

# NI 449x Calibration Procedure

Français    Deutsch    日本語    한국어    简体中文

[ni.com/manuals](http://ni.com/manuals)

This document contains information about calibrating National Instruments NI 449x devices. It does *not* discuss programming techniques or compiler configuration. The National Instruments DAQmx driver includes help files that have compiler-specific instructions and detailed function explanations. You can install these help files when you install NI-DAQmx on the calibration computer.

## Contents


---

Conventions .....	2
Software .....	3
Documentation .....	3
Calibration Interval .....	3
Password .....	3
Getting Calibration Information.....	4
Test Equipment .....	6
Test Conditions .....	6
Calibration Procedure .....	7
Initial Setup.....	7
Self-Calibration.....	8
Verification Procedure .....	8
Analog Input Performance Verification .....	8
Timebase Frequency Accuracy Verification .....	18
Adjustment Procedure.....	22
Analog Input Adjustment .....	22
Where to Go for Support.....	25

# Conventions

---

The following conventions appear in this document:

- » The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
-  This icon denotes a note, which alerts you to important information.
- bold** Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names and hardware labels.
- italic* Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.
- monospace Monospace text denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.
- NI 449x NI 449x refers to the NI 4495, NI 4496, and NI 4498 unless otherwise noted.
- Number Variable Types The integer variable type used in this document is represented as a whole number. Any floating point variable type is represented as a number with a decimal point. For example, the number 1,024 is written as 1024 if it is an integer, and as 1024.0 if it is a floating point value.
- Return Value A parameter which points to data that is modified by a function call has a variable name in the form <variableTypeReturnValue>, for example, `uInt32PointerReturnValue`. All function calls return a status integer that you must verify. Since all the functions return this status integer, the function descriptions in this document omit the status integer.

# Software

---

Calibration requires the latest NI-DAQmx driver. NI-DAQmx includes high-level function calls to simplify the task of writing software to calibrate devices. The driver supports many programming languages, including LabVIEW, LabWindows™/CVI™, C/C++, C#, and Visual Basic .NET.

If you want to write the calibration procedure using LabVIEW, NI recommends using the Full or Professional Development System or the Developer Suite Professional Edition. These editions include frequency-domain analysis tools not included in the Base Development System.

# Documentation

---

If you are using the NI-DAQmx driver, the following documents are your primary references for writing your calibration utility:

- The *NI-DAQmx C Reference Help* includes information about the functions in the driver.
- The *DAQ Getting Started Guide* for NI-DAQ 8.0 or later provides instructions for installing and configuring NI-DAQ devices.
- The *NI-DAQmx Help* includes information about creating applications that use the NI-DAQmx driver.
- *NI 449x Specifications* contains information about specific device features. The limits used to verify the accuracy of the devices are based on the specifications found in this document.

These documents are installed with NI-DAQmx. You can also download the latest versions from the NI Web site at [ni.com/manuals](http://ni.com/manuals).

# Calibration Interval

---

NI 449x devices should be externally calibrated at a regular interval as defined by the measurement accuracy requirements of your application. NI recommends that you perform a complete calibration at least once every year. Based on your measurement accuracy needs, you can shorten this interval to 90 days or six months. Self-calibration can be performed as needed or when the temperature varies by 5 °C or more from the last external calibration.

# Password

---

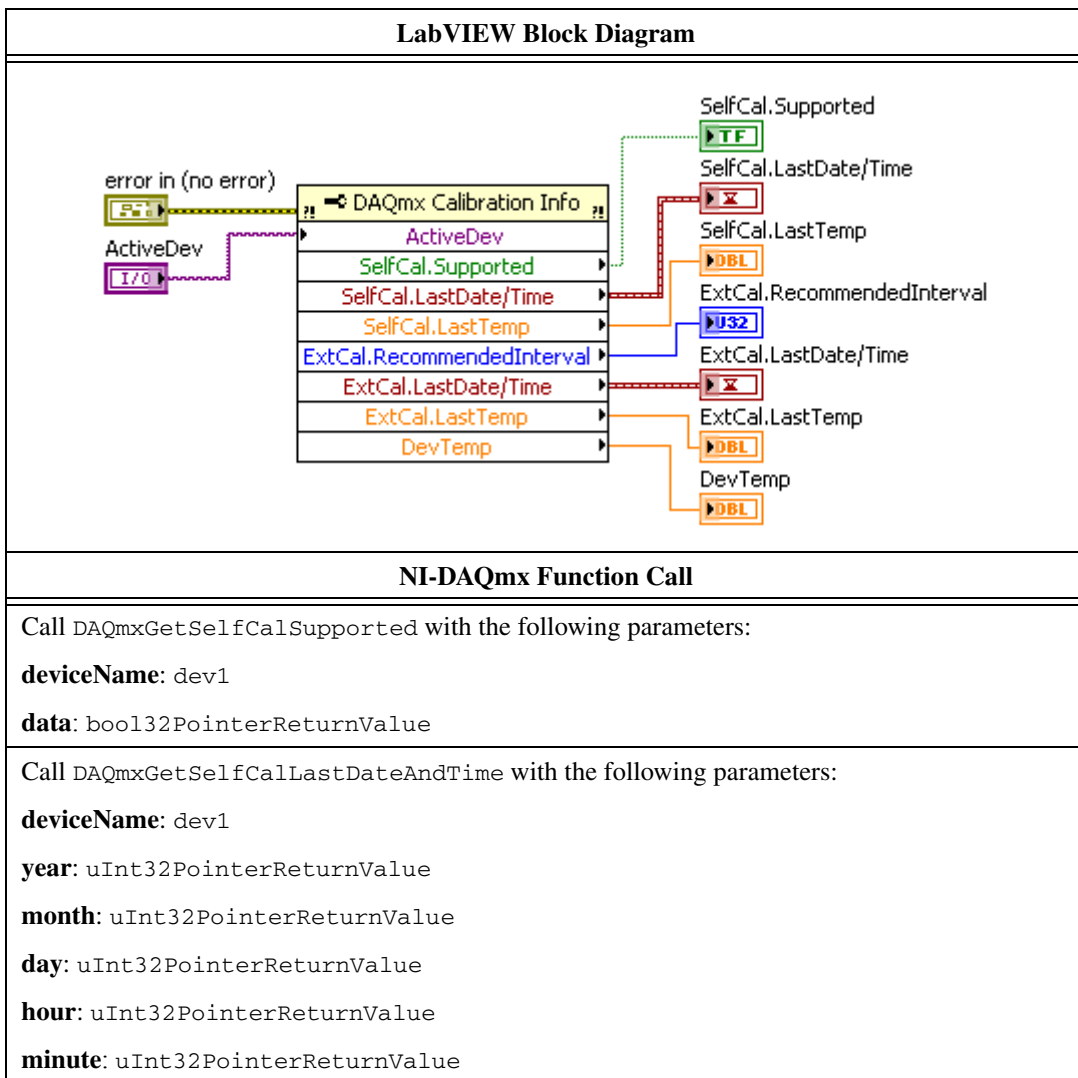
The default password for password-protected operations is NI.

# Getting Calibration Information

The calibration information for the NI 449x devices contain the following:

- Whether the device supports self-calibration
- Date and time of the last self-calibration
- Recommended interval for external calibration
- Date and time of the last external calibration

You can access this information in the Measurement & Automation Explorer (MAX) or by using the following LabVIEW VIs.



Call DAQmxGetSelfCalLastTemp with the following parameters:

**deviceName:** dev1

**data:** float64PointerReturnValue

Call DAQmxGetExtCalRecommendedInterval with the following parameters:

**deviceName:** dev1

**data:** uInt32PointerReturnValue

Call DAQmxGetExtCalLastDateAndTime with the following parameters:

**deviceName:** dev1

**year:** uInt32PointerReturnValue

**month:** uInt32PointerReturnValue

**day:** uInt32PointerReturnValue

**hour:** uInt32PointerReturnValue

**minute:** uInt32PointerReturnValue

Call DAQmxGetExtCalLastTemp with the following parameters:

**deviceName:** dev1

**data:** float64PointerReturnValue0.

Call DAQmxGetCalDevTemp with the following parameters:

**deviceName:** dev1

**data:** float64PointerReturnValue

# Test Equipment

---

Table 1 lists the instruments NI recommends for calibrating the NI 449x.

**Table 1.** NI 449x Calibration Equipment Specifications

Instrument	Critical Specifications	Suggested Model
Calibrator	Frequency Range: 20 Hz–92.2 kHz Voltage Range: up to 9 V <sub>pk</sub> AC Accuracy: $\pm 0.05\%$ <sup>1</sup> , 20 Hz–95 kHz DC Accuracy: $\pm 115$ ppm <sup>1</sup> at 5 V	Fluke 5700A
Function Generator	Frequency Range: up to 10 kHz Frequency Accuracy: $\pm 2$ ppm <sup>2</sup> Voltage Range: up to 9 V <sub>pk</sub>	HP/Agilent 33250A
<sup>1</sup> AC accuracy is the sum of all errors, including percent of reading error and temperature error, but excluding percent of range error. <sup>2</sup> Frequency accuracy is the sum of all errors, including initial accuracy and stability errors.		

If the recommended instrument is not available, select a substitute calibration standard that meets the given specifications.

## Test Conditions

---

Follow these guidelines to optimize the connections and the environment during verification and adjustment:

- Keep connections to the NI 449x short. Long cables and wires act as antennae, picking up extra noise that can affect measurements.
- Use 50  $\Omega$  BNC coaxial cable for all connections to the device.
- Keep relative humidity between 10 and 80%, noncondensing, or consult the device documentation for the optimum relative humidity.
- Maintain the temperature between 18 and 28 °C, or refer to the device specifications for the optimum temperature range.
- Allow a warm-up time of at least 15 minutes to ensure that the measurement circuitry of the NI 449x is at a stable operating temperature.
- Allow a warm-up time for all the instruments and equipment, according to the manufacturers' instructions.

# Calibration Procedure

---

This section provides instructions for verifying and adjusting the NI 449x device.

The calibration procedure consists of the following steps:

1. Initial Setup—Configure the device in NI-DAQmx.
2. Self-Calibration—Measure the onboard reference voltage of the device and adjust the self-calibration constants.
3. NI 449x Verification Procedure—Verify the accuracy of the device prior to calibration.
4. NI 449x Adjustment Procedure—Adjust the device calibration constants with respect to known voltage sources.
5. Perform another verification to ensure that the device operates with the specifications after adjustment.

The initial setup, self-calibration, verification, and adjustment procedures are explained in the following sections.



**Note** You must compare the verification limits provided in this procedure with the most recent specifications. Refer to the latest *NI 449x Specifications* document at [ni.com/manuals/](http://ni.com/manuals/).

## Initial Setup

NI-DAQmx automatically detects all NI 449x devices. However, for the driver to communicate with the device, you must configure the device in NI-DAQmx.

Complete the following steps to configure a device in NI-DAQmx:


1. Install the NI-DAQmx driver software.
2. Power off the computer that will hold the device and install the device in an available slot.
3. Power on the computer and launch MAX.
4. Right-click the device name and select **Self-Test** to ensure that the device is working properly.



**Note** When a device is configured with MAX, it is assigned a device name. Each function call uses this device name to determine which DAQ device to calibrate. This document uses `dev1` to refer to the device name. In the following procedures, use the device name as it appears in MAX.

## Self-Calibration

After warm up, call self-calibration before completing the first verification. The DAQmx Self Calibrate VI, as shown below, measures the onboard reference voltage of the device and adjusts the self-calibration constants to account for any errors caused by short-term fluctuations in the operating environment. When you self-calibrate a device, no external signal connections are necessary.

LabVIEW Block Diagram	NI-DAQmx Function Call
	Call DAQmxSelfCal with the following parameters: <b>deviceName:</b> dev1

## Verification Procedure

Verification determines how well the NI 449x device is meeting specifications. By completing this procedure, you can see how the device has drifted over time, which helps you determine the appropriate calibration interval for your application.



**Note** Use the before adjustment values found in the tables to verify NI 449x devices before adjustments are made. Use the after adjustment values found in the tables to verify NI 449x devices after adjustments are made.

## Analog Input Performance Verification

This section verifies the analog input (AI) performance of all NI 449x devices. Refer to the *NI 449x Specifications* for the number of channels and performance specifications.

## AI DC Offset Verification

Complete the following steps to verify the AI DC offset:

1. Connect a BNC shorting cap to the analog input channel 0 of the device. The terminator grounds the input channel because it is in pseudodifferential configuration. Wait 5 seconds for inputs to settle due to AC coupling on the NI 4496/4498.
2. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 3 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call <code>DAQmxCreateTask</code> with the following parameter: <b>taskHandle:</b> mytaskHandleReturnValue

Use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

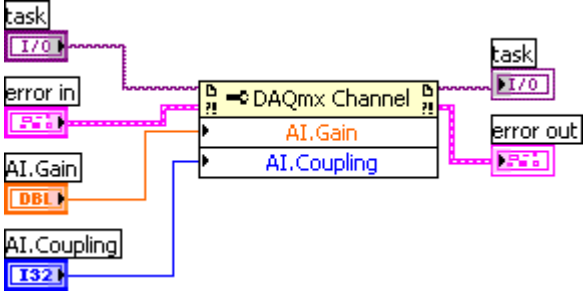
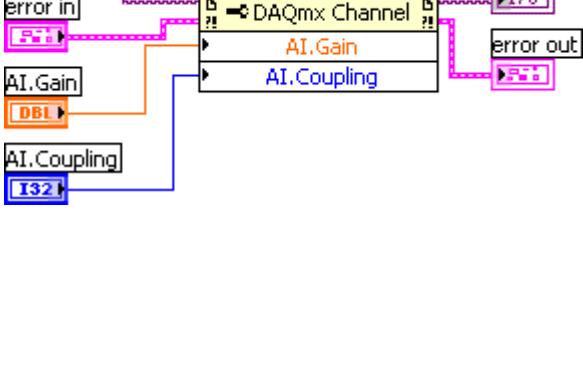
3. Create an AI voltage channel through NI-DAQmx using the `DAQmx Create Virtual Channel VI`, as shown below.



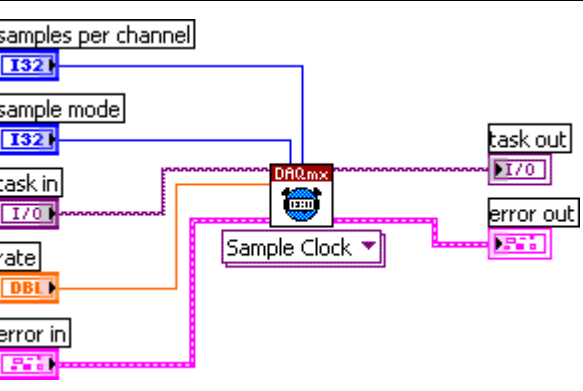
**Note** Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call <code>DAQmxCreateAIVoltageChan</code> with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>physicalChannel:</b> dev1/ai0</p> <p><b>nameToAssignToChannel:</b> ""</p> <p><b>terminalConfig:</b> DAQmx_Val_PseudoDiff</p> <p><b>minVal:</b> -10.0</p> <p><b>maxVal:</b> 10.0</p> <p><b>units:</b> DAQmx_Val_Volts</p> <p><b>CustomScaleName:</b> NULL</p>

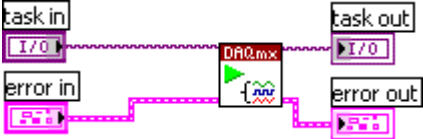
4. Modify the AI voltage channel property using the following functions.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a central 'DAQmx Channel' block with two input properties: 'AI.Gain' and 'AI.Coupling'. The 'AI.Gain' property is connected to a numeric control set to 0.0. The 'AI.Coupling' property is connected to a dropdown menu set to 'DC'. The block is connected to a 'task' block, which has 'I/O' and 'error in' inputs and 'I/O' and 'error out' outputs.</p>	<p>Call DAQmxSetAIGain with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> 0.0 (gain value)</p>
 <p>The diagram shows a central 'DAQmx Channel' block with two input properties: 'AI.Gain' and 'AI.Coupling'. The 'AI.Gain' property is connected to a numeric control set to 0.0. The 'AI.Coupling' property is connected to a dropdown menu set to 'DC'. The block is connected to a 'task' block, which has 'I/O' and 'error in' inputs and 'I/O' and 'error out' outputs.</p>	<p>Call DAQmxSetAICoupling with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> DAQmx_Val_DC (NI 4495 only)</p> <p>or</p> <p><b>data:</b> DAQmx_Val_AC (NI 4496 and NI 4498 only)</p>

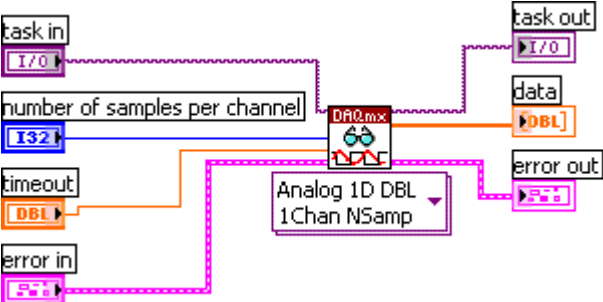
5. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a central 'DAQmx' block with a 'Sample Clock' dropdown menu. The 'Sample Clock' is set to 'Sample Clock'. The block is connected to a 'task in' block, which has 'I/O' and 'error in' inputs and 'I/O' and 'error out' outputs. The 'task in' block is connected to a 'task out' block, which has 'I/O' and 'error out' outputs. The 'task in' block is also connected to a 'task out' block, which has 'I/O' and 'error out' outputs.</p>	<p>Call DAQmxCfgSampClkTiming with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>source:</b> NULL</p> <p><b>rate:</b> 204800.0</p> <p><b>activeEdge:</b> DAQmx_Val_Rising</p> <p><b>sampleMode:</b> DAQmx_Val_FiniteSamps</p> <p><b>sampsPerChanToAcquire:</b> 102400</p>

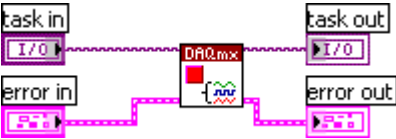
6. Start the task using the DAQmx Start Task, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a central DAQmx Start Task VI. It has four input terminals: 'task in' (I/O), 'error in' (Error), 'task out' (I/O), and 'error out' (Error). The 'task in' and 'error in' terminals are connected to corresponding 'task out' and 'error out' terminals. The DAQmx Start Task VI is represented by a green square with a white triangle and a red 'X'.</p>	<p>Call DAQmxStartTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

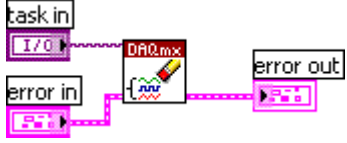
7. Acquire data using the DAQmx Read VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a central DAQmx Read VI. It has six input terminals: 'task in' (I/O), 'number of samples per channel' (I32), 'timeout' (DBL), 'error in' (Error), 'task out' (I/O), and 'data' (DBL). It has three output terminals: 'task out' (I/O), 'data' (DBL), and 'error out' (Error). The 'task in' and 'error in' terminals are connected to corresponding 'task out' and 'error out' terminals. The 'number of samples per channel' terminal is connected to a constant value of 132. The 'timeout' terminal is connected to a constant value of 10.0. The DAQmx Read VI is represented by a red square with a white triangle and a red 'X'. Below it is a dropdown menu set to 'Analog 1D DBL 1Chan NSamp'.</p>	<p>Call DAQmxReadAnalogF64 with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>numSampsPerChan:</b> -1</p> <p><b>timeout:</b> 10.0</p> <p><b>fillMode:</b> DAQmx_Val_GroupByChannel</p> <p><b>arraySizeInSamps:</b> 102400</p> <p><b>reserved:</b> NULL</p> <p><b>readArray:</b> float64ArrayReturnValue</p> <p><b>sampsPerChanRead:</b> int32PointerReturnValue</p>

8. Stop the task using the DAQmx Stop Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a central DAQmx Stop Task VI. It has four input terminals: 'task in' (I/O), 'error in' (Error), 'task out' (I/O), and 'error out' (Error). The 'task in' and 'error in' terminals are connected to corresponding 'task out' and 'error out' terminals. The DAQmx Stop Task VI is represented by a red square with a white triangle and a red 'X'.</p>	<p>Call DAQmxStopTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

9. Clear the task using the DAQmx Clear Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxClearTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

- Average all the values in the array returned from the acquisition. The average is the resulting offset for AI 0 at 0 dB device gain setting. Compare this value to the limits in Table 2.

**Table 2.** AI Offset Limits

Device	Device Gain	Before/After Adjustment	Device Input Offset	
			Min (mV)	Max (mV)
NI 4495 NI 4496	0 dB, 20 dB	Before Adjustment	-2	2
NI 4498	0 dB, 10 dB, 20 dB, 30 dB			
NI 4495 NI 4496	0 dB, 20 dB	After Adjustment	-0.5	0.5
NI 4498	0 dB, 10 dB, 20 dB, 30 dB			

- Repeat steps 2 through 10 for each gain value that is set in step 4 using the device gain values in Table 2.
- Repeat steps 1 through 11 for all remaining analog input channels of the device. Replace ai0 with the appropriate channel name in function call parameters physicalChannel and channel.

## AI Gain Accuracy Verification

Complete the following steps to verify AI gain accuracy:

1. Connect the output of the calibrator to the analog input channel 0 of the device.
2. Program the calibrator amplitude and frequency as listed in Table 3. Wait 5 seconds for the inputs to settle.
3. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 4 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call <code>DAQmxCreateTask</code> with the following parameter: <b>taskHandle:</b> <code>mytaskHandleReturnValue</code>

Use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

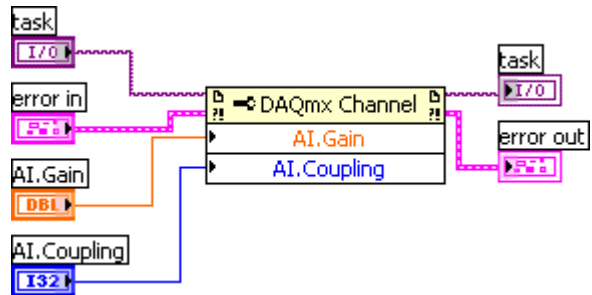
4. Create an AI voltage channel through NI-DAQmx using the `DAQmx Create Virtual Channel VI`, as shown below.



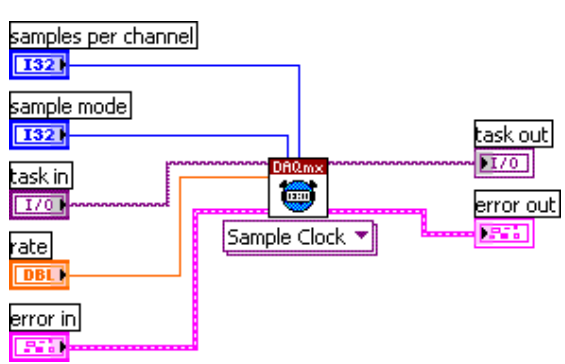
**Note** Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call <code>DAQmxCreateAIVoltageChan</code> with the following parameters:</p> <p><b>taskHandle:</b> <code>myTaskHandle</code></p> <p><b>physicalChannel:</b> <code>dev1/ai0</code></p> <p><b>nameToAssignToChannel:</b> <code>""</code></p> <p><b>terminalConfig:</b> <code>DAQmx_Val_PseudoDiff</code></p> <p><b>minVal:</b> <code>-10.0</code></p> <p><b>maxVal:</b> <code>10.0</code></p> <p><b>units:</b> <code>DAQmx_Val_Volts</code></p> <p><b>CustomScaleName:</b> <code>NULL</code></p>

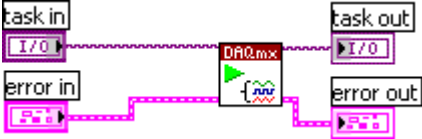
5. Modify the AI voltage channel property using the following functions.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Channel block with two sub-properties: AI.Gain and AI.Coupling. The AI.Gain property is connected to a DBL control labeled 'AI.Gain'. The AI.Coupling property is connected to an I32 control labeled 'AI.Coupling'. The DAQmx Channel block is connected to a task in, which is connected to an error in control. The DAQmx Channel block is also connected to a task out, which is connected to an error out control.</p>	<p>Call DAQmxSetAIGain with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> 0.0 (gain value)</p> <hr/> <p>Call DAQmxSetAICoupling with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> DAQmx_Val_DC (NI 4495 only)</p> <p>or</p> <p><b>data:</b> DAQmx_Val_AC (NI 4496 and NI 4498 only)</p>

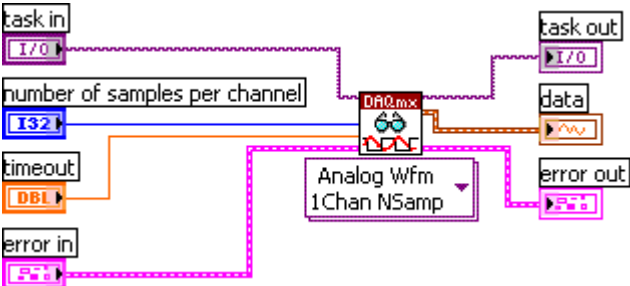
6. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Timing VI block. The 'samples per channel' property is connected to an I32 control labeled 'samples per channel'. The 'sample mode' property is connected to an I32 control labeled 'sample mode'. The 'rate' property is connected to a DBL control labeled 'rate'. The DAQmx Timing VI block is connected to a task in, which is connected to an error in control. The DAQmx Timing VI block is also connected to a task out, which is connected to an error out control.</p>	<p>Call DAQmxCfgSampClkTiming with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>source:</b> NULL</p> <p><b>rate:</b> 204800.0</p> <p><b>activeEdge:</b> DAQmx_Val_Rising</p> <p><b>sampleMode:</b> DAQmx_Val_FiniteSamps</p> <p><b>sampsPerChanToAcquire:</b> 65536</p> <p><b>Note:</b> For 20 Hz only, acquire 131072 samples per channel.</p>

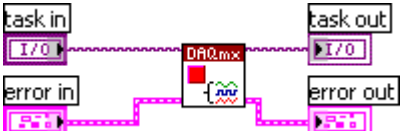
7. Start the task using the DAQmx Start Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxStartTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

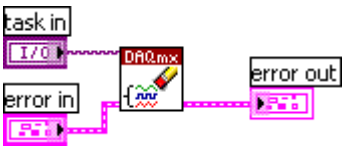
8. Acquire data using the DAQmx Read VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxReadAnalogF64 with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>numSampsPerChan:</b> -1</p> <p><b>timeout:</b> 10.0</p> <p><b>fillMode:</b> DAQmx_Val_GroupByChannel</p> <p><b>arraySizeInSamps:</b> 65536</p> <p><b>reserved:</b> NULL</p> <p><b>readArray:</b> float64ArrayReturnValue</p> <p><b>sampsPerChanRead:</b> int32PointerReturnValue</p>

9. Stop the task using the DAQmx Stop Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxStopTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

10. Clear the task using the DAQmx Clear Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxClearTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

11. If you use C function calls, perform an FFT on the array of data and find the amplitude of the fundamental harmonic. If you use LabVIEW, measure the amplitude of the input using the Amplitude and Level Measurements VI with the Apply Window option on. This is used to calculate the gain accuracy for the AI O at the 0 dB device gain setting.
12. Compare this amplitude to the limits (V) in Table 3. You will use this value as  $x$  in Table 5.

**Table 3.** AI Gain Accuracy Limits

Device Gain	Calibrator Output		Device Input Amplitude (Before Adjustment)		Device Input Amplitude (After Adjustment)	
	Amplitude ( $V_{pk}$ )	Frequency (Hz)	Min ( $V_{pk}$ )	Max ( $V_{pk}$ )	Min ( $V_{pk}$ )	Max ( $V_{pk}$ )
0 dB	9.0	1000	8.897	9.104	8.948	9.052
10 dB	3.0	1000	2.966	3.035	2.983	3.017
20 dB	0.9	1000	0.8897	0.9104	0.8948	0.9052
30 dB	0.3	1000	0.2966	0.3035	0.2983	0.3017

13. Repeat steps 2 through 12 for each gain value that is set in step 5. Modify the calibrator output amplitude in step 2. Refer to Table 3 for calibrator output amplitude and the device gain value.
14. Repeat steps 1 through 13 for all remaining analog input channels of the device. Replace ai0 with the appropriate channel name in function call parameters physicalChannel and channel.

## AI Flatness Verification

Complete the following steps to verify AI flatness:

1. Connect the output of the calibrator to the analog input channel 0 of the device.
2. Repeat steps 2 through 11 of the *AI Gain Accuracy Verification* section using the frequency from Table 5 and compare the amplitude value to the limits (V) in Table 5. For 20 Hz only, acquire 131072 samples per channel in step 6. The minimum and maximum values are the fractions multiplied by the corresponding measured value,  $x$ , you determined in the *AI Gain Accuracy Verification* section.
3. Repeat steps 1 through 2 of this section for each row in Table 5.
4. Repeat steps 2 and 3 for each gain value. Modify the calibrator output amplitude in step 2. Refer to Table 3 for the calibrator output amplitude and the device gain value.

**Table 4.** Maximum AI Flatness Specifications (Relative to 1 kHz)

Frequency Band	Flatness (dB)
20 Hz to 20 kHz	$\pm 0.005$
20 Hz to 45 kHz	$\pm 0.016$
20 Hz to 92.2 kHz	$\pm 0.08$

**Table 5.** AI Flatness Limits

Calibrator Output Frequency (Hz)	Device Input Amplitude	
	Min ( $V_{pk}$ )	Max ( $V_{pk}$ )
20	$0.99942 * x$	$1.00058 * x$
20,000	$0.99942 * x$	$1.00058 * x$
45,000	$0.99816 * x$	$1.00184 * x$
92,200	$0.99083 * x$	$1.00925 * x$

5. Repeat steps 1 through 4 for all remaining analog input channels of the device. Replace `ai0` with the appropriate channel name in function call parameters `physicalChannel` and `channel`.

## Timebase Frequency Accuracy Verification

This section describes the verification process for the timebase frequency accuracy on all the NI 449x devices. All analog inputs use a single circuit on each device. Therefore, you need to verify only the analog input frequency accuracy on a single channel to verify this circuit accuracy.

Complete the following steps to verify the timebase:

1. Connect the function generator to the analog input channel 0 of the device.
2. Output a sine wave using the function generator. The sine wave should have an amplitude of  $9 V_{pk}$  with no DC offset and a frequency of 10 kHz.
3. If you use LabVIEW, skip this step. If you use C function calls, create a task using `DAQmxCreateTask`, as shown below. The task is created in step 4 in LabVIEW.

LabVIEW Block Diagram	NI-DAQmx Function Call
LabVIEW does not require this step.	Call <code>DAQmxCreateTask</code> with the following parameter: <b>taskHandle:</b> <code>mytaskHandleReturnValue</code>

Use the object `myTaskHandle` in all subsequent NI-DAQmx function calls.

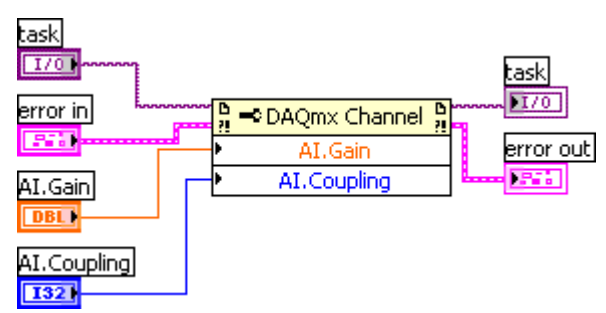
4. Create an AI voltage channel through NI-DAQmx using the DAQmx Create Virtual Channel VI, as shown below.



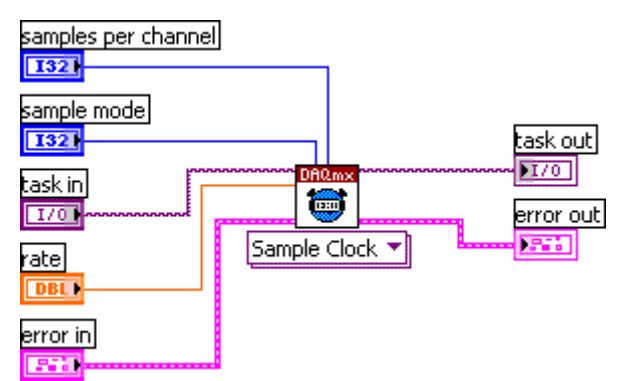
**Note** Refer to the NI-DAQmx function parameters for the LabVIEW input values.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxCreateAIVoltageChan with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>physicalChannel:</b> dev1/ai0</p> <p><b>nameToAssignToChannel:</b> ""</p> <p><b>terminalConfig:</b> DAQmx_Val_PseudoDiff</p> <p><b>minVal:</b> -10.0</p> <p><b>maxVal:</b> 10.0</p> <p><b>units:</b> DAQmx_Val_Volts</p> <p><b>CustomScaleName:</b> NULL</p>

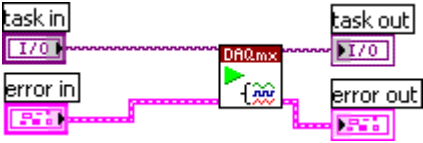
5. Modify the AI voltage channel property using the following functions.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Channel block with two input properties: AI.Gain and AI.Coupling. The AI.Gain property is connected to a DBL control labeled 'AI.Gain'. The AI.Coupling property is connected to an I32 control labeled 'AI.Coupling'. The DAQmx Channel block is connected to a task block, which has an I/O connector block and an error out connector block.</p>	<p>Call DAQmxSetAIGain with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> 0.0</p> <hr/> <p>Call DAQmxSetAICoupling with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>channel:</b> dev1/ai0</p> <p><b>data:</b> DAQmx_Val_DC (NI 4495 only)</p> <p>or</p> <p><b>data:</b> DAQmx_Val_AC (NI 4496 and NI 4498 only)</p>

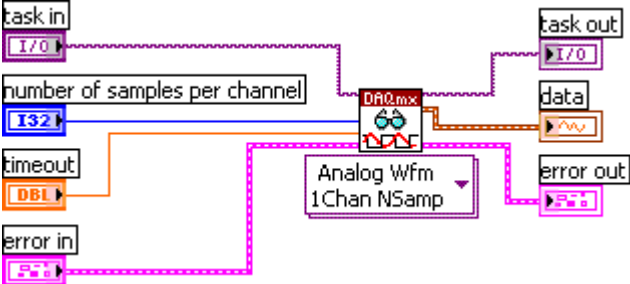
6. Configure the timing properties of the acquisition using the DAQmx Timing VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Timing VI block. It has several input properties: samples per channel (I32 control), sample mode (I32 control), rate (DBL control), and error in (FTI control). The samples per channel and sample mode properties are connected to I32 controls labeled 'samples per channel' and 'sample mode' respectively. The rate property is connected to a DBL control labeled 'rate'. The error in property is connected to an FTI control labeled 'error in'. The DAQmx Timing VI block is connected to a task block, which has a task out connector block, an I/O connector block, and an error out connector block. A 'Sample Clock' dropdown menu is also present in the diagram.</p>	<p>Call DAQmxCfgSampClkTiming with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>source:</b> NULL</p> <p><b>rate:</b> 40000.0</p> <p><b>activeEdge:</b> DAQmx_Val_Rising</p> <p><b>sampleMode:</b> DAQmx_Val_FiniteSamps</p> <p><b>sampsPerChanToAcquire:</b> 2560000</p>

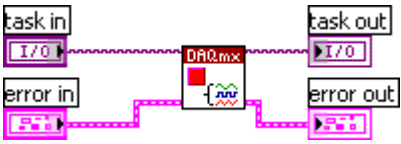
7. Start the task using the DAQmx Start Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Start Task VI block. It has two input terminals on the left: 'task in' (I/O) and 'error in' (E). It has two output terminals on the right: 'task out' (I/O) and 'error out' (E). The block is connected to a task handle variable 'myTaskHandle'.</p>	<p>Call DAQmxStartTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

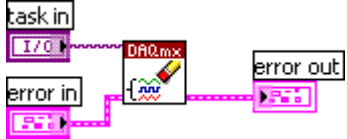
8. Create an array of double float of size 2560000 that will be filled with data. Acquire data using the DAQmx Read VI, as shown below. The acquisition takes over 1 minute.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Read Analog F64 VI block. It has four input terminals on the left: 'task in' (I/O), 'number of samples per channel' (I32), 'timeout' (DB1), and 'error in' (E). It has three output terminals on the right: 'task out' (I/O), 'data' (waveform), and 'error out' (E). The block is connected to a task handle variable 'myTaskHandle' and a waveform indicator 'Analog Wfm 1Chan NSamp'.</p>	<p>Call DAQmxReadAnalogF64 with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p> <p><b>numSampsPerChan:</b> -1</p> <p><b>timeout:</b> 100.0</p> <p><b>fillMode:</b> DAQmx_Val_GroupByChannel</p> <p><b>arraySizeInSamps:</b> 2560000</p> <p><b>reserved:</b> NULL</p> <p><b>readArray:</b> float64ArrayReturnValue</p> <p><b>sampsPerChanRead:</b> int32PointerReturnValue</p>

9. Stop the task using the DAQmx Stop Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
 <p>The diagram shows a DAQmx Stop Task VI block. It has two input terminals on the left: 'task in' (I/O) and 'error in' (E). It has two output terminals on the right: 'task out' (I/O) and 'error out' (E). The block is connected to a task handle variable 'myTaskHandle'.</p>	<p>Call DAQmxStopTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

10. Clear the task using the DAQmx Clear Task VI, as shown below.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxClearTask with the following parameters:</p> <p><b>taskHandle:</b> myTaskHandle</p>

11. If you use C function calls, perform an FFT on the array of data. If you use LabVIEW, use the Analog 1D Wfm 1chan Nsamp polymorphic VI to acquire data in step 8. Measure the exact frequency of the peak around 10 kHz using the Extract Single Tone Information VI with the following inputs:

- **advanced search»approx freq** : 10000
- **advanced search»search** : 1
- **export signals** : 0 (none)

12. Compare the detected frequency to the limits in Table 6.

**Table 6.** Timebase Frequency Accuracy Limits

Function Generator Output		Device Input Frequency	
Amplitude ( $V_{pk}$ )	Frequency (Hz)	Min (Hz)	Max (Hz)
9.0	10,000.00	9,999.70	10,000.30

## Adjustment Procedure

The NI 449x adjustment procedure adjusts the analog input calibration constants. At the end of each adjustment procedure, the new constants are stored in the external calibration area of the device EEPROM.

### Analog Input Adjustment

Complete the following steps to adjust the analog input:

1. Connect the output of the calibrator to all the analog input channels of the device, using the BNC T-connectors to split the signal in a tree formation to all the inputs. Wait 5 seconds for the input to settle in an AC coupled device.
2. Use the calibrator to output a  $200\text{ mV}_{rms}$  signal at 1 kHz.

- Initialize the AI calibration using the DAQmx Initialize External Calibration VI, as shown below.



**Note** Refer to the NI-DAQmx function parameters for the LabVIEW input values.

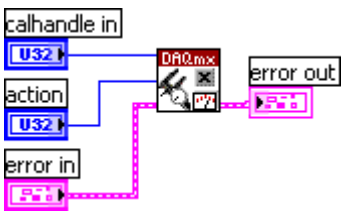
LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxInitExtCal with the following parameters:</p> <p><b>deviceName:</b> dev1</p> <p><b>password:</b> NI</p> <p><b>calHandle:</b> myCalHandle</p>

Use the object myCalHandle in all subsequent NI-DAQmx function calls.

- Perform the AI calibration using the DAQmx Adjust DSA AI Calibration VI, as shown below. The reference voltage parameter is the actual rms voltage value, in volts, that is output by the calibrator.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call DAQmxAdjustDSAAICal with the following parameters:</p> <p><b>calHandle:</b> myCalHandle</p> <p><b>referenceVoltage:</b> 0.2</p>

5. Finish the AI calibration with the DAQmx Close External Calibration VI, as shown below. Use the action `cancel` if there has been any error during the AI calibration or if you do not want to save the new AI calibration constants in the device EEPROM. Use the action `commit` if you want to save the new AI calibration constants in the device EEPROM.

LabVIEW Block Diagram	NI-DAQmx Function Call
	<p>Call <code>DAQmxCloseExtCal</code> with the following parameters:</p> <p><b>calHandle:</b> <code>myCalHandle</code></p> <p><b>action:</b> <code>DAQmx_Val_Action_Commit</code> or <code>DAQmx_Val_Action_Cancel</code></p>

The device is now calibrated with respect to your external source. After calibrating the device, verify the analog input operation by repeating the [Verification Procedure](#) section using the after adjustment values found in the tables to verify NI 449x devices after adjustments are made.

# Where to Go for Support

---

The National Instruments Web site is your complete resource for technical support. At [ni.com/support](http://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.

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 electronic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting [ni.com/certification](http://ni.com/certification). If your product supports calibration, you can obtain the calibration certificate for your product at [ni.com/calibration](http://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 to help address your support needs. For telephone support in the United States, create your service request at [ni.com/support](http://ni.com/support) and follow the calling instructions or dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 1800 300 800, Austria 43 662 457990-0,  
Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599,  
Canada 800 433 3488, China 86 21 5050 9800,  
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00,  
Finland 358 (0) 9 725 72511, France 01 57 66 24 24,  
Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737,  
Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400,  
Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710,  
Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466,  
New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60,  
Poland 48 22 328 90 10, Portugal 351 210 311 210,  
Russia 7 495 783 6851, Singapore 1800 226 5886,  
Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197,  
Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00,  
Switzerland 41 56 2005151, Taiwan 886 02 2377 2222,  
Thailand 662 278 6777, Turkey 90 212 279 3031,  
United Kingdom 44 (0) 1635 523545

National Instruments, NI, [ni.com](http://ni.com), and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on [ni.com/legal](http://ni.com/legal) for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at [ni.com/patents](http://ni.com/patents).