness Verification

Iteration	Frequency	Number of Samples	Number of Cycles	Published Specification
1	50 kHz	2,000	1	REF
2	1 MHz	1,000	10	-0.4 dB to +0.6 dB
3	5 MHz	1,000	50	-0.4 dB to +0.6 dB
4	10 MHz	1,000	100	-0.4 dB to +0.6 dB
5	20 MHz	1,000	200	-0.4 dB to +0.6 dB
6	30 MHz	1,000	300	-0.4 dB to +0.6 dB
7	40 MHz	1,000	400	-0.4 dB to +0.6 dB
8	43 MHz	1,000	430	-3.0 dB to +0.6 dB

15. Create an array of waveform samples. Each waveform should have samples and cycles that correspond to the current iteration in Table 7.
16. (LabVIEW only) You must call the Sine Pattern VI to create an array of waveform samples with an **amplitude** of 1, and **samples** and **cycles** that correspond to the current iteration in Table 7 for step 15. Wire the **Sinusoidal Pattern** output of the Sine Pattern VI to the **Waveform Data Array** input of the `niFgen Create Waveform (DBL) VI` in step 17.
17. Call `niFgen_CreateWaveformF64` (`niFgen Create Waveform (DBL) VI`) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **wfmSize**: The size in samples of the waveform that you created in step 15
 - **wfmArray**: The array of waveform samples (double representation)
 - **wfmHandle**: A pointer to a waveform. The variable passed by reference through this parameter acts as a handle to the waveform and can be used for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.

18. Call `niFgen_SetAttributeViInt32` (NI-FGEN Arbitrary Waveform Handle property) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value**: The `wfmHandle` from step 17
19. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
20. Allow the power meter to stabilize for 10 seconds.
21. Measure and record the power (W_f) of the positive output in Watts. Use the recorded power at 50 kHz as reference power (W_{ref}).
22. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the current generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
23. For iterations 2-16 in Table 7, using the recorded power values, calculate the deviation from the reference (50 kHz) power using the following equation:

$$Flatness(dB) = 10 \log \left(\frac{W_f}{W_{ref}} \right)$$

24. Compare *Flatness (dB)* to the *Published Specification* for the current iteration in Table 7.
25. Return to step 15 until all iterations are completed.
26. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources that NI-FGEN uses. Use the following parameter:
 - **vi**: The session handle returned from `niFgen_init`

Verifying the Direct Analog Path Flatness

To verify the direct analog path flatness of the NI 5442, complete the following steps:

1. Connect the NI 5442 CH 0 front panel connector to the power meter using the required adapter.
2. Call `niFgen_init` (niFgen Initialize VI) using the following parameters:
 - **resourceName**: The name of the device that you want to verify. You can find this name under Devices and Interfaces in MAX.
 - **IDQuery**: `VI_TRUE`
 - **resetDevice**: `VI_TRUE`
 - **vi**: A pointer to a `ViSession`. The variable passed by reference through this parameter receives the value that identifies the session created by this function. This value acts as the session handle and is passed as the first parameter to all subsequent NI-FGEN functions.

3. Call `niFgenSetAttributeViBoolean` (NI-FGEN Output Enabled property) to disable the NI 5442 output. Use the following parameters:
 - **channelName:** "0"
 - **attributeID:** `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value:** `VI_FALSE`
 - **vi:** The session handle returned from `niFgen_init`
4. Call `niFgen_Commit` (`niFgen Commit VI`) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_init`
5. Null the power meter according to the power meter documentation.
6. Configure the power meter using the following settings:
 - Average: 16
 - Measure: Watts
7. Call `niFgen_SetAttributeViReal64` to set the offset (NI-FGEN Offset property) using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeID:** `NIFGEN_ATTR_ARB_OFFSET`
 - **value:** 0
8. Call `niFgen_SetAttributeViInt32` to set the analog path (NI-FGEN Analog Path property) using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeID:** `NIFGEN_ATTR_ANALOG_PATH`
 - **value:** `NIFGEN_VAL_DIRECT_ANALOG_PATH`
9. Call `niFgen_SetAttributeViReal64` to set the gain (NI-FGEN Gain property) using the following parameters:
 - **vi:** The session handle returned from `niFgen_Init`
 - **channelName:** "0"
 - **attributeID:** `NIFGEN_ATTR_ARB_GAIN`
 - **value:** 0.5
10. Call `niFgen_SetAttributeViBoolean` (`niFgen` property node: **Output Attributes»Analog Filter Enabled**) to set the analog filter state using the following parameters:
 - **vi:** The session handle returned from `niFgen_init`
 - **channelName:** "0"
 - **attributeID:** `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value:** `VI_TRUE`

11. Call `niFgen_SetAttributeViBoolean` to set the digital filter state (niFgen property node: **Output Attributes»Digital Filter Enabled**) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
 - **value**: `VI_TRUE`
12. Call `niFgenSetAttributeViBoolean` (NI-FGEN Output Enabled property) to enable the NI 5421/5441 output. Use the following parameters:
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`
 - **vi**: The session handle returned from `niFgen_init`
13. Call `niFgen_ConfigureSampleRate` (niFgen Set Sample Rate VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **Sample Rate**: 100 MS/s (100000000)
14. Repeat steps 14 through 22 for each iteration in Table 8, changing the *Number of Samples* and *Number of Cycles* for each iteration.

Table 8. NI 5442 Setup for Direct Analog Path Flatness Verification

Iteration	Frequency	Number of Samples	Number of Cycles	Published Specification
1	50 kHz	2,000	1	REF
2	1 MHz	1,000	10	-0.1 dB to +0.5 dB
3	5 MHz	1,000	50	-0.1 dB to +0.5 dB
4	10 MHz	1,000	100	-0.1 dB to +0.5 dB
5	20 MHz	1,000	200	-0.1 dB to +0.5 dB

15. Create an array of waveform samples. Each waveform should have samples and cycles that correspond to the current iteration in Table 8.
16. (LabVIEW only) You must call the Sine Pattern VI to create an array of waveform samples with an **amplitude** of 1, and **samples** and **cycles** that correspond to the current iteration in Table 8 for step 15. Wire the **Sinusoidal Pattern** output of the Sine Pattern VI to the **Waveform Data Array** input of the niFgen Create Waveform (DBL) VI in step 17.
17. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"

- **wfmSize**: The size in samples of the waveform that you created in step 15
 - **wfmArray**: The array of waveform samples (double representation)
 - **wfmHandle**: A pointer to a waveform. The variable passed by reference through this parameter acts as a handle to the waveform and can be used for setting the active waveform, changing the data in the waveform, building sequences of waveforms, or deleting the waveform when it is no longer needed.
18. Call `niFgen_SetAttributeViInt32` (NI-FGEN Arbitrary Waveform Handle property) using the following parameters:
 - **vi**: The session handle returned from `niFgen_init`
 - **channelName**: "0"
 - **attributeID**: `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value**: The **wfmHandle** from step 17
 19. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
 20. Allow the power meter to stabilize for 10 seconds.
 21. Measure and record the power (W_f) of the positive output in Watts. Use the recorded power at 50 kHz as reference power (W_{ref}).
 22. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the current generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_init`
 23. For iterations 2-12 in Table 8, using the recorded power values, calculate the deviation from the reference (50 kHz) power using the following equation:

$$Flatness(dB) = 10 \log\left(\frac{W_f}{W_{ref}}\right)$$

24. Compare *Flatness* (dB) calculated in step 23 to the *Published Specification* for the current iteration in Table 8.
25. Return to step 15 until all iterations are completed.
26. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, destroy the instrument driver session and all of its properties, and release any memory resources that NI-FGEN uses. Use the following parameter:
 - **vi**: The session handle returned from `niFgen_init`

Updating the Calibration Date and Temperature

If the NI 5442 passes 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. Call `niFgen_InitExtCal` (niFgen Init Ext Cal VI) to open an NI-FGEN external calibration session 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" (default)
2. Call `niFgen_CloseExtCal` (niFgen Close Ext Cal VI) to close the instrument driver session and save the calibration date and temperature to the onboard EEPROM. Use the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **action**: `NIFGEN_VAL_EXT_CAL_COMMIT`. This setting stores the date and temperature of the system at the time of calibration.

Adjustment

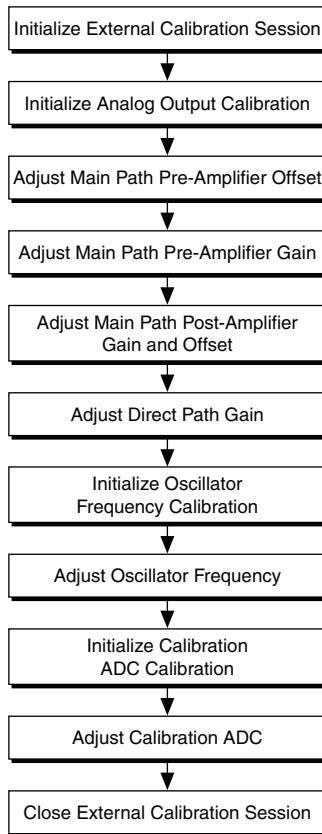
If the NI 5442 successfully passes all verification procedures within the calibration test limits, NI nevertheless recommends adjustment to guarantee its published specifications for the next two years. If you choose not to adjust your device, refer to the [Updating the Calibration Date and Temperature](#) section for instructions for updating the calibration date and temperature without performing an adjustment. If the NI 5442 is not within the calibration test limits for each verification procedure, perform the adjustment procedure to improve the accuracy of the NI 5442. Refer to [Appendix A: Calibration Procedure Options](#) for more information about which procedures to perform.

The external calibration procedure adjusts the analog output, the oscillator frequency, and the onboard calibration ADC. *Analog output adjustment* characterizes the DC gains and the offsets of the analog path to ensure the analog voltage accuracy. *Oscillator frequency adjustment* characterizes the onboard oscillator to ensure frequency accuracy. *Calibration ADC adjustment* characterizes the onboard ADC gain and offset so that self-calibration results in an accurately calibrated device.

To perform an adjustment, create an external calibration session by calling the niFgen Init Ext Cal VI or the `nFgen_InitExtCal` function. The external calibration session adjusts 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 or the `niFgen_CloseExtCal` function.

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

Figure 2. NI 5442 External Calibration Procedure Programming Flow



Initializing the External Calibration Session

Call `niFgen_InitExtCal` (`niFgen Init Ext Cal VI`) to open an NI-FGEN external calibration session 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" (default)

Adjusting the Analog Output

The analog output adjustment procedure has several subprocedures that adjust the following characteristics:

- Main path pre-amplifier offset
- Main path pre-amplifier gain

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

In each of these sub-procedures, the device is in several configurations and takes several output measurements. You then pass these measurements to NI-FGEN to determine the calibration constants for the device.

Initializing Analog Output Calibration

1. Call `niFgen_InitializeAnalogOutputCalibration` (niFgen Initialize Analog Output Calibration VI) to initialize analog output calibration using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
2. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 0
3. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the fixed Low-Gain analog path 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`
4. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 2,000
5. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value**: 32,767
6. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to disable the analog filter using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"

- **attributeId:** NIFGEN_ATTR_ANALOG_FILTER_ENABLED
 - **value:** VI_FALSE
7. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION
 - **value:** 0
 8. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Post-Amplifier Attenuation**) to set the post-amplifier attenuation using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION
 - **value:** 0
 9. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Basic Operation»Output Impedance**) to set the output impedance using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_OUTPUT_IMPEDANCE
 - **value:** 50.00
 10. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation»Output Enabled**) to enable the analog output using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_OUTPUT_ENABLED
 - **value:** VI_TRUE
 11. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`

Adjusting the Main Path Pre-amplifier Offset

1. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the fixed Low-Gain analog path 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

2. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration» Post-Amplifier Attenuation**) to set the post-amplifier attenuation using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
 - **value**: 0
3. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 0

Steps 4 through 7 use the values listed in Table 6. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 9. Attributes and Values for Main Path Pre-amplifier Offset

Iteration	Analog Filter Enable	Pre-amplifier Attenuation	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

4. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to set the analog filter 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 9
5. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation 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 9
6. Complete the following steps to take voltage measurements at the NI 5442 CH 0 front panel connector into a high-impedance load:
 - a. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 2000
 - b. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value**: 50000
 - c. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values 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 device output voltage. This measurement is `measurement 0`, which is used in step 7.
 - f. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 1000

- g. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device 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 device output voltage. This measurement is measurement 1, which is used in step 7.
 - j. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value**: 15000
 - k. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device 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 device output voltage. This measurement is measurement 2, which is used in step 7.
7. Call `niFgen_CalAdjustMainPathPreAmpOffset` (niFgen Cal Adjust Main Path Pre Amp Offset VI) to adjust the pre-amplifier Main path offset 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 9
 - **gainDACValues**: (2000, 1000)
 - **offsetDACValues**: (50000, 15000)
 - **measuredOutputs**: (measurement 0, measurement 1, measurement 2)

Adjusting the Main Path Pre-amplifier Gain

1. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the fixed Low-Gain analog path 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`

2. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Post-Amplifier Attenuation**) to set the post-amplifier attenuation using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeld:** `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
 - **value:** 0
3. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeld:** `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value:** 32000

Steps 4 through 7 use the values listed in Table 10. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 10. Attributes and Values for Main Path Pre-amplifier Gain

Iteration	Analog Filter Enable	Pre-amplifier Attenuation	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

Table 10. Attributes and Values for Main Path Pre-amplifier Gain (Continued)

Iteration	Analog Filter Enable	Pre-amplifier Attenuation	Current Configuration
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

4. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes» Analog Filter Enabled**) to set the analog filter state 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 10
5. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration» Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation 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
6. Complete the following steps to take voltage measurements at the NI 5442 CH 0 front panel connector into a high-impedance load:
 - a. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_GAIN_DAC_VALUE
 - **value:** 1500
 - b. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static VI) to set the main DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **value:** 25233
 - c. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values 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 device output voltage. This measurement is measurement 0, which is used in step 7.
 - f. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 2000
 - g. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: -29232
 - h. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - i. Wait 500 ms for the output to settle.
 - j. Use the DMM to measure the voltage output by the device. This measurement is measurement 1, which is used in step 7.
7. Call `niFgen_CalAdjustMainPathPreAmpGain` (niFgen Cal Adjust Main Path Pre Amp Gain VI) to adjust the pre-amplifier Main path gain and offset 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 10
 - **mainDACValues**: (25233, -29232)
 - **gainDACValues**: (1500, 2000)
 - **offsetDACValues**: (32000)
 - **measuredOutputs**: (measurement 0, measurement 1)

Adjusting the Main Path Post-amplifier Gain and Offset

1. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 0
2. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to disable the analog filter using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"

- **attributeId:** NIFGEN_ATTR_ANALOG_FILTER_ENABLED
 - **value:** VI_FALSE
3. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION
 - **value:** 0
 4. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_GAIN_DAC_VALUE
 - **value:** 2,000
 5. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the analog path 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

Steps 6 through 8 use the values listed in Table 8. Complete these steps for the row corresponding to Iteration 1, then repeat them for each of the remaining iterations.

Table 11. Attributes and Values for the Main Path Post-amplifier Gain and Offset

Iteration	Post-amplifier Attenuation	Current Configuration
1	0	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_0DB
2	12	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_12DB
3	24	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_24DB
4	36	NIFGEN_VAL_CAL_CONFIG_MAIN_PATH_LOW_GAIN_36DB

6. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Post-Amplifier Attenuation**) to set the post-amplifier attenuation 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 11
7. Complete the following steps to take voltage measurements at the NI 5442 CH 0 front panel connector into a high-impedance load:
- a. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_OFFSET_DAC_VALUE
 - **value:** 50,000
 - b. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values 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 device output voltage. This measurement is `measurement 0`, which is used in step 8.
 - e. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** NIFGEN_ATTR_OFFSET_DAC_VALUE
 - **value:** 15,000
 - f. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device 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 device output voltage. This measurement is `measurement 1`, which is used in step 8.
8. Call `niFgen_CalAdjustMainPathPostAmpGainAndOffset` (niFgen Cal Adjust Main Path Post Amp Gain And Offset VI) to adjust the post-amplifier Main path gain and offset 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 11
 - **mainDACValues:** {0, 0}
 - **gainDACValues:** {2000}
 - **offsetDACValues:** {50000, 15000}
 - **measuredOutputs:** {`measurement 0`, `measurement 1`}

Adjusting the Direct Path Gain

1. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 0
2. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 2,000
3. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value**: 32,767
4. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to disable the analog filter using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_FALSE`
5. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION`
 - **value**: 0
6. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Post-Amplifier Attenuation**) to set the post-amplifier attenuation using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **attributeId**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
 - **value**: 0

7. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Basic Operation» Output Impedance**) to set the output impedance using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
 - **value:** 50.00
8. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to enable the analog output using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value:** `VI_TRUE`
9. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes» Analog Path**) to select the Direct analog path using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ANALOG_PATH`
 - **value:** `NIFGEN_VAL_DIRECT_ANALOG_PATH`
10. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
11. Complete the following steps to take voltage measurements at the NI 5442 CH 0 front panel connector into a high-impedance load:
 - a. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **value:** 32767
 - b. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value:** 1800
 - c. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device 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 device output voltage. This measurement is measurement 0, which is used in step 12.
- f. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: NIFGEN_ATTR_GAIN_DAC_VALUE
 - **value**: 2600
- g. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device 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 device output voltage. This measurement is measurement 1, which is used in step 12.
- j. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: -32767
- k. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: NIFGEN_ATTR_GAIN_DAC_VALUE
 - **value**: 1500
- l. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
- m. Wait 500 ms for the output to settle.
- n. Use the DMM to measure the device output voltage. This measurement is measurement 2, which is used in step 12.
- o. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: NIFGEN_ATTR_GAIN_DAC_VALUE
 - **value**: 2300

- p. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - q. Wait 500 ms for the output to settle.
 - r. Use the DMM to measure the device output voltage. This measurement is `measurement 3`, which is used in step 12.
12. Call `niFgen_CalAdjustDirectPathGain` (niFgen Cal Adjust Direct Path Gain VI) to adjust the Direct path gain using the following parameters:
- **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **configuration:** (32767, -32767)
 - **gainDACValues:** (1800, 2600, 1500, 2300)
 - **offsetDACValues:** (50000, 15000)
 - **measuredOutputs:** (measurement 0, measurement 1, measurement 2, measurement 3)

Adjusting the Oscillator Frequency

Adjusting the oscillator frequency involves generating a sine wave at a desired frequency, and then iteratively measuring the frequency, passing the measured value to NI-FGEN so that the oscillator can be adjusted, and then remeasuring the resulting frequency. This process is repeated until the difference between the desired and measured frequency falls within the desired 4.5 ppm tolerance. The adjustment ensures the frequency accuracy of the onboard oscillator.

1. Call `niFgen_InitializeOscillatorFrequencyCalibration` (niFgen Initialize Oscillator Frequency Calibration VI) to initialize oscillator frequency calibration using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
2. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output**»**Sample Rate**) to set the sample rate using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_SAMPLE_RATE`
 - **value:** 100000000
3. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output**»**Arbitrary Waveform Gain**) to set the arbitrary waveform gain using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_ARB_GAIN`
 - **value:** 1



Note You can adjust this value based on which measurement device you use.

4. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Arbitrary Waveform Output»Arbitrary Waveform Offset**) to set the arbitrary waveform offset using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_OFFSET`
 - **value**: 0



Note You can adjust this value based on which measurement device you use.

5. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_TRUE`
6. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Digital Filter Enabled**) to enable the digital filter using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
 - **value**: `VI_TRUE`
7. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Output Attributes»Digital Filter Interpolation Factor**) to set the digital filter interpolation factor using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_DIGITAL_FILTER_INTERPOLATION_FACTOR`
 - **value**: 4
8. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Basic Operation»Output Impedance**) to set the output impedance using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
 - **value**: 50.00

9. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to enable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`
10. Generate an array of waveform samples.

Each waveform should have 10 samples per cycle, with a total of 500 samples and 50 sine wave cycles. The 100 MS/s sample rate with 10 samples per cycle results in a 10 MHz sine wave waveform.



Note The sample values of this waveform must fall between -1.0 and 1.0.k.

11. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) instance of the niFgen Create Waveform (poly) VI) to create an arbitrary waveform using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **numberOfSamples**: The size in samples (500) of the waveform you created in step 10
 - **wfmData[]**: The array of waveform samples that you created in step 10
12. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Arbitrary Waveform» Arbitrary Waveform Handle**) to set the waveform handle using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_ARB_WAVEFORM_HANDLE`
 - **value**: The **waveformHandle** from step 11 (sine waveform handle)
13. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
14. Measure the frequency of the generated waveform. This value is the measured frequency, which is used in step 15.
15. Repeat steps 15a through 15d for as long as the difference between the measured frequency and the desired frequency (10 MHz) is greater than the tolerance (4.5 ppm).

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 5442.

 - a. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`

- b. Call `niFgen_CalAdjustOscillatorFrequency` (niFgen Cal Adjust Oscillator Frequency VI) to adjust the oscillator frequency using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **desiredFrequencyInHz**: 10,000,000
 - **measuredFrequencyInHz**: The measured frequency value (in Hz)
 - c. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate waveform generation using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - d. Measure the frequency of the generated waveform. This value is the measured frequency.
16. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort waveform generation using the following parameter:
- **vi**: The session handle returned from `niFgen_InitExtCal`

Adjusting the Calibration ADC

The NI 5442 has an onboard calibration ADC that is used during self-calibration. Adjusting the calibration ADC involves characterizing the gain and offset associated with the ADC so that performing self-calibration results in an accurately calibrated device.

1. Call `niFgen_InitializeCalADCCalibration` (niFgen Initialize Cal ADC Calibration VI) to initialize ADC calibration using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
2. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 0
3. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Output Attributes»Analog Path**) to select the fixed Low-Gain analog path 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`
4. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Gain DAC Value**) to set the gain DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
 - **value**: 1700

5. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Offset DAC Value**) to set the offset DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
 - **value**: 32767
6. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Output Attributes»Analog Filter Enabled**) to disable the analog filter using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
 - **value**: `VI_FALSE`
7. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Pre-Amplifier Attenuation**) to set the pre-amplifier attenuation using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION`
 - **value**: 0
8. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Calibration»Post-Amplifier Attenuation**) to set the post-amplifier attenuation using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
 - **value**: 0
9. Call `niFgen_SetAttributeViReal64` (niFgen property node: **Basic Operation»Output Impedance**) to set the output impedance using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
 - **value**: 50.00
10. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation»Output Enabled**) to enable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeld**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`

11. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
12. Wait 500 ms for the output to settle.
13. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Cal ADC Input**) to set the calibration ADC input 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. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **value**: 27232
15. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation»Output Enabled**) to disable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_FALSE`
16. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
17. Wait 500 ms for the output to settle.
18. Call `niFgen_ReadCalADC` (niFgen Read CAL ADC VI) to measure the analog output voltage with the onboard calibration ADC 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 0, which is used in step 32.
19. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation»Output Enabled**) to enable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_TRUE`

20. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
21. Wait 500 ms for the output to settle.
22. Use the DMM to measure the NI 5442 output voltage directly into the DMM into a high-impedance load. This measurement is `external measurement 0`, which is used in step 32.
23. Call `niFgen_WriteBinary16AnalogStaticValue` (niFgen Write Binary 16 Analog Static Value VI) to set the main DAC value using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **value:** 10232
24. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to disable the analog output using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value:** `VI_FALSE`
25. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
26. Wait 500 ms for the output to settle.
27. Call `niFgen_ReadCalADC` (niFgen Read CAL ADC VI) to measure the analog output voltage with the onboard calibration ADC 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 32.
28. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to enable the analog output using the following parameters:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
 - **channelName:** "0"
 - **attributeId:** `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value:** `VI_TRUE`
29. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi:** The session handle returned from `niFgen_InitExtCal`
30. Wait 500 ms for the output to settle.

31. Use the DMM to measure the NI 5442 output voltage directly into the DMM (into a high-impedance load). This value is `external measurement 1`, which is used in step 32.
32. Call `niFgen_CalAdjustCalADC` (niFgen Cal Adjust Cal ADC VI) to adjust the ADC calibration using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **voltagesMeasuredExternally**: (`external measurement 0`, `external measurement 1`)
 - **voltagesMeasuredWithCalADC**: (`cal ADC measurement 0`, `cal ADC measurement 1`)
33. Call `niFgen_SetAttributeViBoolean` (niFgen property node: **Basic Operation» Output Enabled**) to disable the analog output using the following parameters:
 - **vi**: The session handle returned from `niFgen_InitExtCal`
 - **channelName**: "0"
 - **attributeId**: `NIFGEN_ATTR_OUTPUT_ENABLED`
 - **value**: `VI_FALSE`
34. Call `niFgen_SetAttributeViInt32` (niFgen property node: **Calibration»Cal ADC Input**) to set the calibration ADC input 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`
35. Call `niFgen_Commit` (niFgen Commit VI) to commit the attribute values to the device using the following parameter:
 - **vi**: The session handle returned from `niFgen_InitExtCal`

Closing the External Adjustment Session

When you have completed all the adjustment stages, you must close the external adjustment session to store the new calibration constants in the onboard EEPROM.

Call `niFgen_CloseExtCal` (niFgen Close Ext Cal VI) 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, and 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.

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. A properly verified 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 that determines whether an adjustment process should 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.

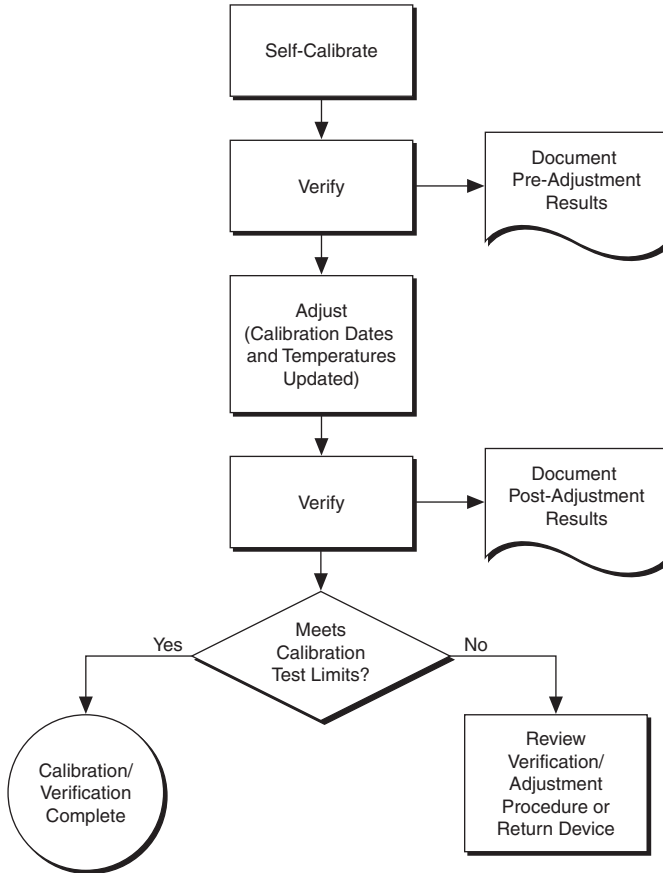
Published specification values are less restrictive than the calibration test limits. If all the output errors determined during verification fall within the published specifications, but not within the calibration test limits, the device currently meets its published specifications. The device will 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 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, 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 5442 meets or exceeds its published specifications for a two-year calibration interval. At the end of the complete calibration procedure, verify that the output error falls within the calibration test limits. Figure 3 shows the programming flow for complete calibration.

Figure 3. 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 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.

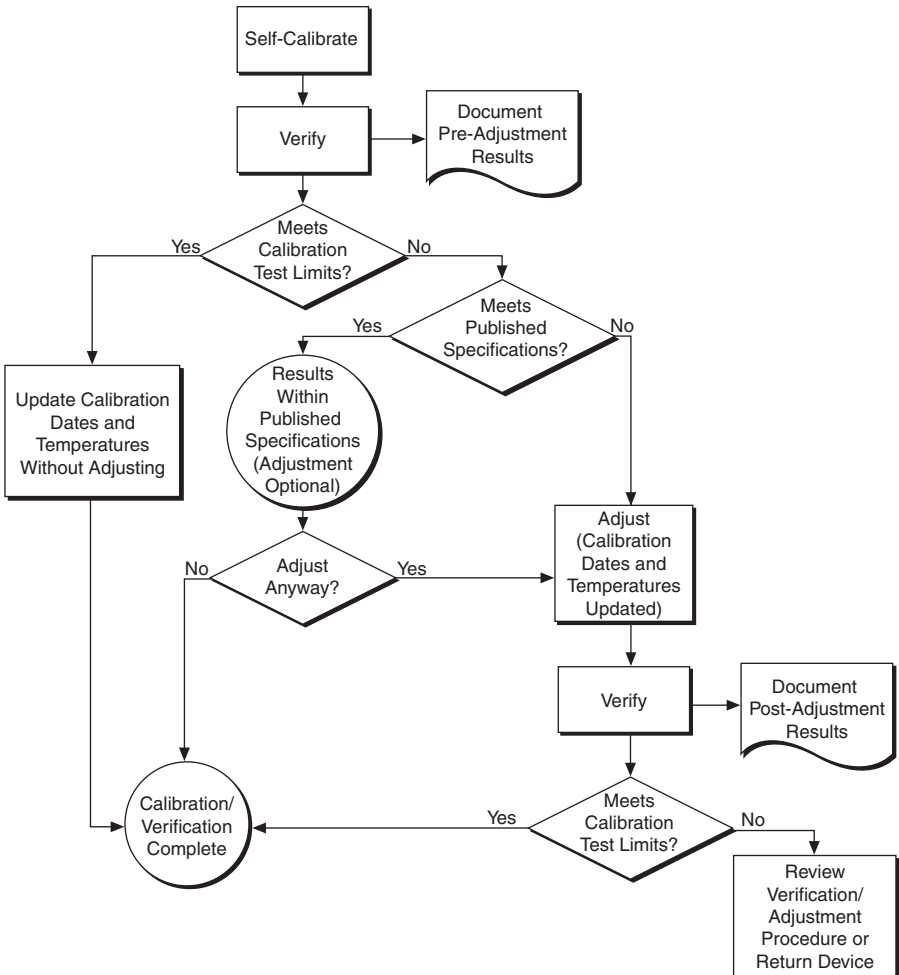
If all the output errors determined during the first verification fall within the published specifications, but not within the calibration test limits, adjustment is also optional. However, you cannot update the calibration date because the device will not necessarily operate within the published specifications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

Refer to Figure 4 for a visual representation of the programming flow for the optional calibration.

Figure 4. Optional Calibration Programming Flow



Appendix B: Calibration Utilities

NI-FGEN supports several calibration utilities that allow you to perform the following functions:

- Retrieve information about adjustments performed on the NI 5442.
- Restore an external calibration.
- Change the external calibration password.
- Store small amounts of information in the onboard EEPROM.



Note You can retrieve some data using MAX or the FGEN SFP; 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 Configuration Utility»Calibration** to launch 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 VIs are the niFgen Calibration 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 functions are the niFgen Calibration 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 website 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.

Visit ni.com/services for NI Factory Installation Services, repairs, extended warranty, and other services.

Visit ni.com/register to register your National Instruments product. Product registration facilitates technical support and ensures that you receive important information updates from NI.

A Declaration of Conformity (DoC) is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification. If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world. For telephone support in the United States, create your service request at ni.com/support or dial 1 866 ASK MYNI (275 6964). For telephone support outside the United States, visit the Worldwide Offices section of ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for more information on National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patents Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the readme file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS. U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14s, DFAR 252.227-7014, and DFAR 252.227-7015.

© 2008–2015 National Instruments. All rights reserved.