COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

Sell For Cash Get Credit Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

APEX WAVES

Bridging the gap *between the manufacturer and your legacy test system.*

1-800-915-6216

www.apexwaves.com

sales@apexwaves.com

Request a Quote CLICK HERE *PXI-5441*

CALIBRATION PROCEDURE

NI 5421/5441

100 MS/s Arbitrary Waveform Generator

This document contains the verification and adjustment procedures for the NI 5421/5441 arbitrary waveform generator. This calibration procedure is intended for metrology labs.

Refer to ni.com/calibration for more information about calibration solutions.

Contents

NATIONAL INSTRUMENTS

Software

Calibrating the NI 5421 requires you to install NI-FGEN version 2.0 or later on the calibration system. Calibrating the NI 5441 requires you to install NI-FGEN version 2.3 or later on the calibration system. You can download NI-FGEN from the National Instruments website at `ni.com/downloads`. NI-FGEN supports programming the *Self-Calibration Procedures* and the *Calibration Procedures* in the LabVIEW, LabWindows™/CVI™, and C application development environments (ADEs). When you install NI-FGEN, you only need to install support for the ADE that you intend to use.

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

Calibration functions are C function calls or LabVIEW VIs in NI-FGEN. The C function calls are valid for any compiler capable of calling a 32-bit DLL. Many of the functions use constants defined in the `niFgen.h` file. To use these constants in C, you must include `niFgen.h` in your code when you write the calibration procedure. Refer to the following table for file locations.

Table 1. Calibration File Locations

File Name and Location	Description
`IVI\Bin\niFgen_32.dll`	The NI-FGEN library, which provides the functionality for calibrating the NI 5421/5441.
`IVI\Lib\msc\niFgen.lib` `IVI\Lib\bc\niFgen.lib`	Allows you to create applications that call functions in the `niFgen_32.dll`: • For Microsoft Visual C/C++, link to `msc\niFgen.lib`. • For LabWindows/CVI, link to the library appropriate to your current compatibility mode (`msc` for Microsoft Visual C/C++).
`IVI\Include\niFgen.h`	A header file for the accessible functions in the `niFgen_32.dll`. You must include this file in any C code that you write to call these functions.
`<LabVIEW>\instr.lib\niFgen\niFgen.llb` (LabVIEW)	Contains VIs that correspond to the functions in the `niFgen_32.dll`.
`IVI\Drivers\niFgen\niFgen.fp` (CVI)	Contains the function panels for the functions in the `niFgen_32.dll`.

The calibration process is described in the *Self-Calibration Procedures* and the *Calibration Procedures* sections, including step-by-step instructions on calling the appropriate calibration functions.

Documentation

Consult the following documents for information about the NI 5421/5441, NI-FGEN, and your application software. All documents are available at ni.com/manuals and are installed with the software.

NI Signal Generators Getting Started Guide

Contains instructions for installing and configuring NI signal generators.

NI PXI/PCI-5421 Specifications

Provides the published specification values and calibration interval for the NI 5421. Refer to the most recent *NI PXI/PCI-5421 Specifications* online at ni.com/manuals.

NI PXI-5441 Specifications

Provides the published specification values and calibration interval for the NI 5441. Refer to the most recent *NI PXI-5441 Specifications* online at ni.com/manuals.

NI Signal Generators Help

Contains detailed information about the NI 5421/5441 and the NI-FGEN LabVIEW VI and C function programming references. Access this help file by selecting **Start»All Programs»National Instruments»NI-FGEN»Documentation»NI Signal Generators Help**.

Self-Calibration Procedures

The NI 5421/5441 can perform self-calibration, which adjusts the gain and offset of the main and direct analog paths. Self-calibration uses only an onboard analog-to-digital converter (ADC) to measure the output voltage. You can implement self-calibration on the NI 5421/5441 by following procedures similar to the *Verifying the DC Gain and Offset Accuracy* and the *Adjusting the Analog Output* procedures described in this document. However, output impedance, oscillator frequency, and the calibration ADC are not adjusted during self-calibration.

You can initiate self-calibration interactively from Measurement & Automation Explorer (MAX) or from the FGEN Soft Front Panel (SFP). Alternatively, you can initiate self-calibration programmatically using NI-FGEN.

MAX

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

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

FGEN Soft Front Panel

To initiate self-calibration from the FGEN Soft Front Panel (SFP), complete the following steps:

1. Select the device that you want to calibrate using the **Device Configuration** dialog box (**Edit»Device Configuration**).
2. Open the **Calibration** dialog box (**Utility»Calibration**).
3. Click **Perform self-calibration**.

NI-FGEN

To self-calibrate the NI 5421/5441 programmatically using NI-FGEN, complete the following steps:

1. Call `niFgen_init` (niFgen Initialize VI) to open an NI-FGEN session using the following parameters:
   - **resourceName**: The name of the device that you want to calibrate. 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.
2. Call `niFgen_SelfCal` (niFgen Self Cal VI) using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`
3. Call `niFgen_close` (niFgen Close VI) to close the NI-FGEN session using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`

Calibration Options

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. You can use verification to ensure that the adjustment process was successful or to determine if the adjustment process needs to be performed.

Adjustment is the process of measuring and compensating for device performance to improve the output accuracy. Performing an adjustment updates the calibration date, resetting the calibration interval. The device is guaranteed to meet or exceed its published specifications for the duration of the calibration interval.

This document provides two sets of test limits for most verification stages, the *calibration test limits* and the *published specifications*. The calibration test limits are more restrictive than the published specifications. 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.

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

Calibration Requirements

This section describes the test equipment and test conditions required for calibration.

Test Equipment

External calibration requires different equipment for each applicable specification. Refer to Table 2 for a list of equipment.

Table 2. Equipment Required for Calibrating the NI 5421/5441

Instrument	Recommended Instrument	Applicable Specification	Minimum Specifications
Digital multimeter (DMM)	NI PXI-4070 Agilent/HP 34401A Keithley 2000	AC accuracy, DC gain and offset, and frequency accuracy	DCV accuracy: ≤0.05% DC input impedance: ≥1 GΩ ACV accuracy: ≤0.16% AC input impedance: ≥1 MΩ Bandwidth: ≥100 kHz
Male banana-to-female BNC adapter	—		—
Male BNC-to-female SMB cable	—		50 Ω, RG-223
Frequency counter	Agilent/HP 53131A or HP 53132A with timebase option 001, 010, or 012	Frequency accuracy	Ability to measure 10 MHz or higher sine waves Frequency accuracy to ±500 ppb
Male BNC-to-female SMB cable	—		50 Ω, RG-223
Power meter/sensor	R&S NRP-Z91	Frequency response (flatness)	VSWR: (50 kHz to 120 MHz) ≤1.11 Relative power accuracy: ≤0.022 dB
Type N female-to-SMB plug adapter	Pasternak PE9316		VSWR: 1.3

Test Conditions

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

- Keep connections to the NI 5421/5441 short. Long cables and wires act as antennas, picking up extra noise that can affect measurements.
- Keep relative humidity below 80%.
- Maintain a temperature between 18 °C and 28 °C.
- Observe the 15 minute warm-up time.
- Allow the DMM to warm up for its recommended warm up interval.

System Setup

Before you begin the calibration procedure, connect the Reference Clock output from your counter to the CLK IN connector on your DUT (device under test).

Calibration Procedures

The complete calibration procedure consists of self-calibrating, verifying the performance of the NI 5421/5441, adjusting the calibration constants, and re-verifying performance after the adjustments. In some cases, the complete calibration procedure may not be required. Refer to the *Calibration Options* section for more information.

The external calibration procedure automatically stores the calibration date to allow traceability.

Verifying NI 5421/5441 Specifications

Note Always self-calibrate the NI 5421/5441 before beginning a verification procedure.

This section provides instructions for verifying the NI 5421/5441 specifications. This section also includes instructions for updating the calibration cycle.

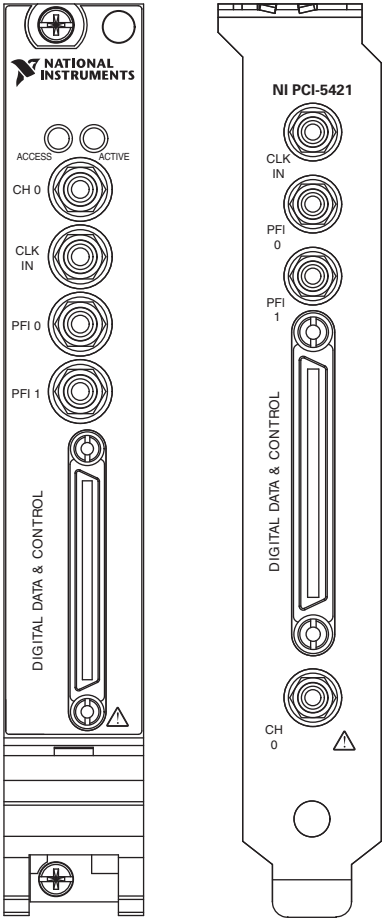
You can verify the following specifications for the NI 5421/5441:

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

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 5421/5441 to NI for repair.

Refer to Table 2 for information about which instrument to use for verifying each specification. Refer to Figure 1 for the names and locations of the NI PXI-5421/5441 and the NI PCI-5421 front panel connectors. The NI PXI-5421/5441 is pictured on the left. The NI PCI-5421 is pictured on the right.

Figure 1. NI PXI-5421/5441 and NI PCI-5421 Front Panel Connectors



Verifying the Oscillator Frequency Accuracy

This test verifies the frequency accuracy of the oscillator on the NI 5421/5441. Verification involves generating a 10 MHz sine wave with the NI 5421/5441 and measuring the sine wave frequency with one of the instruments from Table 2.

To verify the frequency accuracy of the oscillator on the NI 5421/5441, complete the following steps:

1. Connect the NI 5421/5441 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. 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_ConfigureSampleRate` (niFgen Set Sample Rate VI) using the following parameters:
   - **vi**: The session handle returned from `niFgen_init`
   - **Sample Rate**: 100 MS/s (`100000000`)
4. 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**: `1`

   Note You can adjust the gain value based on which measuring device you use.
5. 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`

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

6. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the digital filter state (NI-FGEN Digital Filter Enabled property) 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` to set the digital filter interpolation factor (NI-FGEN Digital Filter Interpolation Factor property) 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. Each waveform should have 10 samples per cycle with a total of 500 samples and 50 sine wave cycles. Because you set the sample rate to 100 MS/s and use 10 samples per cycle, the resulting waveform is a 10 MHz sine wave.

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

10. (LabVIEW Only) You must call the Sine Pattern VI to create the array of waveform samples for step 9. Specify 500 **samples**, an **amplitude** of 1, and 50 **cycles**. 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 11.
11. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_init`
    - **wfmSize**: The size in samples (`500`) of the waveform you created in step 9 or step 10.
    - **wfmData**: The array of waveform samples you created in step 9 or step 10.
    - **wfmHandle**: The variable passed by reference through this parameter receives the value (waveform handle) that identifies the waveform created by this function.
12. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`

13. Measure the frequency output of the NI 5421/5441.

    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 3 for frequency ranges.

Table 3. Calibration Limits for Frequency Accuracy

Frequency Limit	As-Found Limits	As-Left Limits
Low	9,999,750 Hz	9,999,955 Hz
High	10,000,250 Hz	10,000,045 Hz

14. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
15. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, to destroy the instrument driver session and all of its properties, and to 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 5421/5441 into a high-impedance load by generating a number of DC voltages and offsets, measuring the voltage with a DMM, and comparing the NI 5421/5441 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 analog path gain

Verifying the Main Analog Path Gain

Complete the following steps to verify the NI 5421/5441 main analog path gain.

1. Connect the NI 5421/5441 CH 0 front panel connector to the DMM.
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` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` 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**: `10000000000`
5. 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_MAIN_ANALOG_PATH`
6. Call `niFgen_SetAttributeViReal64` to set the output impedance (NI-FGEN Output Impedance property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_init`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
   - **value**: `50`
7. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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. (LabVIEW Only) You must call the Initialize Array function to create the array of samples for step 8. Wire 1.0 to the **element** input and specify a dimension size of 500. Wire the **initialized array** output of the Initialize Array function to the **Waveform Data Array** input of the niFgen Create Waveform (DBL) VI in step 10.

10. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_init`
    - **wfmSize**: The size in samples (`500`) of the waveform you created in step 8 or step 9
    - **wfmData**: The array of waveform samples that you created in step 8 or step 9
    - **wfmHandle**: The variable passed by reference through this parameter receives the value (waveform handle) that identifies the waveform created by this function (positive full-scale handle).
11. 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).
12. (LabVIEW Only) You must call the Initialize Array function to create the array of samples for step 11. Wire -1.0 to the **element** input and specify a dimension size of 500. Wire the **initialized array** output of the Initialize Array function to the **Waveform Data Array** input of the niFgen Create Waveform (DBL) VI in step 13.
13. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_init`
    - **wfmSize**: The size in samples (`500`) of the waveform that you created in step 11 or step 12
    - **wfmData**: The array of waveform samples that you created in step 11 or step 12
    - **wfmHandle**: The variable passed by reference through this parameter receives the value (waveform handle) that identifies the waveform created by this function (negative full-scale handle).
14. 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`

Repeat steps 15 through 25 for each of the 24 iterations listed in Table 4, changing the *Gain* value for each iteration

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

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	As-Found Limits (Volts)	As-Left Limits(Volts)
1	12.000000	12.000000	-12.000000	±0.048500	±0.019700
2	10.000000	10.000000	-10.000000	±0.040500	±0.016500
3	7.000000	7.000000	-7.000000	±0.028500	±0.011700

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

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	As-Found Limits (Volts)	As-Left Limits(Volts)
4	5.000000	5.000000	-5.000000	±0.020500	±0.008500
5	3.500000	3.500000	-3.500000	±0.014500	±0.006100
6	2.500000	2.500000	-2.500000	±0.010500	±0.004500
7	2.000000	2.000000	-2.000000	±0.008500	±0.003700
8	1.650000	1.650000	-1.650000	±0.007100	±0.003140
9	1.250000	1.250000	-1.250000	±0.005500	±0.002500
10	0.850000	0.850000	-0.850000	±0.003900	±0.001860
11	0.600000	0.600000	-0.600000	±0.002900	±0.001460
12	0.415000	0.415000	-0.415000	±0.002160	±0.001164
13	0.300000	0.300000	-0.300000	±0.001700	±0.000980
14	0.205000	0.205000	-0.205000	±0.001320	±0.000828
15	0.150000	0.150000	-0.150000	±0.001100	±0.000740
16	0.105000	0.105000	-0.105000	±0.000920	±0.000668
17	0.075000	0.075000	-0.075000	±0.000800	±0.000620
18	0.055000	0.055000	-0.055000	±0.000720	±0.000588
19	0.037500	0.037500	-0.037500	±0.000650	±0.000560
20	0.026000	0.026000	-0.026000	±0.000604	±0.000542
21	0.018500	0.018500	-0.018500	±0.000574	±0.000530
22	0.013000	0.013000	-0.013000	±0.000552	±0.000521
23	0.009000	0.009000	-0.009000	±0.000536	±0.000514
24	0.006500	0.006500	-0.006500	±0.000526	±0.000510

Note: *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*)

15. 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**: The *Gain* value listed in Table 4 for the current iteration
16. Call `niFgen_SetAttributeViInt32` to choose the positive full-scale DC waveform (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 10 (positive full-scale handle)
17. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
18. Measure the DC voltage from the NI 5421/5441. This value is the *Measured Positive Full-Scale Value*.
19. Determine the error for positive full scale using the following formula:

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

    Compare this error to the calibration limits listed in Table 4.
20. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
21. Call `niFgen_SetAttributeViInt32` to choose the negative full-scale DC waveform (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 13 (negative full-scale handle)
22. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
23. Measure the DC voltage from the NI 5421/5441. This value is the *Measured Negative Full-Scale Value*.

24. Determine the error for negative full scale using the following formula:

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

Compare this error to the calibration limits listed in Table 4.

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

Verifying the Main Analog Path Offset

To verify the offset of the NI 5421/5441 main analog path, complete 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. (LabVIEW Only) You must call the Initialize Array function to create the array of samples for step 1. Wire 0.0 to the **element** input and specify a dimension size of 500. Wire the **initialized array** output of the Initialize Array function to the **Waveform Data Array** input of the niFgen Create Waveform (DBL) VI in step 3.
3. Call `niFgen_CreateWaveformF64` (niFgen Create Waveform (DBL) VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_init`
    - **wfmSize**: The size in samples (`500`) of the waveform that you created in step 1 or step 2
    - **wfmData**: The array of waveform samples that you created in step 1 or step 2
    - **wfmHandle**: The variable passed by reference through this parameter receives the value (waveform handle) that identifies the waveform created by this function (mid-scale handle).
4. Call `niFgen_SetAttributeViInt32` to choose the mid-scale handle DC waveform (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 3 (mid-scale handle)

Repeat steps 5 through 16 for each of the 24 iterations listed in Table 5, changing the *Ideal Positive Offset*, *Ideal Negative Offset*, and *Gain* values for each iteration.

Table 5. Values for Verifying the Offset of the Main Analog Path

Iteration	Gain	Ideal Positive Offset (Volts)	Ideal Negative Offset (Volts)	As-Found Limits (Volts)	As-Left Limits (Volts)
1	12.000000	6.000000	-6.000000	±0.051500	±0.021500
2	10.000000	5.000000	-5.000000	±0.043000	±0.018000
3	7.000000	3.500000	-3.500000	±0.030250	±0.012750
4	5.000000	2.500000	-2.500000	±0.021750	±0.009250
5	3.500000	1.750000	-1.750000	±0.015375	±0.006625
6	2.500000	1.250000	-1.250000	±0.011125	±0.004875
7	2.000000	1.000000	-1.000000	±0.009000	±0.004000
8	1.650000	0.825000	-0.825000	±0.007513	±0.003388
9	1.250000	0.625000	-0.625000	±0.005813	±0.002688
10	0.850000	0.425000	-0.425000	±0.004113	±0.001988
11	0.600000	0.300000	-0.300000	±0.003050	±0.001550
12	0.415000	0.207500	-0.207500	±0.002264	±0.001226
13	0.300000	0.150000	-0.150000	±0.001775	±0.001025
14	0.205000	0.102500	-0.102500	±0.001371	±0.000859
15	0.150000	0.075000	-0.075000	±0.001138	±0.000763
16	0.105000	0.052500	-0.052500	±0.000946	±0.000684
17	0.075000	0.037500	-0.037500	±0.000819	±0.000631
18	0.055000	0.027500	-0.027500	±0.000734	±0.000596
19	0.037500	0.018750	-0.018750	±0.000659	±0.000566
20	0.026000	0.013000	-0.013000	±0.000611	±0.000546
21	0.018500	0.009250	- 0.009250	±0.000579	±0.000532
22	0.013000	0.006500	- 0.006500	±0.000555	±0.000523
23	0.009000	0.004500	- 0.004500	±0.000538	±0.000516
24	0.006500	0.003250	- 0.003250	±0.000528	±0.000511

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

5. 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**: The *Ideal Positive Offset* value listed in Table 5 for the current iteration
6. 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**: The *Gain* value listed in Table 5 for the current iteration
7. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`
8. Measure the positive DC voltage from the NI 5421/5441. This value is the *Measured Positive Offset Value*.
9. Determine the error for positive offset using the following formula:

   Error Positive Offset =
   (*Measured Positive Offset Value*) - (*Ideal Positive Offset Value*)

   Compare this error to the calibration limits listed in Table 5.
10. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
11. 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**: The *Ideal Negative Offset* value listed in Table 5 for the current iteration
12. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
13. Measure the negative DC voltage from the NI 5421/5441. This value is the *Measured Negative Offset Value*.
14. Determine the error for negative offset using the following formula:

    Error Negative Offset =
    (*Measured Negative Offset Value*) - (*Ideal Negative Offset Value*)

Compare this error to the calibration limits listed in Table 5.

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

Verifying the Direct Analog Path Gain

Complete the following steps to verify the NI 5421/5441 direct analog path gain.

Note The offset is not adjustable for the direct analog path.

1. 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`
2. 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`

Repeat steps 3 through 12 for each of the seven iterations listed in Table 6, changing the *Gain* value for each iteration.

Table 6. 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)	As-Found Limits (Volts)	As-Left Limits (Volts)
1	1.000000	1.000000	-1.000000	±0.025000	±0.004000	±0.001600
2	0.950000	0.950000	-0.950000	±0.025000	±0.003800	±0.001520
3	0.900000	0.900000	-0.900000	±0.025000	±0.003600	±0.001440
4	0.850000	0.850000	-0.850000	±0.025000	±0.003400	±0.001360
5	0.800000	0.800000	-0.800000	±0.025000	±0.003200	±0.001280

Table 6. Values for Verifying the Gain of the Direct Analog Path (Continued)

Iteration	Gain	Ideal Positive Full-Scale (Volts)	Ideal Negative Full-Scale (Volts)	Offset Limit (Volts)	As-Found Limits (Volts)	As-Left Limits (Volts)
6	0.750000	0.750000	-0.750000	±0.025000	±0.003000	±0.001200
7	0.710000	0.710000	-0.710000	±0.025000	±0.002840	±0.001136

Note: *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` 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**: The *Gain* value listed in Table 6 for the current iteration
4. Call `niFgen_SetAttributeViInt32` to choose the positive full-scale DC waveform (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 10 of the *Verifying the Main Analog Path Gain* section (positive full-scale handle)
5. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`
6. Measure the positive DC voltage from the NI 5421/5441. This value is the *Measured Positive Full-Scale Value*.
7. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the waveform generation using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`
8. Call `niFgen_SetAttributeViInt32` to choose the negative full-scale DC waveform (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 13 of the *Verifying the Main Analog Path Gain* section (negative full-scale handle)

9. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the waveform generation using the following parameter:
   - **vi**: The session handle returned from `niFgen_init`
10. Measure the negative DC voltage from the NI 5421/5441. This value is the *Measured Negative Full-Scale Value*.
11. Call `niFgen_AbortGeneration` (niFgen Abort Generation VI) to abort the 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 6 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 limits listed in Table 6, perform an external adjustment.
16. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, to destroy the instrument driver session and all of its properties, and to release any memory resources that NI-FGEN uses. Use the following parameter:
    - **vi**: The session handle returned from `niFgen_init`

Verifying the AC Voltage Amplitude Absolute Accuracy

This test verifies the AC voltage amplitude absolute accuracy of the NI 5421/5441 using a DMM. Complete the following steps to verify the AC accuracy of the NI 5421/5441.

1. Connect the NI 5421/5441 CH 0 front panel connector to the DMM. Connect the positive terminal to the center pin of the NI 5421/5441 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_ConfigureSampleRate` (niFgen Set Sample Rate VI) using the following parameters:
   - **vi**: The session handle returned from `niFgen_init`
   - **Sample Rate**: 100 MS/s (`100000000`)
4. 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`
5. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_init`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_ANALOG_FILTER_ENABLED`
   - **value**: `VI_FALSE`
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 array should contain a single cycle sine wave of 2,000 samples and an amplitude of 1.
8. (LabVIEW Only) You must call the niFgen Util Create Waveform Data VI to generate the 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 9.
9. 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 that you created in step 7 or step 8
   - **wfmData**: The array of waveform samples (double representation) that you created in step 7 or step 8
   - **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.

Table 7. Values for Verifying the AC Voltage Amplitude Absolute Accuracy

Iteration	Gain	DMM Range (V_{RMS})	Expected Amplitude (V_{RMS})	Test Limit ($-V_{RMS}$)	Test Limit ($+V_{RMS}$)
1	12.000000	50	8.485281	-0.085560	0.170413
2	10.000000	50	7.071068	-0.0714178	0.142128
3	7.000000	5	4.949747	-0.050205	0.099702
4	5.000000	5	3.535534	-0.036062	0.071418
5	3.500000	5	2.474874	-0.025456	0.050205
6	2.500000	5	1.767767	-0.018385	0.036062
7	2.000000	5	1.414214	-0.014849	0.028991
8	1.650000	5	1.166726	-0.012374	0.024042
9	1.250000	5	0.883883	-0.009546	0.018385
10	0.850000	5	0.601041	-0.006718	0.012728
11	0.600000	0.5	0.424264	-0.004950	0.009192
12	0.415000	0.5	0.293449	-0.003642	0.006576
13	0.300000	0.5	0.212132	-0.002828	0.004950
14	0.205000	0.5	0.144957	-0.0021587	0.003606
15	0.150000	0.5	0.106066	-0.001768	0.002828
16	0.105000	0.5	0.074246	-0.001450	0.002192
17	0.075000	0.5	0.053033	-0.001237	0.001768
18	0.055000	0.5	0.038809	-0.001096	0.001485
19	0.037500	0.5	0.026517	-0.003359	0.006010
20	0.026000	0.5	0.018385	-0.0008910	0.001075
21	0.018500	0.5	0.013081	-0.000838	0.000969
22	0.013000	0.5	0.009192	-0.000799	0.000891
23	0.009000	0.5	0.006364	-0.000771	0.000834
24	0.006500	0.5	0.004596	-0.000753	0.000799

10. Configure the DMM using the following settings:
    - Function: AC voltage
    - Range: Refer to Table 7
    - Input impedance: 1 MΩ
    - 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 9. The input impedance should be equal to or greater than the values indicated in Table 2, *Equipment Required for Calibrating the NI 5421/5441*.

11. Repeat steps 9 through 18 for each of the 24 iterations listed in Table 7, changing the *Gain* and *DMM Range (V_{RMS})* values for each iteration.
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 7 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 5421/5441 to settle.
15. Measure and record the output voltage amplitude with the DMM. This value is the measured amplitude, $measured V_{RMS}$.
16. Calculate the peak amplitude error using the following equation:

$$expected V_{RMS} - measured V_{RMS} = error$$

17. Compare the output error to the test limits in Table 7 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. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, to destroy the instrument driver session and all of its properties, and to release any memory resources that NI-FGEN uses. Use the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
20. If any of the errors are greater than the *As-Found Limits*, perform an external adjustment.

Verifying Frequency Response (Flatness)

This test verifies the frequency response (flatness) of the NI 5421/5441 using a power meter. The flatness verification has two subprocedures that verify the following:

- Main analog path flatness: low-gain amplifier and high-gain amplifier
- Direct analog path flatness

Verifying the Main Analog Path Flatness

Complete the following steps to verify the main analog path flatness of the NI 5421/5441.

1. Connect the NI 5421/5441 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 5421/5441 output. Use the following parameters:
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED`
   - **value**: `VI_FALSE`
   - **vi**: The session handle returned from `niFgen_init`
4. Null the power meter according to the power meter documentation.
5. Configure the power meter using the following settings:
   - Average: 16
   - Measure: Watts
6. 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`
7. 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`

8. Call `niFgen_SetAttributeViReal64` to set the gain (NI-FGEN Gain property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_ARB_GAIN`
   - **value**: `1`
9. 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`
10. 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`)

Repeat steps 11 through 20 for each iteration in Table 8, changing the *Number of Samples* and *Number of Cycles* for each iteration.

Table 8. NI 5421/5441 Setup for Main Analog Path Flatness Verification

Iteration	Frequency	Number of Samples	Number of Cycles	Published Specification	
				Low-Gain Amplifier	High-Gain Amplifier
1	50 kHz	2,000	1	REF	REF
2	100 kHz	1,000	1	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB
3	1 MHz	1,000	10	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB
4	5 MHz	1,000	50	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB
5	10 MHz	1,000	100	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB
6	15 MHz	1,000	150	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB

Table 8. NI 5421/5441 Setup for Main Analog Path Flatness Verification (Continued)

Iteration	Frequency	Number of Samples	Number of Cycles	Published Specification	
				Low-Gain Amplifier	High-Gain Amplifier
7	20 MHz	1,000	200	-1.0 dB to +0.5 dB	-1.2 dB to +0.5 dB
8	43 MHz	1,000	430	-3.0 dB to +0.5 dB	-3.0 dB to +0.5 dB

11. Create an array of waveform samples. Each waveform should have samples and cycles that correspond to the current iteration in Table 8.
12. (LabVIEW Only) You must call the Sine Pattern VI to create the array of waveform samples for step 11. Specify an **amplitude** of `0.5`, and **samples** and **cycles** that correspond to the current iteration in Table 8. 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 13.
13. 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 11 or step 12
    - **wfmArray**: The array of waveform samples that you created in step 11 or step 12 (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.
14. 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 13

    a. Call `niFgen_SetAttributeViBoolean` to set the digital filter state (NI-FGEN Digital Filter Enabled property) using the following parameters:
       - **vi**: The session handle returned from `niFgen_InitExtCal`
       - **channelName**: `"0"`
       - **attributeID**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
       - **value**: `VI_TRUE`

b. Call `niFgen_SetAttributeViReal64` to set the digital filter interpolation factor (NI-FGEN Digital Filter Interpolation Factor property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_DIGITAL_FILTER_INTERPOLATION_FACTOR`
   - **value**: `4`

15. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
16. Allow the power meter to stabilize for 10 seconds.
17. 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}).
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. For iterations 2-7 in Table 8, using the recorded power values, calculate the deviation from the reference (50 kHz) power using the following equation:

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

20. Compare the *Flatness (dB)* calculated in step 19 to the *Published Specification* value for the current amplifier path listed in Table 8.
21. To verify the flatness for the High-Gain Amplifier Path, repeat the process from step 1, but in step 8, set the **Gain** to `3`.
22. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, to destroy the instrument driver session and all of its properties, and to 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

Complete the following steps to verify the direct analog path flatness of the NI 5421/5441, complete the following steps:

1. Connect the NI 5421/5441 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 5421/5441 output. Use the following parameters:
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_ENABLED`
   - **value**: `VI_FALSE`
   - **vi**: The session handle returned from `niFgen_init`
4. Null the power meter according to the power meter documentation.
5. Configure the power meter using the following settings:
   - Average: 16
   - Measure: Watts
6. 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`
7. 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`
8. Call `niFgen_SetAttributeViReal64` to set the gain (NI-FGEN Gain property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_ARB_GAIN`
   - **value**: `0.5`
9. 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`

10. 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`)

Repeat steps 11 through 20 for each iteration in Table 9, changing the *Number of Samples* and *Number of Cycles* for each iteration.

Table 9. NI 5421/5441 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	100 kHz	1,000	1	-0.4 dB to +0.3 dB
3	1 MHz	1,000	10	-0.4 dB to +0.3 dB
4	5 MHz	1,000	50	-0.4 dB to +0.3 dB
5	10 MHz	1,000	100	-0.4 dB to +0.3 dB
6	15 MHz	1,000	150	-0.4 dB to +0.4 dB
7	20 MHz	1,000	200	-0.4 dB to +0.4 dB
8	25 MHz	1,000	250	-0.4 dB to +0.6 dB
9	30 MHz	1,000	300	-0.4 dB to +0.6 dB
10	35 MHz	1,000	350	-0.4 dB to +0.6 dB
11	40 MHz	1,000	400	-0.4 dB to +0.6 dB
12	43 MHz	1,000	430	-3.0 dB to +0.6 dB

11. Create an array of waveform samples. Each waveform should have samples and cycles that correspond to the current iteration in Table 9.
12. (LabVIEW Only) You must call the Sine Pattern VI to create the array of waveform samples for step 11. Specify an **amplitude** of 1, and **samples** and **cycles** that correspond to the current iteration in Table 9. 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 13.
13. 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 11 or step 12.
    - **wfmArray**: The array of waveform samples from step 11 or step 12 (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.

14. 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 13

    a. Call `niFgen_SetAttributeViBoolean` to set the digital filter state (NI-FGEN Digital Filter Enabled property) using the following parameters:
        - **vi**: The session handle returned from `niFgen_InitExtCal`
        - **channelName**: `"0"`
        - **attributeID**: `NIFGEN_ATTR_DIGITAL_FILTER_ENABLED`
        - **value**: `VI_TRUE`

    b. Call `niFgen_SetAttributeViReal64` to set the digital filter interpolation factor (NI-FGEN Digital Filter Interpolation Factor property) using the following parameters:
        - **vi**: The session handle returned from `niFgen_InitExtCal`
        - **channelName**: `"0"`
        - **attributeID**: `NIFGEN_ATTR_DIGITAL_FILTER_INTERPOLATION_FACTOR`
        - **value**: `4`

15. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) using the following parameter:
    - **vi**: The session handle returned from `niFgen_init`
16. Allow the power meter to stabilize for 10 seconds.
17. 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}).
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. For iterations 2-11 in Table 9, using the recorded power values, calculate the deviation from the reference (50 kHz) power using the following equation:

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

20. Compare the *Flatness (dB)* values calculated in step 19 to the *Published Specification* listed in Table 9.

21. Call `niFgen_close` (niFgen Close VI) to close the instrument driver session, to destroy the instrument driver session and all of its properties, and to release any memory resources that NI-FGEN uses. Use the following parameter:
    - **vi**: The session handle returned from `niFgen_init`

Adjusting the NI 5421/5441

If the NI 5421/5441 successfully passes all verification within the calibration test limits, adjustment is recommended, but not required, to guarantee its published specifications for the next two years. If the NI 5421/5441 is not within the calibration test limits for each verification procedure, perform the adjustment procedure to improve the accuracy of the NI 5421/5441. Refer to the *Calibration Options* section to determine which procedures to perform.

An adjustment is required only once every two years. The adjustment procedure automatically updates the calibration date and temperature in the EEPROM of the signal generator.

If the NI 5421/5441 passed verification within the calibration test limits and you do not want to do 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 you want to calibrate. This name can be found under Devices and Interfaces in MAX.
    - **password**: The password required to open an external calibration session. If this password has not been changed since manufacturing, the password is `"NI"`.
    - **vi**: A pointer to a ViSession. The variable passed by reference through this parameter receives the value that identifies the external calibration 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.
2. Call `niFgen_CloseExtCal` (niFgen Close Ext Cal VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_InitExtCal`
    - **action**: `NIFGEN_VAL_EXT_CAL_COMMIT`

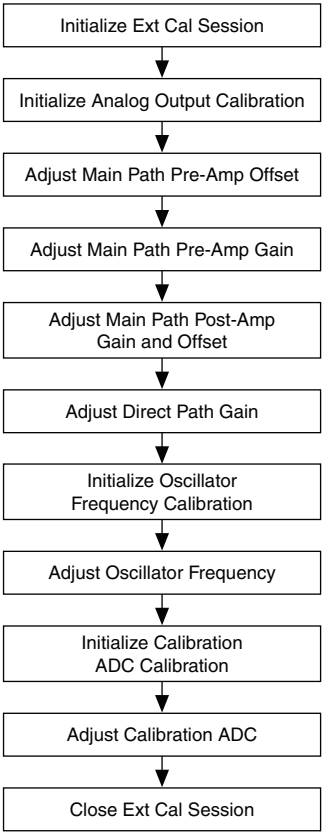
The external calibration procedure adjusts the analog output, the oscillator frequency, and the calibration ADC. Analog output adjustment characterizes the DC gains and the offsets of the analog path to ensure the analog output voltage accuracy. Adjusting the oscillator frequency adjusts the onboard 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.

You cannot perform an external calibration using a standard NI-FGEN session. You must create an external calibration session using `niFgen_InitExtCal` (niFgen Init Ext Cal VI). An external calibration session allows you to use NI-FGEN functions and attributes that are specifically for external calibration, while still allowing you to use all the standard NI-FGEN functions and attributes with the external calibration session.

Along with the standard NI-FGEN attributes, the external calibration session uses a set of calibration constants that are determined during the calibration procedure and stored in the device onboard memory when the session is closed. NI-FGEN uses these calibration constants during a standard NI-FGEN session to ensure that the device operates within its specifications.

You must close an external calibration session by using `niFgen_CloseExtCal` (niFgen Close Ext Cal VI), as shown in the following figure.

Figure 2. NI 5421/5441 External Calibration Procedure



Adjusting the Analog Output

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

- Main analog path preamplifier offset
- Main analog path preamplifier gain
- Main analog path postamplifier gain and offset
- Direct analog path gain

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

Initializing Analog Output Calibration

1. Call `niFgen_InitializeAnalogOutputCalibration` (niFgen Initialize Analog Output Calibration VI) 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` to set the analog path value (NI-FGEN Analog Path property) 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
   - **value**: `2000`
5. Call `niFgen_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
   - **value**: `32767`
6. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) 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` to set the postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) 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` to set the output impedance (NI-FGEN Output Impedance property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
   - **value**: `50`
10. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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 Analog Path Preamplifier Offset

1. Call `niFgen_SetAttributeViInt32` to set the analog path value (NI-FGEN Analog Path property) 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` to set the postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) 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`
4. Repeat steps 5 through 8 for each of the 10 iterations listed in the following table, changing the *Analog Filter Enable*, *Preamplifier Attenuation*, and *Current Configuration* values for each iteration.

Table 10. Attributes and Values for Main Analog Path Preamplifier Offset

Iteration	Analog Filter Enable	Preamplifier 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

5. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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.

6. Call `niFgen_SetAttributeViReal64` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION`
   - **value**: The *Preamplifier Attenuation* value for the current iteration from Table 10.
7. Complete the following steps to take the voltage measurements at the NI 5421/5441 CH 0 front panel connector into a high-impedance load:
   a. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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` to set the offset DAC value (NI-FGEN Offset DAC Value property) 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 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 voltage generated by the device. This measurement is `measurement 0`, which is used in step 8.
   f. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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 voltage generated by the device. This measurement is `measurement 1`, which is used in step 8.

j. Call `niFgen_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) 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 voltage generated by the device. This measurement is `measurement 2`, which is used in step 8.

8. Call `niFgen_CalAdjustMainPathPreAmpOffset` (niFgen Cal Adjust Main Path Pre Amp Offset VI) 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
- **gainDACValues**: An array containing two elements—the two values (`2000`, `1000`) that you set as the gain DAC in the order that you measured them
- **offsetDACValues**: An array containing two elements—the two values (`50000`, `15000`) that you set as the offset DAC in the order that you measured them
- **measuredOutputs**: An array containing three elements—the three output voltages (`measurement 0`, `measurement 1`, `measurement 2`) that you measured in the order that you measured them.

Adjusting the Main Analog Path Preamplifier Gain

1. Call `niFgen_SetAttributeViInt32` to set the analog path value (NI-FGEN Analog Path property) 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` to set the postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) 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_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
   - **value**: `32000`

Repeat steps 4 through 7 for each of the 10 iterations listed in Table 11, changing the *Analog Filter Enable*, *Preamplifier Attenuation*, and *Current Configuration* values for each iteration.

Table 11. Attributes and Values for Main Analog Path Preamplifier Gain

Iteration	Analog Filter Enable	Preamplifier 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` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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 11

5. Call `niFgen_SetAttributeViReal64` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_PRE_AMPLIFIER_ATTENUATION`
   - **value**: The *Preamplifier Attenuation* value for the current iteration from Table 11
6. Complete the following steps to take the voltage measurements at the NI 5421/5441 CH 0 front panel connector into a high-impedance load:
   a. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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 Value 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 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 voltage generated by the device. This measurement is `measurement 0`, which is used in step 7.
   f. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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 generated 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) 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**: An array containing two elements—the two values (`25233`, `-29232`) that you set for the main DAC—in the order that you measured them
   - **gainDACValues**: An array containing two elements—the two values (`1500`, `2000`) that you set for the gain DAC—in the order that you measured them
   - **offsetDACValues**: An array containing one element—the value (`32000`) that you set for the offset DAC
   - **measuredOutputs**: An array containing two elements—the two output voltages (`measurement 0`, `measurement 1`) that you measured—in the order that you measured them

Adjusting the Main Analog Path Postamplifier 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` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
   - **value**: `2000`

Repeat steps 5 through 8 for each of the eight iterations listed in Table 12, changing the *Analog Path*, *Postamplifier Attenuation*, and *Current Configuration* values for each iteration.

Table 12. Attributes and Values for the Main Analog Path Postamplifier Gain and Offset

Iteration	Analog Path	Post-amplifier Attenuation	Current Configuration
1	NIFGEN_VAL_FIXED_ LOW_GAIN_ANALOG_PATH	0	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ LOW_GAIN_0DB
2	NIFGEN_VAL_FIXED_ LOW_GAIN_ANALOG_PATH	12	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ LOW_GAIN_12DB
3	NIFGEN_VAL_FIXED_ LOW_GAIN_ANALOG_PATH	24	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ LOW_GAIN_24DB
4	NIFGEN_VAL_FIXED_ LOW_GAIN_ANALOG_PATH	36	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ LOW_GAIN_36DB
5	NIFGEN_VAL_FIXED_ HIGH_GAIN_ANALOG_PATH	0	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ HIGH_GAIN_0DB
6	NIFGEN_VAL_FIXED_ HIGH_GAIN_ANALOG_PATH	12	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ HIGH_GAIN_12DB
7	NIFGEN_VAL_FIXED_ HIGH_GAIN_ANALOG_PATH	24	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ HIGH_GAIN_24DB
8	NIFGEN_VAL_FIXED_ HIGH_GAIN_ANALOG_PATH	36	NIFGEN_VAL_CAL_ CONFIG_MAIN_PATH_ HIGH_GAIN_36DB

5. Call `niFgen_SetAttributeViInt32` to set the analog path value (NI-FGEN Analog Path property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`

- **attributeID**: `NIFGEN_ATTR_ANALOG_PATH`
- **value**: The *Analog Path* value for the current iteration from Table 12

6. Call `niFgen_SetAttributeViReal64` to set the Postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
   - **value**: The *Postamplifier Attenuation* value for the current iteration from Table 12
7. Complete the following steps to take the voltage measurements at the NI 5421/5441 CH 0 front panel connector into a high-impedance load:
   a. Call `niFgen_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
      - **vi**: The session handle returned from `niFgen_InitExtCal`
      - **channelName**: `"0"`
      - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
      - **value**: `50000`
   b. 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`
   c. Wait 500 ms for the output to settle.
   d. Use the DMM to measure the voltage generated by the device. This measurement is `measurement 0`, which is used in step 8.
   e. Call `niFgen_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
      - **vi**: The session handle returned from `niFgen_InitExtCal`
      - **channelName**: `"0"`
      - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
      - **value**: `15000`
   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 voltage generated by the device. 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) 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 12
- **mainDACValues**: An array containing two elements—the values (`0`, `0`)—that you set on the main DAC**gainDACValues**: An array containing one element—the value (`2000`)—that you set on the gain DAC
- **offsetDACValues**: An array containing two elements—the two values (`50000`, `15000`)—that you set on the offset DAC in order
- **measuredOutputs**: An array containing two elements—the two output voltages (`measurement 0`, `measurement 1`)—that you measured in order

Adjusting the Direct Analog 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_GAIN_DAC_VALUE`
   - **value**: `2000`
3. Call `niFgen_SetAttributeViInt32` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
   - **value**: `32767`
4. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) 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` to set the postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_POST_AMPLIFIER_ATTENUATION`
   - **value**: `0`
7. Call `niFgen_SetAttributeViReal64` to set the output impedance (NI-FGEN Output Impedance property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
   - **value**: `50`
8. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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` to set the analog path value (NI-FGEN Analog Path property) 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 the voltage measurements at the NI 5421/5441 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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 voltage generated by the device. This measurement is `measurement 0`, which is used in step 12.

f. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `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 voltage generated by the device. 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `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 voltage generated by the device. This measurement is `measurement 2`, which is used in step 12.

o. Call `niFgen_SetAttributeViInt32` to set the gain DAC value (NI-FGEN Gain DAC Value property) using the following parameters:

- **vi**: The session handle returned from `niFgen_InitExtCal`
- **channelName**: `"0"`
- **attributeID**: `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 voltage generated by the device. This measurement is `measurement 3`, which is used in step 12.

12. Call `niFgen_CalAdjustDirectPathGain` (niFgen Cal Adjust Direct Path Gain VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_InitExtCal`
    - **channelName**: `"0"`
    - **mainDACValues**: An array containing two elements—the two values (`32767`, `-32767`) that you set on the main DAC in order
    - **gainDACValues**: An array containing four elements—the four values (`1800`, `2600`, `1500`, `2300`) that you set on the gain DAC in order
    - **measuredOutputs**: An array containing four elements—the four output voltages (`measurement 0`, `measurement 1`, `measurement 2`, `measurement 3`) that you measured in order

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. Repeat this process until the difference between the desired and measured frequency falls within the desired tolerance, which is 4.5 ppm. This adjustment ensures the frequency accuracy of the onboard oscillator.

1. Call `niFgen_InitializeOscillatorFrequencyCalibration` (niFgen Initialize Oscillator Frequency Calibration VI) using the following parameter:
    - **vi**: The session handle returned from `niFgen_InitExtCal`
2. Call `niFgen_SetAttributeViReal64` to set the sample rate (niFgen Set Sample Rate VI) 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` to set the gain (NI-FGEN Gain property) 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 measuring device you use.

4. Call `niFgen_SetAttributeViReal64` to set the offset (NI-FGEN Offset property) 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 measuring device you use.

5. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the digital filter state (NI-FGEN Digital Filter Enabled property) 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` to set the digital filter interpolation factor (NI-FGEN Digital Filter Interpolation Factor property) 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` to set the output impedance (NI-FGEN Output Impedance property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`

- **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
- **value**: `50`

9. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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. Because you set the sample rate to 100 MS/s and because there are 10 samples per cycle, the resulting waveform is a 10 MHz sine wave.

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

11. Call `niFgen_CreateArbWaveform` (niFgen Create Arbitrary Waveform VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_InitExtCal`
    - **wfmSize**: 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
    - **wfmHandle**: The variable passed by reference through this parameter receives the value (waveform handle) that identifies the waveform created by this function.
12. Call `niFgen_SetAttributeViInt32` to choose the sine waveform (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 11 (sine waveform handle)
13. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the 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 5421/5441.

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

b. Call `niFgen_CalAdjustOscillatorFrequency` (niFgen Cal Adjust Oscillator Frequency VI) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **desiredFrequencyInHz**: The desired frequency (`10000000`) of the generated sinusoid in Hz
   - **measuredFrequencyInHz**: The `measured frequency` of the generated sinusoid in Hz

c. Call `niFgen_InitiateGeneration` (niFgen Initiate Generation VI) to initiate the 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 the waveform generation using the following parameter:
    - **vi**: The session handle returned from `niFgen_InitExtCal`

Adjusting the Calibration ADC

The NI 5421/5441 has an onboard calibration ADC used during self-calibration. To adjust the calibration ADC, characterize the gain and offset associated with this ADC so that a self-calibration results in an accurately calibrated device.

1. Call `niFgen_InitializeCalADCCalibration` (niFgen Initialize Cal ADC Calibration VI) 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` to set the analog path value (NI-FGEN Analog Path property) 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` to set the gain DAC value (NI-FGEN Gain DAC Value property) 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` to set the offset DAC value (NI-FGEN Offset DAC Value property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OFFSET_DAC_VALUE`
   - **value**: `32767`
6. Call `niFgen_SetAttributeViBoolean` to set the analog filter state (NI-FGEN Analog Filter Enabled property) 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` to set the preamplifier attenuation (NI-FGEN Pre-Amplifier Attenuation property) 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` to set the postamplifier attenuation (NI-FGEN Post-Amplifier Attenuation property) 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` to set the output impedance (NI-FGEN Output Impedance property) using the following parameters:
   - **vi**: The session handle returned from `niFgen_InitExtCal`
   - **channelName**: `"0"`
   - **attributeID**: `NIFGEN_ATTR_OUTPUT_IMPEDANCE`
   - **value**: `50`
10. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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`

12. Wait 500 ms for the output to settle.
13. Call `niFgen_SetAttributeViInt32` to set the calibration ADC input (NI-FGEN Cal ADC Input property) 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` to disable the analog output (NI-FGEN Output Enabled property) 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**: 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` to enable the analog output (NI-FGEN Output Enabled property) 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 5421/5441 voltage output 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` to disable the analog output (NI-FGEN Output Enabled property) 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**: A ViReal64 variable. The variable passed by reference through this parameter receives the voltage measured by the onboard calibration ADC. This value is `cal ADC measurement 1`, which is used in step 32.
28. Call `niFgen_SetAttributeViBoolean` to enable the analog output (NI-FGEN Output Enabled property) 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 5421/5441 voltage output directly into the DMM (into a high-impedance load). This measurement is `external measurement 1`, which is used in step 32.

32. Call `niFgen_CalAdjustCalADC` (niFgen Cal Adjust Cal ADC VI) using the following parameters:
    - **vi**: The session handle returned from `niFgen_InitExtCal`
    - **voltagesMeasuredExternally**: An array containing two elements—the two voltages (`external measurement 0`, `external measurement 1`) that you measured with the DMM—in the order that you measured them
    - **voltagesMeasuredWithCalADC**: An array containing two elements—the two voltages (`cal ADC measurement 0`, `cal ADC measurement 1`) that you measured with the onboard calibration ADC—in the order that you measured them.
33. Call `niFgen_SetAttributeViBoolean` to disable the analog output (NI-FGEN Output Enabled property) 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` to set the calibration ADC input (NI-FGEN Cal ADC Input property) 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**: One of the following values:
  - If the external adjustment procedure completed without any errors, use `NIFGEN_VAL_EXT_CAL_COMMIT`. This function stores the new calibration constants, updated calibration dates, updated calibration temperatures in the onboard EEPROM.
  - If any errors occurred during the external adjustment procedure or if you want to abort the operation, use `NIFGEN_VAL_EXT_CAL_ABORT`. This function discards the new calibration constants and does not change any of the calibration data stored in the onboard EEPROM.

Calibration Utilities

NI-FGEN supports several calibration utilities that allow you to retrieve information about adjustments performed on the NI 5421/5441, restore an external calibration, change the external calibration password, and store small amounts of information in the onboard EEPROM. You can retrieve some data using MAX or the FGEN SFP; however, you can retrieve all the data using NI-FGEN.

MAX

To retrieve data using MAX, complete the following steps:

1. Launch MAX.
2. Select the device from which you want to retrieve information from **My System»Devices and Interfaces»PXI System**.
3. Select the **Calibration Tab** on the lower right corner.

   You should see information about the last 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. Select the device from which you want to retrieve information using the Device Configuration dialog box (**Edit»Device Configuration**).
3. Open the Calibration dialog box (**Utility»Calibration**).

   You should see information about the last dates for both external and self-calibration.

NI-FGEN

NI-FGEN provides a full complement of calibration utility functions and VIs. Refer to the *NI Signal Generators Help* for the complete function reference, including the following utility functions:

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

Worldwide Support and Services

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 event

370559G-01 Apr19