

# CALIBRATION PROCEDURE

# NI PXI-4130

This document contains instructions for writing a manual calibration procedure for the NI PXI-4130 programmable, high-power source measure unit (SMU). Calibration is generally performed at National Instruments or a metrology lab with an external high-precision digital multimeter (DMM) and some additional test equipment. For more information about calibration, visit [ni.com/calibration](http://ni.com/calibration).

## Contents

---

Conventions .....	2
Software Requirements .....	2
Documentation Requirements .....	3
Password .....	4
Calibration Interval .....	4
Test Equipment .....	4
Test Conditions .....	5
Calibration Procedures .....	5
Initial Setup .....	6
Verification .....	6
Verifying Voltage Programming Accuracy .....	8
Verifying Voltage Measurement Accuracy .....	14
Verifying Current Programming Accuracy .....	20
Verifying Current Measurement Accuracy .....	28
Adjustment .....	36
Adjusting Voltage Programming Accuracy .....	37
Adjusting Voltage Measurement Accuracy .....	43
Adjusting Current Programming Accuracy .....	49
Adjusting Current Measurement Accuracy .....	59
Appendix A: Calibration Options .....	65
Complete Calibration .....	66
Optional Calibration .....	67
Appendix B: Calibration Utilities .....	68
Calibration Function References .....	68
Where to Go for Support .....	69

# Conventions

---

The following conventions are used in this manual:

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



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

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

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

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

## Software Requirements

---

To calibrate the NI PXI-4130, you must install NI-DCPower version 1.2 or later on the calibration system. NI-DCPower includes all the functions and VIs necessary for calibration. You can download the latest version of NI-DCPower at [ni.com/idnet](http://ni.com/idnet).

NI-DCPower supports programming the calibration procedures in C and LabVIEW. For LabWindows™/CVI™, C calibration functions are installed in and are accessible from the NI-DCPower function panel, `niDCPower.fp`. For LabVIEW, calibration VIs are installed in the `niDCPower.llb` and accessible in LabVIEW from the Functions palette. Refer to Table 1 for file locations.

In this document, the LabVIEW VI is shown first, followed by the corresponding C function call. C function calls are valid for any compiler capable of calling a 32-bit DLL. Many of the functions use constants defined in the `niDCPower.h` file. To use these constants in C, you must include `niDCPower.h` in the calibration program.

For more information about calibration VIs and functions, refer to the *NI DC Power Supplies and SMUs Help*, accessible at **Start» All Programs»National Instruments»NI-DCPower»Documentation»NI DC Power Supplies and SMUs Help**.

**Table 1.** Calibration File Locations (NI-DCPower 1.2 or Later)

File Name and Location	Description
IVI\Bin\niDCPower_32.dll	NI-DCPower driver containing the entire NI-DCPower API, including calibration functions.
IVI\Lib\msc\niDCPower.lib	NI-DCPower library for Microsoft C containing the entire NI-DCPower API, including calibration functions.
<LabVIEW>\instr.lib\niDCPower Calibrate\niDCPower.llb	LabVIEW VI library containing VIs for calling the NI-DCPower calibration API. You can access calibration functions from the NI-DCPower calibration section of the LabVIEW function palette.
IVI\Drivers\niDCPower\niDCPower.fp	CVI function panel file that includes calibration function prototypes and help on using NI-DCPower in the CVI environment.
IVI\Include\niDCPower.h	NI-DCPower header file, which you must include in any C program accessing calibration functions. This file includes the entire NI-DCPower API, including calibration functions.

## Documentation Requirements

You might find the following documentation helpful as you write the calibration procedure:

- *NI PXI-4130 Specifications*
- *NI DC Power Supplies and SMUs Getting Started Guide*
- *NI DC Power Supplies and SMUs Help*, including LabVIEW VI and C function programming references

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

# Password

---

The default password for password-protected operations is NI.

# Calibration Interval

---

The measurement accuracy requirements of your application determine how often you should calibrate your device. NI recommends that you perform a complete calibration for the NI PXI-4130 at least once a year. You can shorten this calibration interval based on the accuracy demands of your application. Refer to [Appendix A: Calibration Options](#) for more information.

# Test Equipment

---

Table 2 lists the equipment required to calibrate the NI PXI-4130. If you do not have the recommended equipment, select a substitute calibration standard using the specifications listed in Table 2.

**Table 2.** Required Equipment Specifications for NI PXI-4130 Calibration

Required Equipment	Recommended Equipment	Specifications
Digital multimeter (DMM)	NI 4071	Voltage: better than $\pm 50$ ppm accuracy, better than 30 $\mu\text{V}$ resolution; Current: better than $\pm 0.04\%$ accuracy, better than 1 $\mu\text{A}$ resolution
External calibrator	Fluke 5700A/5720A	—
Auxiliary power supply	NI APS-4100	11 V to 15.5 V, 5 A
Twisted pair, shielded cabling wire	Belden 83319E 009100	18 AWG to 22 AWG

# Test Conditions

---

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

- Keep cabling wire as short as possible. Long cables and wires act as antennae, picking up extra noise that can affect measurements. To further reduce noise, twist signal/common wires together.
- Verify that all connections, including front panel connections, are secure.
- Ensure that the PXI chassis fan speed is set to HI, that the fan filters are clean, and that the empty slots contain filler panels. For more information, refer to the *Maintain Forced-Air Cooling Note to Users* document available at [ni.com/manuals](http://ni.com/manuals).
- Keep relative humidity between 10% and 90%, noncondensing.
- Maintain an ambient temperature of  $23\text{ }^{\circ}\text{C} \pm 5\text{ }^{\circ}\text{C}$ .
- Allow a warm-up time of at least 30 minutes after the NI-DCPower driver is loaded. Unless manually disabled, the NI-DCPower driver automatically loads with the operating system and enables the device. The warm-up time ensures that the measurement circuitry of the NI PXI-4130 is at a stable operating temperature.

## Calibration Procedures

---

The calibration process includes the following procedures:

1. *Initial Setup*—Install the device and configure it in Measurement & Automation Explorer (MAX).
2. *Verification*—Verify the existing operation of the device. This procedure confirms whether the device is operating within its specified range prior to calibration.
3. *Adjustment*—Perform an external adjustment of the device that adjusts the calibration constants with respect to a known voltage source. The adjustment procedure automatically stores the calibration date on the EEPROM to allow traceability.
4. *Reverification*—Repeat the verification procedure to ensure that the device is operating within its specifications after adjustment.

These procedures are described in more detail in the following sections.



**Note** The complete external calibration procedure consists of verifying the performance of the SMU, adjusting the calibration constants, and verifying performance again after the adjustments. In some cases, a complete calibration procedure may not be required. Refer to [Appendix A: Calibration Options](#) for more information.

# Initial Setup

Refer to the *NI DC Power Supplies and SMUs Getting Started Guide* for information about how to install the software and hardware and how to configure the device in MAX.

# Verification

This section describes the program you must write to verify the published specifications for the NI PXI-4130.

Verification consists of generating and measuring a series of outputs using the NI PXI-4130, verifying the accuracy with a DMM, and comparing the results to the calibration test limits. If the results fall within the test limits, the NI PXI-4130 meets its published specifications, and adjustment is optional. If the results fall outside of the test limits, you must adjust the NI PXI-4130.

Verification tests the following NI PXI-4130 specifications:

- Voltage programming accuracy
- Voltage measurement accuracy
- Current programming accuracy
- Current measurement accuracy

Tables 3 and 4 list configuration information for the calibration equipment required for verification.

**Table 3.** Calibration Equipment Configuration for Voltage Programming and Measurement Verification/Adjustment

NI PXI-4130		DMM*		
Channel(s)	Range	Function	Range <sup>†</sup>	Input Impedance <sup>†</sup>
0	6 V	DC Voltage	10 V	10 GΩ
1	6 V		10 V	10 GΩ
	20 V		100 V	10 MΩ

\* Use the highest resolution available on the DMM. The DMM should have a minimum of 6.5 digit resolution.

<sup>†</sup> Assumes an NI 4071 DMM. For all other DMMs, use the range and input impedance closest to the values listed in this table.

**Table 4.** Calibration Equipment Configuration for Current Programming and Measurement Verification/Adjustment

NI PXI-4130		DMM*				Calibrator Resistance
Channel	Range	Function	Range	Input Impedance	Resolution in Digits	
0	1 A	DC Current	1 A	N/A	6.5	N/A
1	200 $\mu$ A	DC Voltage	10 V	10 G $\Omega$	7.5	10 k $\Omega$
	2 mA	DC Voltage	10 V	10 G $\Omega$	7.5	1 k $\Omega$
	20 mA	DC Voltage	10 V	10 G $\Omega$	7.5	100 $\Omega$
	200 mA	DC Voltage	10 V	10 G $\Omega$	7.5	10 $\Omega$
	2 A	DC Current	3 A	N/A	6.5	N/A

\* Use the highest resolution available on the DMM.



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

Verification of the NI PXI-4130 is complete only after you have successfully completed all tests in this section.

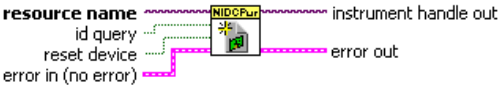


**Note** If verification fails post-adjustment, confirm that you have met the required *Test Conditions* before you return the NI PXI-4130 to NI for repair.

# Verifying Voltage Programming Accuracy

Complete the following steps to verify the voltage programming accuracy of the NI PXI-4130. Complete this test for each iteration in Table 5.

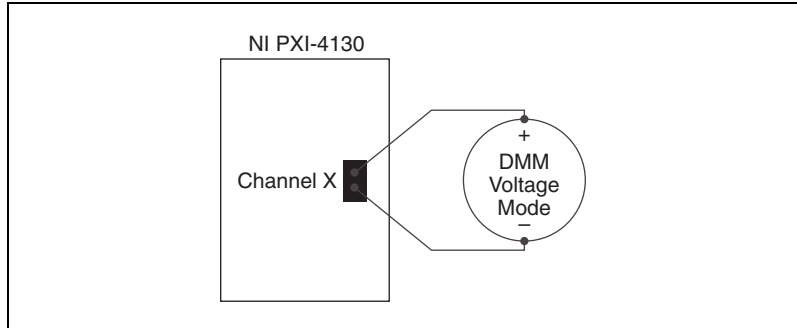
1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_init</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>idQuery:</b> <code>VI_FALSE</code></p> <p><b>resetDevice:</b> <code>VI_TRUE</code></p>

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130 as shown in Figure 1.




**Note** Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.



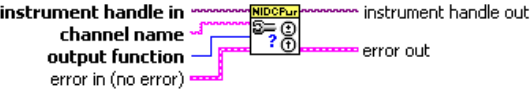
**Figure 1.** Voltage Accuracy Verification and Adjustment Setup for the NI PXI-4130

3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.

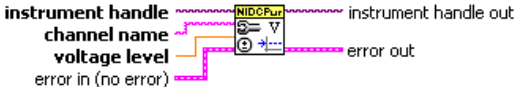
- Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

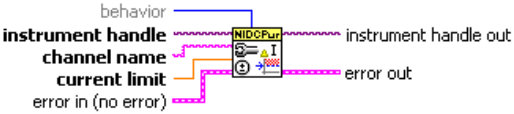
- Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputFunction</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <code>x</code></p> <p><b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>

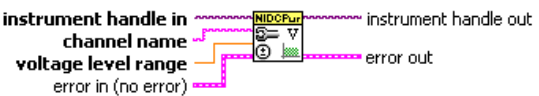
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <code>x</code></p> <p><b>level:</b> The <i>Output</i> value for the iteration of channel <code>x</code> in Table 5</p>

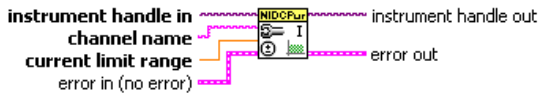
- Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimit</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>behavior:</b> <code>NIDCPOWER_VAL_CURRENT_REGULATE</code>  <b>limit:</b> 0.5</p>


- Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevelRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 5</p>


- Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimitRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> 1</p>

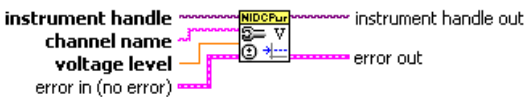
10. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureOutputEnabled</code>                      with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_TRUE</code></p>

11. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_init</code></p>

12. Wait 3 s for the output of the NI PXI-4130 to settle.
13. Measure the output voltage with the DMM.
14. Record the measurement.
15. To calculate the output error, subtract the *Output* value for the iteration of channel *x* from the measurement you recorded in step 14.
16. Compare the output error to the *Test Limit* for the iteration of channel *x* in Table 5. If the output error is outside the test limit, you must adjust the NI PXI-4130.
17. Repeat steps 3 through 16 for each iteration of channel *x* in Table 5.
18. Set the voltage level to 0 using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureVoltageLevel</code>                      with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>level:</b> 0</p>

19. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_FALSE</code></p>

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unverified channels in Table 5. When you have verified all iterations per channel, this part of the verification is complete.
22. End the session using the niDCPower Close VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_close</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

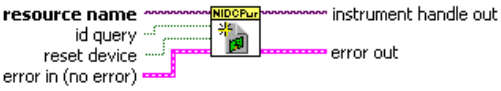
**Table 5.** NI PXI-4130 Output Parameters and Test Limits for Voltage Programming Accuracy Verification

Channel	Voltage Level Range (V)	Iteration	Output (V)	Test Limit (mV)
0	6	1	0	±4.00
		2	1.5	±4.75
		3	3	±5.50
		4	4.5	±6.25
		5	6	±7.00
1	6	1	0	±1.50
		2	1.5	±2.01
		3	3	±2.52
		4	4.5	±3.03
		5	6	±3.54
		6	-1.5	±2.01
		7	-3	±2.52
		8	-4.5	±3.03
		9	-6	±3.54
	20	1	0	±1.80
		2	5	±3.50
		3	10	±5.20
		4	15	±6.90
		5	20	±8.60
		6	-5	±3.50
		7	-10	±5.20
		8	-15	±6.90
		9	-20	±8.60

## Verifying Voltage Measurement Accuracy

Complete the following steps to verify the voltage measurement accuracy of the NI PXI-4130. Complete this test for each iteration in Table 6.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.


LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_init</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>idQuery:</b> <code>VI_FALSE</code></p> <p><b>resetDevice:</b> <code>VI_TRUE</code></p>

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130, as shown in Figure 1.

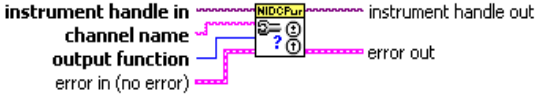


**Note** Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

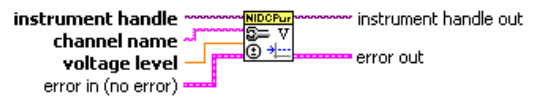
3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

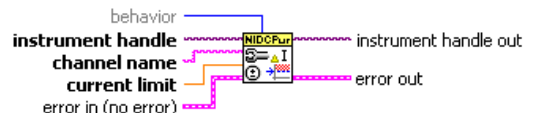
- Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutput</code> Function with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>

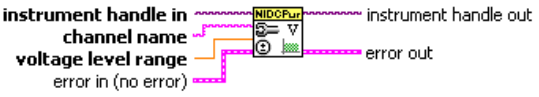
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>level:</b> The <i>Output</i> value for the iteration of channel <code>x</code> in Table 6</p>

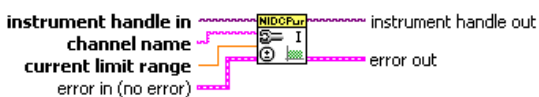
- Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimit</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>behavior:</b> <code>NIDCPOWER_VAL_CURRENT_REGULATE</code>  <b>limit:</b> <code>0.5</code></p>


- Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevelRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 6</p>

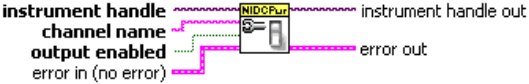
- Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimitRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> <i>l</i></p>


- Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_SetAttributeViInt32</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>attribute:</b> <code>NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE</code>  <b>value:</b> 300</p>

11. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_TRUE</code></p>

12. Apply the configuration using the niDCPower Initiate VI.

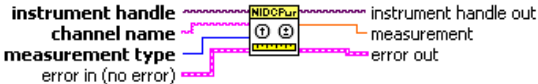
LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

13. Wait 3 s for the output of the NI PXI-4130 to settle.

14. Measure the output voltage with the DMM.

15. Record the measurement.

16. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Measure</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>measurementType:</b> <code>NIDCPOWER_VAL_MEASURE_VOLTAGE</code></p>

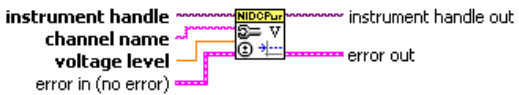
17. Record the measurement.

18. To calculate the measurement error, subtract the measurement you recorded in step 15 from the measurement you recorded in step 17.

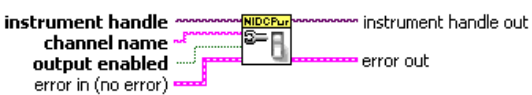
19. Calculate the upper and lower test limits using the offset and gain listed in the *Test Limit* column for the iteration of channel *x* in Table 6. Tolerances are provided instead of absolute limits because the DMM measures a unique value. Each limit is calculated by adding a

percentage of the DMM measurement and an offset voltage. Verify that the measurement error falls between the calculated limits. If the measurement error is outside the test limit, you must adjust the NI PXI-4130.

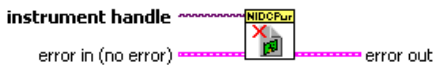
20. Repeat steps 4 through 19 for each iteration of channel  $x$  in Table 6.
21. Set the voltage level to 0 using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <math>x</math>  <b>level:</b> 0</p>

22. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <math>x</math>  <b>enabled:</b> <code>VI_FALSE</code></p>

23. Disconnect the DMM.
24. Repeat steps 2 through 23 for all unverified channels in Table 6. When you have verified all iterations per channel, this part of the verification is complete.
25. End the session using the niDCPower Close VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_close</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>


**Table 6.** NI PXI-4130 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification

<b>Channel</b>	<b>Voltage Level Range (V)</b>	<b>Iteration</b>	<b>Output (V)</b>	<b>Test Limit (V)</b>
0	6	1	0	.05% + 4 mV
		2	1.5	
		3	3	
		4	4.5	
		5	6	
1	20	1	0	.03% + 1.5 mV
		2	5	
		3	10	
		4	15	
		5	20	
		6	-5	
		7	-10	
		8	-15	
		9	-20	

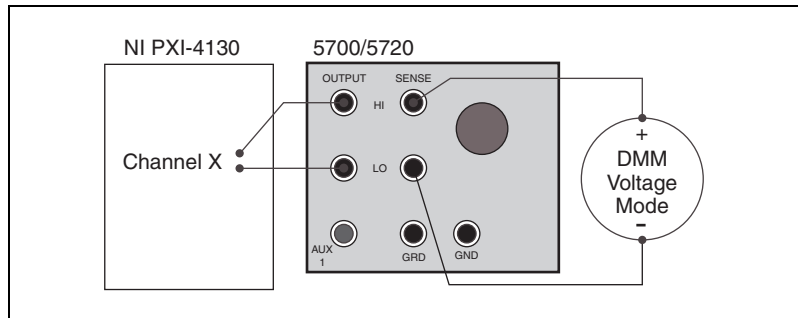
# Verifying Current Programming Accuracy

Complete the following steps to verify the current programming accuracy of the NI PXI-4130. Complete this procedure for each channel iteration in Table 7. Please verify the output accuracy in the exact order listed in Table 7 to minimize any adverse effects caused by resistor heating.

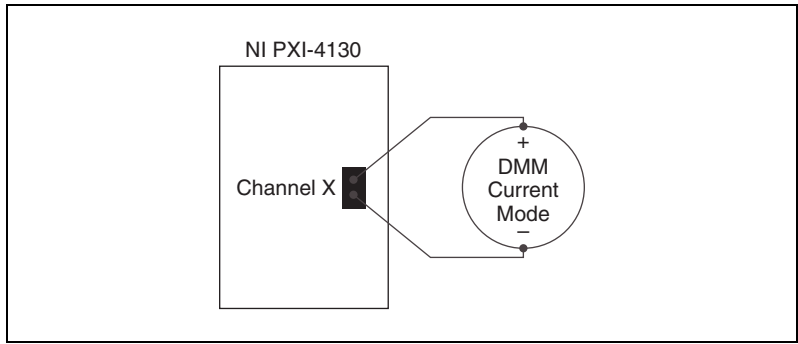
1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_init</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX  <b>idQuery:</b> <code>VI_FALSE</code>  <b>resetDevice:</b> <code>VI_TRUE</code></p>

2. Connect the NI PXI-4130 channel x to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3. The setup in Figure 2 is used for the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.



**Figure 2.** Current Programming Accuracy Verification Setup for the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA Current Ranges of Channel 1



**Figure 3.** Current Programming Accuracy Verification Setup for the 1 A Current Range of Channel 0 and the 2 A Range of Channel 1

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and iteration in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the actual resistance value displayed by the calibrator.

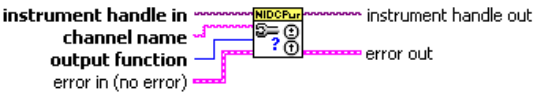


**Note** Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

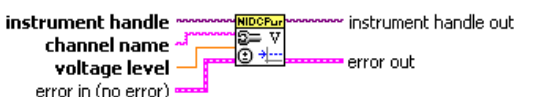
5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_Abort with the following parameter:</p> <p><b>vi:</b> The instrument handle from niDCPower_init</p>

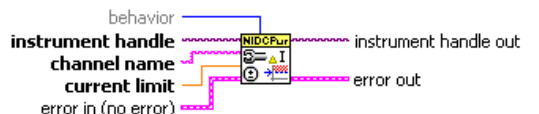
- Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutput</code> Function with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>

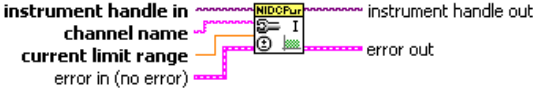
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>voltageLevel:</b> The <i>Voltage Level</i> value for the iteration of channel <code>x</code> in Table 7</p>

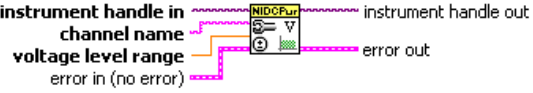
- Configure the current limit for the corresponding channel and iteration in Table 7 using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimit</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <code>x</code>  <b>currentLimit:</b> The <i>Output</i> value for the iteration of channel <code>x</code> in Table 7</p>

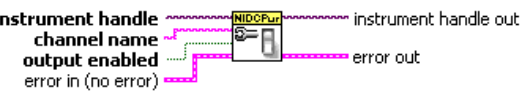
- Configure the current limit range using the niDCPower Configure Current Limit Range VI for the corresponding channel and iteration in Table 7.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle in channel name current limit range error in (no error)</p> <p>instrument handle out error out</p>	<p>Call <code>niDCPower_ConfigureCurrentLimitRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> The <i>Current Limit Range</i> value for the iteration of channel <i>x</i> in Table 7</p>

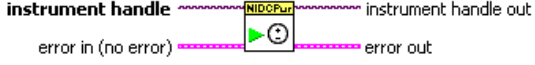
- Configure the voltage level range for the corresponding channel and iteration in Table 7 using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle in channel name voltage level range error in (no error)</p> <p>instrument handle out error out</p>	<p>Call <code>niDCPower_ConfigureVoltageLevelRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 7</p>

- Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle channel name output enabled error in (no error)</p> <p>instrument handle out error out</p>	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_TRUE</code></p>

12. Apply the configuration using the niDCPower Initiate VI.

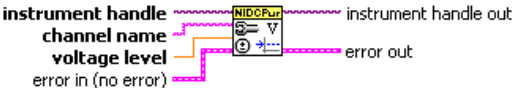
LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

13. Wait 3 s for the output of the NI PXI-4130 to settle.
14. Measure the output voltage or current using the DMM.
15. Record the measurement.
16. For the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 15 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM. Subtract the *Output* value for the iteration of channel  $x$  in Table 7 from the output current calculated above to obtain the output error.
17. Compare the output error to the *Test Limit* for the iteration of channel  $x$  in Table 7. If the output error is outside the test limit, you must adjust the NI PXI-4130.
18. Repeat steps 2 through 17 for all iterations of channel  $x$  in Table 7 per current range.



**Note** For channel 1, each current limit range has several iterations with a positive voltage level and another set of iterations with a negative voltage level.

19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <math>x</math></p> <p><b>level:</b> 0</p>

20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_FALSE</code></p>

21. Disconnect the DMM and the calibrator.
22. Repeat steps 3 through 21 for the all unverified channels in Table 7. When you have verified all iterations per channel and range, this part of the verification is complete.
23. End the session using the niDCPower Close VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_close</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

**Table 7.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Verification

Channel(s)	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)	Test Limit
0	1 A	6 V	6 V	1	50 mA	±4.08 mA
				2	350 mA	±4.53 mA
				3	700 mA	±5.05 mA
1	200 µA	20 V	20 V	1	4.00 µA	±0.101 µA
				2	50.0 µA	±0.115 µA
				3	0.10 mA	±0.130 µA
				4	0.15 mA	±0.145 µA
				5	0.20 mA	±0.160 µA
			-20 V	6	4.00 µA	±0.101 µA
				7	0.50 mA	±0.115 µA
				8	0.10 mA	±0.130 µA
				9	0.15 mA	±0.145 µA
				10	0.20 mA	±0.160 µA
	2 mA	20 V	20 V	1	40.0 µA	±1.01 µA
				2	0.50 mA	±1.15 µA
				3	1.00 mA	±1.30 µA
				4	1.50 mA	±1.45 µA
				5	2.00 mA	±1.60 µA
		-20 V	20 V	6	40.0 µA	±1.01 µA
				7	0.50 mA	±1.15 µA
				8	1.00 mA	±1.30 µA
				9	1.50 mA	±1.45 µA
				10	2.00 mA	±1.60 µA
20 mA	20 V	20 V	1	0.40 mA	±10.1 µA	
			2	5.00 mA	±11.5 µA	
			3	10.0 mA	±13.0 µA	
			4	15.0 mA	±14.5 µA	


**Table 7.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Verification (Continued)

Channel(s)	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)	Test Limit	
1	20 mA	20 V	20 V	5	20.0 mA	±16.0 μA	
			-20 V	6	0.40 mA	±10.1 μA	
				7	5.00 mA	±11.5 μA	
				8	10.0 mA	±13.0 μA	
				9	15.0 mA	±14.5 μA	
				10	20.0 mA	±16.0 μA	
			200 mA	20 V	1	4.00 mA	±0.101 mA
					2	50.0 mA	±0.115 mA
					3	100 mA	±0.130 mA
					4	150 mA	±0.145 mA
	5	200 mA			±0.160 mA		
	-20 V	6		4.00 mA	±0.101 mA		
		7		50.0 mA	±0.115 mA		
		8		100 mA	±0.130 mA		
		9		150 mA	±0.145 mA		
		10		200 mA	±0.160 mA		
	2 A	20 V	1	40.0 mA	±1.05 mA		
			2	0.5 A	±1.60 mA		
			3	1.0 A	±2.70 mA		
			4	1.5 A	±4.68 mA		
			5	2.0 A	±8.40 mA		
		-20 V	6	40.0 mA	±1.05 mA		
			7	0.5 A	±1.60 mA		
			8	1.0 A	±2.70 mA		
			9	1.5 A	±4.68 mA		
			10	2.0 A	±8.40 mA		

# Verifying Current Measurement Accuracy

Complete the following steps to verify the current measurement accuracy of the NI PXI-4130. Complete this procedure for each channel iteration per supported range in Table 8.

1. Open a session and obtain a session handle using the niDCPower Initialize VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_init</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX  <b>idQuery:</b> <code>VI_FALSE</code>  <b>resetDevice:</b> <code>VI_TRUE</code></p>

2. If the *Voltage Level* in Table 8 for this iteration of channel  $x$  is 0, skip to step 5. Do *not* connect the DMM or the Fluke 5700A/5720A calibrator to the channel  $x$  output terminals of the NI PXI-4130.

For *Output* values other than 0, connect the NI PXI-4130 channel  $x$  to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.


The setup in Figure 2 is used for the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.



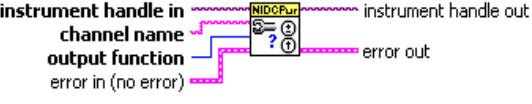
**Note** Channel  $x$  represents the channel under test. Replace the variable  $x$  in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and iteration in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the actual resistance value displayed by the calibrator.

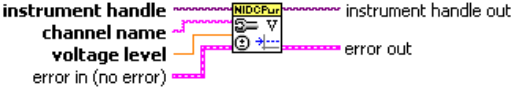
- Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle</p> <p>error in (no error)</p> <p>instrument handle out</p> <p>error out</p>	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

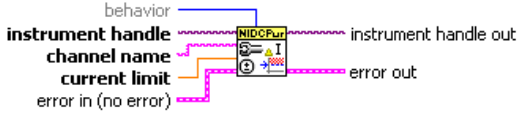
- Set the output function to voltage control using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle in</p> <p>channel name</p> <p>output function</p> <p>error in (no error)</p> <p>instrument handle out</p> <p>error out</p>	<p>Call <code>niDCPower_ConfigureOutputFunction</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <code>x</code></p> <p><b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>

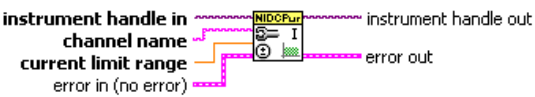
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle</p> <p>channel name</p> <p>voltage level</p> <p>error in (no error)</p> <p>instrument handle out</p> <p>error out</p>	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <code>x</code></p> <p><b>level:</b> The <i>Voltage Level</i> value for the iteration of channel <code>x</code> in Table 8</p>

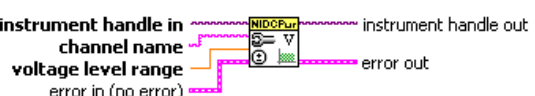
8. Configure the current limit for the corresponding channel and iteration in Table 8 using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureCurrentLimit  with the following parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_init  <b>channelName:</b> <i>x</i>  <b>currentLimit:</b> The <i>Output</i> value for the  iteration of channel <i>x</i> in Table 8</p>

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI for the corresponding channel and iteration in Table 8.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_ConfigureCurrent  LimitRange with the following  parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_init  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> The <i>Current Limit  Range</i> value for the iteration of channel <i>x</i>  in Table 8</p>

10. Configure the voltage level range for the corresponding channel and iteration in Table 8 using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_ConfigureVoltage  LevelRange with the following  parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_init  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level  Range</i> value for the iteration of channel <i>x</i>  in Table 8</p>

11. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_SetAttribute ViInt32</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>attribute:</b>  <code>NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE</code>  <b>value:</b> 300</p>

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_TRUE</code></p>

13. Apply the configuration using the niDCPower Initiate VI.

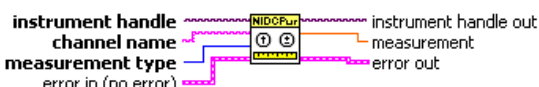
LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

14. Wait 3 s for the output of the NI PXI-4130 to settle.

15. If the *Output* value for this iteration of channel *x* is 0, check that the DMM or calibrator is disconnected from the NI PXI-4130 and skip to step 16.


For all other *Output* values, measure the output voltage or current using the DMM.

16. Record the measurement. If the *Voltage Level* value for this iteration of channel  $x$  is 0, do not take a measurement; record “0” in place of the measurement.
17. For the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM in the previous step.
18. Measure the output current using the niDCPower Measure VI.


LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Measure</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p> <p><b>channelName:</b> <math>x</math></p> <p><b>measurementType:</b> <code>NIDCPOWER_VAL_MEASURE_CURRENT</code></p>

19. Record the measurement.
20. To calculate the measurement error, subtract the measurement you calculated in step 17 from the measurement you recorded in step 19.
21. Calculate the upper and lower test limits using the offset and gain listed in the *Test Limits* column for the iteration of channel  $x$  in Table 8. Tolerances are provided instead of absolute limits because the DMM measures a unique value. Each limit is calculated by adding a percentage of the actual output current and an offset current. Use the measurement from step 17 to calculate the test limits. Verify that the measurement error falls between the calculated limits. If the measurement error is outside the test limits, you must adjust the NI PXI-4130.
22. Repeat steps 2 through 21 for all iterations of channel  $x$  per supported current range in Table 8.

23. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows a LabVIEW VI icon labeled 'niDCPower Configure Voltage Level'. It has four input terminals on the left: 'instrument handle', 'channel name', 'voltage level', and 'error in (no error)'. It has two output terminals on the right: 'instrument handle out' and 'error out'.</p>	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>level:</b> 0</p>


24. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows a LabVIEW VI icon labeled 'niDCPower Configure Output Enabled'. It has four input terminals on the left: 'instrument handle', 'channel name', 'output enabled', and 'error in (no error)'. It has two output terminals on the right: 'instrument handle out' and 'error out'.</p>	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_FALSE</code></p>

25. Disconnect the DMM and the calibrator.

26. Repeat steps 3 through 25 for the all unverified channels in Table 8. When you have verified all iterations per channel and range, this part of the verification is complete.

27. End the session using the niDCPower Close VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows a LabVIEW VI icon labeled 'niDCPower Close'. It has two input terminals on the left: 'instrument handle' and 'error in (no error)'. It has one output terminal on the right: 'error out'.</p>	<p>Call <code>niDCPower_close</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_init</code></p>

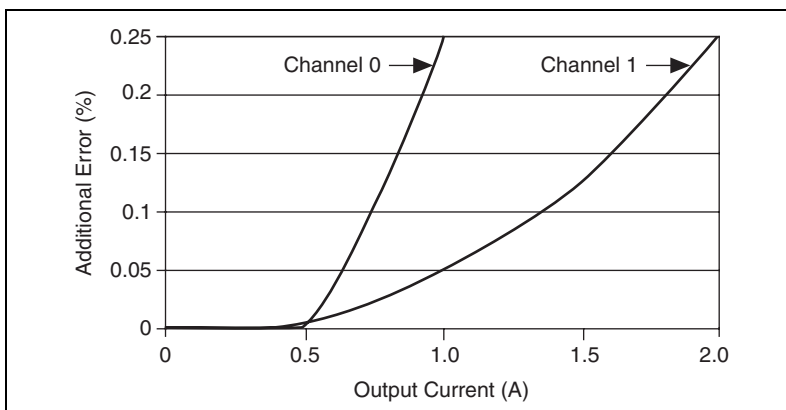
**Table 8.** NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Verification

Channel(s)	Current Limit Range	Voltage Level Range (V)	Voltage Level (V)	Iteration	Output	Test Limit ± (% of reading + offset)			
0	1 A	6	0	1	350 mA	.15% + 4 mA <sup>1</sup>			
			6	2	350 mA				
				3	700 mA				
1	200 μA	20	0	1	50.0 μA	.03% + 0.02 μA			
			20	2	50.0 μA				
				3	0.10 mA				
				4	0.150 mA				
				5	0.20 mA				
			-20	6	50.0 μA				
				7	0.10 mA				
				8	0.15 mA				
				9	0.2.0 mA				
			2 mA	20	0		1	0.50 mA	.03% + 0.2 μA
					20		2	0.50 mA	
							3	1.00 mA	
	4	1.50 mA							
	5	2.00 mA							
	-20	6				0.50 mA			
		7			1.00 mA				
		8			1.50 mA				
		9			2.00 mA				
	20 mA	20	0	1	5.00 mA	.03% + 2.0 μA			
			20	2	5.00 mA				
				3	10.0 mA				
4				15.0 mA					
5				20.0 mA					
-20				6	5.00 mA				

**Table 8.** NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Verification (Continued)

Channel(s)	Current Limit Range	Voltage Level Range (V)	Voltage Level (V)	Iteration	Output	Test Limit ± (% of reading + offset)			
1	20 mA	20 V	-20	7	10.0 mA	.03% + 2.0 μA			
				8	15.0 mA				
				9	20.0 mA				
	200 mA		0	1	50.0 mA	.03% + 40 μA			
				2	50.0 mA				
				3	100 mA				
				4	150 mA				
				5	200 mA				
				-20	6		50.0 mA		
					7		100 mA		
					8		150 mA		
					9		200 mA		
				2 A	0		1	0.5 A	.12% + 200 μA <sup>1</sup>
							2	0.5 A	
							3	1.0 A	
	4		1.5 A						
	5		2.0 A						
	-20 V		6		0.5 A				
			7		1.0 A				
			8		1.5 A				
			9		2.0 A				

<sup>1</sup> For currents ≥ 500 mA, refer to the additional derating information in Figure 4, *Accuracy Derating versus Load Current*.



**Figure 4.** Accuracy Derating versus Load Current

## Adjustment

Adjustment improves the accuracy of the NI PXI-4130 and updates the calibration date and temperature in the EEPROM. Perform an adjustment once a year or when the accuracy of NI PXI-4130 is outside the calibration test limits.

Adjustment corrects the following NI PXI-4130 specifications:

- Voltage programming accuracy
- Voltage measurement accuracy
- Current programming accuracy
- Current measurement accuracy



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




**Note** If the NI PXI-4130 has passed initial verification and is within all test limits, NI recommends, but does not require, an adjustment to guarantee the published specifications for the next year. If you choose to skip adjustment, run the niDCPower Initialize External Calibration VI and end with the niDCPower Close External Calibration VI with **action** set to **Commit** to update the calibration date and onboard calibration temperature without making any adjustments to the device.

After adjustment, repeat the *Verification* section to verify that the adjustment was successful.

## Adjusting Voltage Programming Accuracy

Complete the following steps to adjust the voltage programming accuracy of the NI PXI-4130. Complete this test for each iteration in Table 9.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.


LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_InitExtCal</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>password:</b> NI</p>

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4130, as shown in Figure 1.

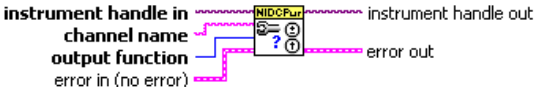


**Note** Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

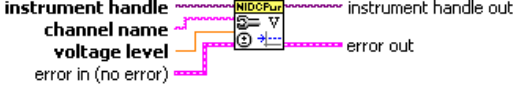
3. Configure the DMM for the range and mode listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p>

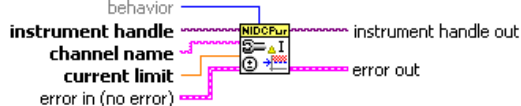
5. Configure the output function using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputFunction</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <i>x</i></p> <p><b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>

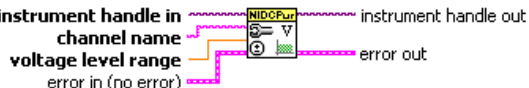
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <i>x</i></p> <p><b>level:</b> The <i>Output</i> value for the iteration of channel <i>x</i> in Table 9</p>

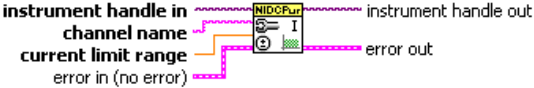
- Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimit</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <i>x</i></p> <p><b>behavior:</b> <code>NIDCPOWER_VAL_CURRENT_REGULATE</code></p> <p><b>limit:</b> 0.5</p>


- Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevelRange</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <i>x</i></p> <p><b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 9</p>


- Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle in channel name current limit range error in (no error)</p> <p>instrument handle out error out</p>	<p>Call niDCPower_ConfigureCurrentLimitRange with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal <b>channelName:</b> <i>x</i> <b>currentLimitRange:</b> 1</p>

- Enable the output using the niDCPower Configure Output Enabled VI.


LabVIEW VI	C/C++ Function Call
 <p>instrument handle channel name output enabled error in (no error)</p> <p>instrument handle out error out</p>	<p>Call niDCPower_ConfigureOutputEnabled with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal <b>channelName:</b> <i>x</i> <b>enabled:</b> VI_TRUE</p>

- Apply the configuration using the niDCPower Initiate VI.


LabVIEW VI	C/C++ Function Call
 <p>instrument handle error in (no error)</p> <p>instrument handle out error out</p>	<p>Call niDCPower_Initiate with the following parameter:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal</p>

- Wait 3 s for the output of the NI PXI-4130 to settle.
- Measure the output voltage with the DMM.
- Record the measurement.
- Repeat steps 3 through 14 for each iteration of channel *x* in Table 9.

16. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>level:</b> <code>0</code></p>

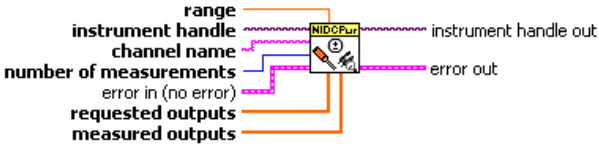
17. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>enabled:</b> <code>VI_FALSE</code></p>

18. Adjust the voltage level using the niDCPower Cal Adjust Voltage Level VI for every voltage level range of channel  $x$ .

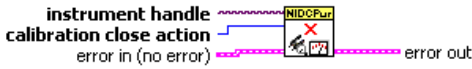


**Note** When calibrating channel 1, each voltage level range requires a unique call to the niDCPower Cal Adjust Voltage Level VI. Positive and negative output values within the same range must be calibrated using unique calls as well.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CalAdjustVoltageLevel</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <math>x</math></p> <p><b>range:</b> The voltage level range of channel <math>x</math></p> <p><b>numberOfMeasurements:</b> An integer value of the total number of measurements. This value should match the number of elements in the <b>requestedOutputs</b> and <b>measuredOutputs</b> arrays</p> <p><b>requestedOutputs:</b> An array composed of the <i>Output</i> values for each iteration of channel <math>x</math> in Table 9 for the range to be calibrated</p> <p><b>measuredOutputs:</b> An array composed of the measurement values you recorded in step 14 for each iteration of channel <math>x</math> in Table 9 for the range to be calibrated</p>

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels in Table 9. When you have adjusted all voltage measurements on all channels, this part of the adjustment is complete.

21. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CloseExtCal</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>action:</b> The instrument handle from <code>niDCPower_VAL_COMMIT</code></p>


**Table 9.** NI PXI-4130 Output Parameters for Voltage Programming Accuracy Adjustment

Channel	Iteration	Voltage Level Range (V)	Output
0	1	6	0 V
	2		3 V
	3		6 V
1	1	6	1 mV
	2		3 V
	3		6 V
	1		-0.1 mV
	2		-3 V
	3		-6 V
	1	20	1 mV
	2		10 V
	3		20 V
	1		-1 mV
	2		-10 V
	3		-20 V

## Adjusting Voltage Measurement Accuracy

Complete the following steps to adjust the voltage measurement accuracy of the NI PXI-4130. Complete this test for each iteration in Table 10.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.


LabVIEW VI	C/C++ Function Call
 <p>The LabVIEW VI icon for <code>niDCPower_InitExtCal</code> shows two input wires: a blue wire labeled "resource name" and a red wire labeled "password". Below the "password" wire is the text "error in (no error)". The icon itself is a yellow rectangle with "niDCPower" and a gear icon. Two output wires emerge from the right: a blue wire labeled "instrument handle out" and a red wire labeled "error out".</p>	<p>Call <code>niDCPower_InitExtCal</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>password:</b> NI</p>

2. Connect the DMM to the channel  $x$  output terminals of the NI PXI-4130, as shown in Figure 1.



**Note** Channel  $x$  represents the channel under test. Replace the variable  $x$  in the program with the actual channel name.

3. Configure the DMM for the mode and range listed for the corresponding channel and range in Table 3.
4. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
 <p>The LabVIEW VI icon for <code>niDCPower_Abort</code> shows one input wire: a blue wire labeled "instrument handle". Below it is the text "error in (no error)". The icon is a yellow rectangle with "niDCPower" and a red square with a white circle containing a minus sign. Two output wires emerge from the right: a blue wire labeled "instrument handle out" and a red wire labeled "error out".</p>	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p>

- Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutput</code> Function with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>outputFunction:</b> <code>NIDCPOWER_VAL_DC_VOLTAGE</code></p>


- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>level:</b> The <i>Output</i> value for the iteration of channel <code>x</code> in Table 10</p>

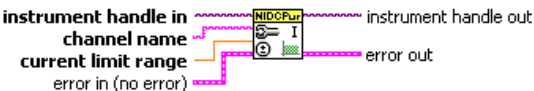
- Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureCurrentLimit</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>behavior:</b> <code>NIDCPOWER_VAL_CURRENT_REGULATE</code>  <b>limit:</b> <code>0.5</code></p>


- Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureVoltageLevelRange with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 10</p>


- Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureCurrentLimitRange with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> 1</p>


- Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_SetAttributeViInt32 with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>attribute:</b>  NIDCPower_ATTR_SAMPLES_TO_AVERAGE  <b>value:</b> 300</p>

11. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle channel name output enabled error in (no error)</p> <p>instrument handle out error out</p>	<p>Call niDCPower_ConfigureOutputEnabled with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal <b>channelName:</b> <i>x</i> <b>enabled:</b> VI_TRUE</p>

12. Apply the configuration using the niDCPower Initiate VI.

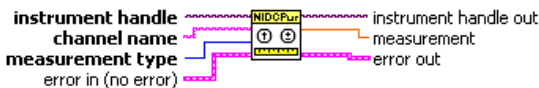
LabVIEW VI	C/C++ Function Call
 <p>instrument handle error in (no error)</p> <p>instrument handle out error out</p>	<p>Call niDCPower_Initiate with the following parameter:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal</p>

13. Wait 3 s for the output of the NI PXI-4130 to settle.

14. Measure the output voltage with the DMM.

15. Record the measurement.


16. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
 <p>instrument handle channel name measurement type error in (no error)</p> <p>instrument handle out measurement error out</p>	<p>Call niDCPower_Measure with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal <b>channelName:</b> <i>x</i> <b>measurementType:</b> NIDCPOWER_VAL_MEASURE_VOLTAGE</p>


17. Record the measurement.

18. Repeat steps 3 through 17 for each iteration of channel *x* in Table 10.

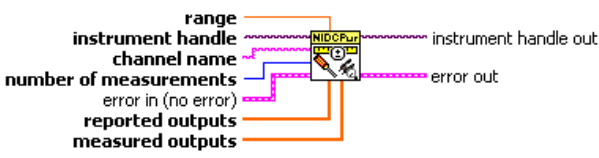
19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureVoltageLevel</code>            with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <i>x</i>  <b>level:</b> 0</p>

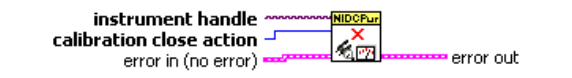
20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureOutputEnabled</code>            with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_FALSE</code></p>

21. Adjust the voltage measurement using the niDCPower Cal Adjust Voltage Measurement VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CalAdjustVoltageMeasurement</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <math>x</math></p> <p><b>range:</b> The voltage range of channel <math>x</math></p> <p><b>numberOfMeasurements:</b> 3</p> <p><b>reportedOutputs:</b> An array composed of the measurements you took with the NI PXI-4130 and recorded in step 17 for each iteration of channel <math>x</math></p> <p><b>measuredOutputs:</b> An array composed of the measurements you took with the DMM and recorded in step 15 for each iteration of channel <math>x</math></p>

22. Disconnect the DMM.
23. Repeat steps 2 through 22 for all unadjusted channels in Table 10. When you have adjusted voltage measurements on all channels, this part of the adjustment is complete.
24. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CloseExtCal</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>action:</b> The instrument handle from <code>niDCPower_VAL_COMMIT</code></p>


**Table 10.** NI PXI-4130 Output Parameters for Voltage Measurement Accuracy Adjustment

Channel	Iteration	Voltage Level Range (V)	Output (V)
0	1	6	0
	2		3
	3		6
1	1	20	-20
	2		0
	3		20

## Adjusting Current Programming Accuracy

Complete the following steps to adjust the current programming accuracy of the NI PXI-4130. Complete this test for each channel iteration in Table 11.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
 <p>resource name password error in (no error)</p> <p>niDCPower InitExtCal</p> <p>instrument handle out error out</p>	<p>Call <code>niDCPower_InitExtCal</code> with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>password:</b> NI</p>

2. Connect the NI PXI-4130 channel  $x$  to the DMM or to the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3. The setup in Figure 2 is used for the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.



**Note** Channel  $x$  represents the channel under test. Replace the variable  $x$  in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.

4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value for the corresponding channel and range in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the resistance value displayed by the calibrator.
5. Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_Abort with the following parameter:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal</p>

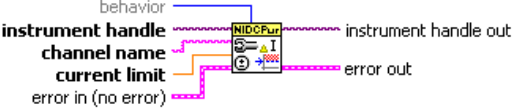
6. Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_ConfigureOutputFunction with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> x  <b>outputFunction:</b> NIDCPOWER_VAL_DC_VOLTAGE</p>

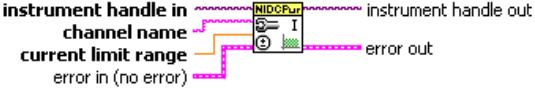
7. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call niDCPower_ConfigureVoltageLevel with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> x  <b>voltageLevel:</b> The <i>Voltage Level</i> value for the iteration of channel x in Table 11</p>

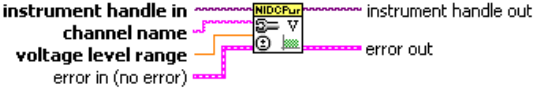
- Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureCurrentLimit  with the following parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>limit:</b> The <i>Output</i> value for the iteration of  channel <i>x</i> in Table 11</p>

- Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureCurrentLimit  Range with the following parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> The <i>Current Limit  Range</i> value for the iteration of channel <i>x</i>  in Table 11</p>

- Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  niDCPower_ConfigureVoltageLevel  Range with the following parameters:</p> <p><b>vi:</b> The instrument handle from  niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level  Range</i> value for the iteration of channel <i>x</i>  in Table 11</p>

11. Enable current overranging using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_SetAttribute</code> <code>ViInt32</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>attribute:</b> <code>NIDCPOWER_ATTR_OVERRANGING_ENABLED</code>  <b>value:</b> <code>VI_TRUE</code></p>

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>enabled:</b> <code>VI_TRUE</code></p>

13. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p>

14. Wait 3 s for the output of the NI PXI-4130 to settle.

15. Measure the output voltage or current using the DMM.

16. Record the measurement.

17. For the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16 by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM.
18. Repeat steps 2 through 17 for all iterations of channel  $x$  in Table 11 per current range.



**Note** For channel 1, each current limit range has several iterations with a positive voltage level and another set of iterations with a negative voltage level.

19. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureVoltageLevel</code>            with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <math>x</math>  <b>voltageLevel:</b> 0</p>

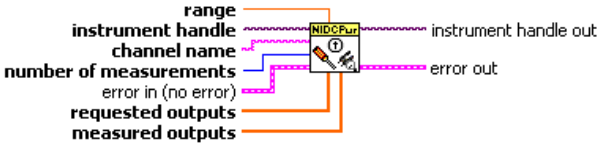
20. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call  <code>niDCPower_ConfigureOutputEnabled</code>            with the following parameters:</p> <p><b>vi:</b> The instrument handle from  <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <math>x</math>  <b>enabled:</b> <code>VI_FALSE</code></p>

21. Adjust the current limit using the niDCPower Cal Adjust Current Limit VI for each current limit range listed in Table 11.

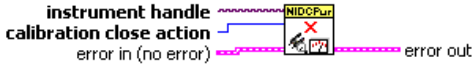


**Note** When calibrating channel 1, each current level range requires a unique call to the niDCPower Cal Adjust Current Level VI. Positive and negative output values within the same range must be calibrated using unique calls as well.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>CalAdjustCurrentLimit</code> with the following parameters:</p> <ul style="list-style-type: none"> <li><b>vi</b>: The instrument handle from <code>niDCPower_InitExtCal</code></li> <li><b>channelName</b>: <math>x</math></li> <li><b>range</b>: The <i>Current Limit Range</i> of channel <math>x</math></li> <li><b>numberOfMeasurements</b>: An integer value of the total number of measurements. This value should match the number of elements in the <b>requestedOutputs</b> and <b>measuredOutputs</b> arrays</li> <li><b>requestedOutputs</b>: An array composed of the <i>Output</i> values for each iteration of channel <math>x</math> in Table 11 for the range to be calibrated</li> <li><b>measuredOutputs</b>: An array composed of the measurements you recorded in step 16 for each iteration of channel <math>x</math> in Table 11 for the range to be calibrated</li> </ul>

22. Repeat steps 2 through 21 for all current ranges of channel  $x$  in Table 11.
23. Disconnect the DMM and the calibrator.
24. Repeat steps 2 through 23 for all unadjusted channels in Table 11. When you have adjusted current programming on all channels, this part of the adjustment is complete.

25. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CloseExtCal</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>action:</b> The instrument handle from <code>niDCPower_VAL_COMMIT</code></p>

**Table 11.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
0	1 A	6 V	6 V	1	20 mA
				2	350 mA
				3	700 mA
1	200 $\mu$ A	20 V	20 V	1	2.00 $\mu$ A
				2	24.0 $\mu$ A
				3	46.0 $\mu$ A
				4	68.0 $\mu$ A
				5	90.0 $\mu$ A
				6	112 $\mu$ A
				7	134 $\mu$ A
				8	156 $\mu$ A
				9	178 $\mu$ A
				10	200 $\mu$ A
			-20 V	11	2.00 $\mu$ A
				12	24.0 $\mu$ A
				13	46.0 $\mu$ A
				14	68.0 $\mu$ A
				15	90.0 $\mu$ A
				16	112 $\mu$ A
				17	134 $\mu$ A

**Table 11.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
1	200 $\mu$ A	20 V	-20 V	18	156 $\mu$ A
				19	178 $\mu$ A
				20	200 $\mu$ A
	2 mA		20 V	1	20.0 $\mu$ A
				2	0.240 mA
				3	0.460 mA
				4	0.680 mA
				5	0.900 mA
				6	1.12 mA
				7	1.34 mA
				8	1.56 mA
				9	1.78 mA
				10	2.00 mA
	20 mA		-20 V	11	20.0 $\mu$ A
				12	0.240 mA
				13	0.460 mA
				14	0.680 mA
				15	0.900 mA
				16	1.12 mA
				17	1.34 mA
18		1.56 mA			
19		1.78 mA			
20		2.00 mA			
20 mA	20 V	1	0.200 mA		
		2	2.40 mA		
		3	4.60 mA		
		4	6.80 mA		
		5	9.00 mA		
		6	11.2 mA		

**Table 11.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output
1	20 mA	20 V	20 V	7	13.4 mA
				8	15.6 mA
				9	17.8 mA
				10	20.0 mA
			-20 V	11	0.200 mA
				12	2.40 mA
				13	4.60 mA
				14	6.80 mA
				15	9.00 mA
				16	11.2 mA
	20 V	17	13.4 mA		
		18	15.6 mA		
		19	17.8 mA		
		20	20.0 mA		
	200 mA	20 V	20 V	1	0.200 mA
				2	2.40 mA
				3	4.60 mA
				4	6.80 mA
				5	9.00 mA
				6	11.2 mA
7				13.4 mA	
8				15.6 mA	
9				17.8 mA	
10				20.0 mA	
-20 V	20 V	-20 V	11	2.00 mA	
			12	24.0 mA	
			13	46.0 mA	
			14	68.0 mA	
			15	90.0 mA	
			16	112 mA	


**Table 11.** NI PXI-4130 Output Parameters and Test Limits for Current Programming Accuracy Adjustment (Continued)

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output			
1	200 mA	20 V	-20 V	17	134 mA			
				18	156 mA			
				19	178 mA			
				20	200 mA			
	2 A			20 V	1	20.0 mA		
					2	240 mA		
					3	460 mA		
					4	680 mA		
					5	900 mA		
					6	1.12 A		
					7	1.34 A		
					8	1.56 A		
				-20 V			9	1.78 A
							10	2.00 A
							11	20.0 mA
							12	240 mA
							13	460 mA
							14	680 mA
							15	900 mA
							16	1.12 A
17	1.34 A							
18	1.56 A							
19	1.78 A							
20	2.00 A							

## Adjusting Current Measurement Accuracy

Complete the following steps to adjust the current measurement accuracy of the NI PXI-4130. Complete this test for each channel iteration in Table 12.

1. Open a session and obtain a session handle using the niDCPower Initialize External Calibration VI.

LabVIEW VI	C/C++ Function Call
 <p>resource name password error in (no error)</p> <p>niDCPower 3%</p> <p>instrument handle out error out</p>	<p>Call niDCPower_InitExtCal with the following parameters:</p> <p><b>resourceName:</b> The device name assigned by MAX</p> <p><b>password:</b> NI</p>

2. If the *Voltage Level* value for this iteration of channel  $x$  is 0, skip to step 5. Do not connect the DMM or the Fluke 5700A/5720A calibrator to the channel  $x$  output terminals of the NI PXI-4130.

For all other *Output* values, connect the NI PXI-4130 channel  $x$  to the DMM or the Fluke 5700A/5720A calibrator, as illustrated in Figure 2 or Figure 3.

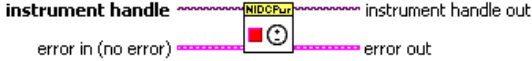
The setup in Figure 2 is used for the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1. The setup in Figure 3 is used for the 1 A current range of channel 0 and the 2 A current range of channel 1.



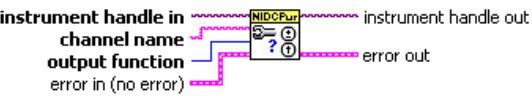
**Note** Channel  $x$  represents the channel under test. Replace the variable  $x$  in the program with the actual channel name.

3. Configure the DMM to the mode and range listed for the corresponding channel and range in Table 4.
4. When applicable, configure the Fluke 5700/5720A calibrator to the *Resistance* value listed for the corresponding channel and range in Table 4. Enable external sense (4-wire mode) on the calibrator. Record the resistance value displayed by the calibrator.

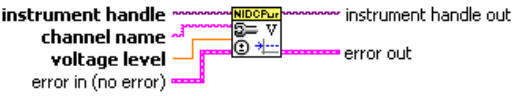
- Place the NI PXI-4130 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Abort</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p>

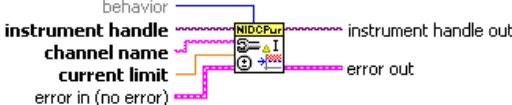
- Set the output function to DC Voltage using the niDCPower Configure Output Function VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputFunction</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>outputFunction:</b> <code>NIDCPOWER_VAL_DCVOLTAGE</code></p>

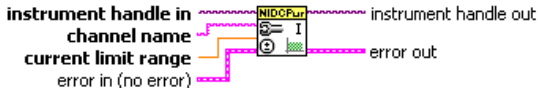
- Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <code>x</code>  <b>level:</b> The <i>Voltage Level</i> value for channel <code>x</code> in Table 12</p>

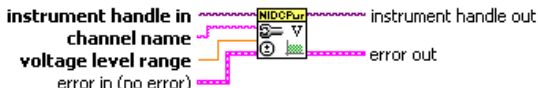
8. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows the 'niDCPower Configure Current Limit VI' block. It has five input terminals on the left: 'behavior' (blue), 'instrument handle' (purple), 'channel name' (orange), 'current limit' (green), and 'error in (no error)' (pink). It has two output terminals on the right: 'instrument handle out' (purple) and 'error out' (pink). The block icon shows a '5' in a circle and a '1' in a square.</p>	<p>Call  niDCPower_ConfigureCurrentLimit  with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>behavior:</b>  NIDCPOWER_VAL_CURRENT_REGULATE  <b>limit:</b> The <i>Output</i> value for the iteration of channel <i>x</i> in Table 12</p>

9. Configure the current limit range using the niDCPower Configure Current Limit Range VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows the 'niDCPower Configure Current Limit Range VI' block. It has four input terminals on the left: 'instrument handle in' (purple), 'channel name' (orange), 'current limit range' (green), and 'error in (no error)' (pink). It has two output terminals on the right: 'instrument handle out' (purple) and 'error out' (pink). The block icon shows a '5' in a circle and a '1' in a square.</p>	<p>Call  niDCPower_ConfigureCurrentLimit  Range with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>currentLimitRange:</b> The <i>Current Limit Range</i> value for the iteration of channel <i>x</i> in Table 12</p>

10. Configure the voltage level range using the niDCPower Configure Voltage Level Range VI.

LabVIEW VI	C/C++ Function Call
 <p>The diagram shows the 'niDCPower Configure Voltage Level Range VI' block. It has four input terminals on the left: 'instrument handle in' (purple), 'channel name' (orange), 'voltage level range' (green), and 'error in (no error)' (pink). It has two output terminals on the right: 'instrument handle out' (purple) and 'error out' (pink). The block icon shows a '5' in a circle and a 'V' in a square.</p>	<p>Call  niDCPower_ConfigureVoltageLevel  Range with the following parameters:</p> <p><b>vi:</b> The instrument handle from niDCPower_InitExtCal  <b>channelName:</b> <i>x</i>  <b>voltageLevelRange:</b> The <i>Voltage Level Range</i> value for the iteration of channel <i>x</i> in Table 12</p>

11. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_SetAttribute ViInt32</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <math>x</math>  <b>attribute:</b>  <code>NIDCPOWER_ATTR_SAMPLES_TO_AVERAGE</code>  <b>value:</b> 300</p>

12. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutput Enabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <math>x</math>  <b>enabled:</b> <code>VI_TRUE</code></p>

13. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Initiate</code> with the following parameter:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p>

14. Wait 3 s for the output of the NI PXI-4130 to settle.

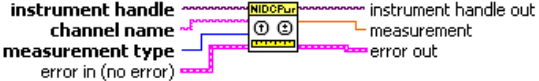
15. If the *Output* value for this iteration of channel  $x$  is 0, skip to step 16. For all other *Output* values, measure the output voltage or current using the DMM.

16. Record the measurement. If the *Voltage Level* value for this iteration of channel  $x$  is 0, you did not take a measurement; record “0” in place of the measurement.

17. For the 200  $\mu$ A, 2 mA, 20 mA, and 200 mA current ranges of channel 1, divide the voltage measurement you recorded in step 16

by the resistance measurement you recorded in step 4 to calculate the output current. For the 1 A current range of channel 0 and the 2 A current range of channel 1, the output current is measured directly by the DMM in step 15.


18. Measure the output current using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_Measure</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <i>x</i>  <b>measurementType:</b>  <code>NIDCPOWER_VAL_MEASURE_CURRENT</code></p>

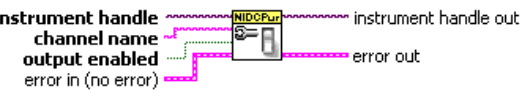
19. Record the measurement.

20. Repeat steps 2 through 19 for all iterations of channel *x* in Table 12 per current range.

21. Set the voltage level to 0 V using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureVoltageLevel</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <i>x</i>  <b>voltageLevel:</b> 0</p>

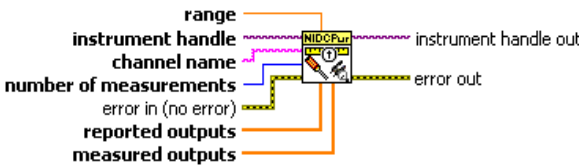
22. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_ConfigureOutputEnabled</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code>  <b>channelName:</b> <i>x</i>  <b>enabled:</b> <code>VI_FALSE</code></p>

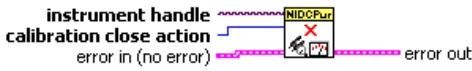
23. Adjust the current measurement using the niDCPower Cal Adjust Current Measurement VI.



**Note** When calibrating channel 1, each current limit range requires a unique call to the niDCPower Cal Adjust Current Measurement VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CalAdjustCurrent</code> Measurement with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>channelName:</b> <math>x</math></p> <p><b>range:</b> The voltage range of channel <math>x</math></p> <p><b>numberOfMeasurements:</b> 3</p> <p><b>reportedOutputs:</b> An array composed of the measurements you took with the NI PXI-4130 and recorded in step 19 for each iteration of channel <math>x</math> in Table 12 for the range to be calibrated</p> <p><b>measuredOutputs:</b> An array composed of the measurements you took with the DMM and recorded in step 16 for each iteration of channel <math>x</math> in Table 12 for the range to be calibrated</p>

24. Disconnect the DMM and the calibrator.
25. Repeat steps 2 through 24 for all unadjusted channels in Table 12. When you have adjusted current measurement on all channels, this part of the adjustment is complete.
26. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	<p>Call <code>niDCPower_CloseExtCal</code> with the following parameters:</p> <p><b>vi:</b> The instrument handle from <code>niDCPower_InitExtCal</code></p> <p><b>action:</b> The instrument handle from <code>niDCPower_VAL_COMMIT</code></p>

When you have successfully completed all adjustment tests, adjustment of the NI PXI-4130 is complete. Repeat the *Verification* section to reverify the performance of the NI PXI-4130 post-adjustment. If the NI PXI-4130 successfully passes all verification tests, calibration is complete.

**Table 12.** NI PXI-4130 Output Parameters and Test Limits for Current Measurement Accuracy Adjustment

Channel	Current Limit Range	Voltage Level Range	Voltage Level	Iteration	Output (A)
0	1 A	6 V	0 V	1	3.50E-01
			6 V	2	3.50E-01
				3	7.00E-01
1	200 $\mu$ A	20 V	0 V	1	2.00E-04
			20 V	2	
				3	
	2 mA		0 V	1	2.00E-03
			20 V	2	
				3	
	20 mA		0 V	1	2.00E-02
			20 V	2	
				3	
	200 mA		0 V	1	2.00E-01
			20 V	2	
				3	
	2 A		0 V	1	8.00E-01
			20 V	2	
				3	

## Appendix A: Calibration Options

Calibration involves verification, and, if necessary, adjustment, and reverification of the NI PXI-4130.

*Verification* is the process of testing to ensure that the accuracy of the device is within certain specifications, or calibration test limits. Perform verification to determine if the device requires adjustment, or perform verification post-adjustment to determine if the adjustment was successful.



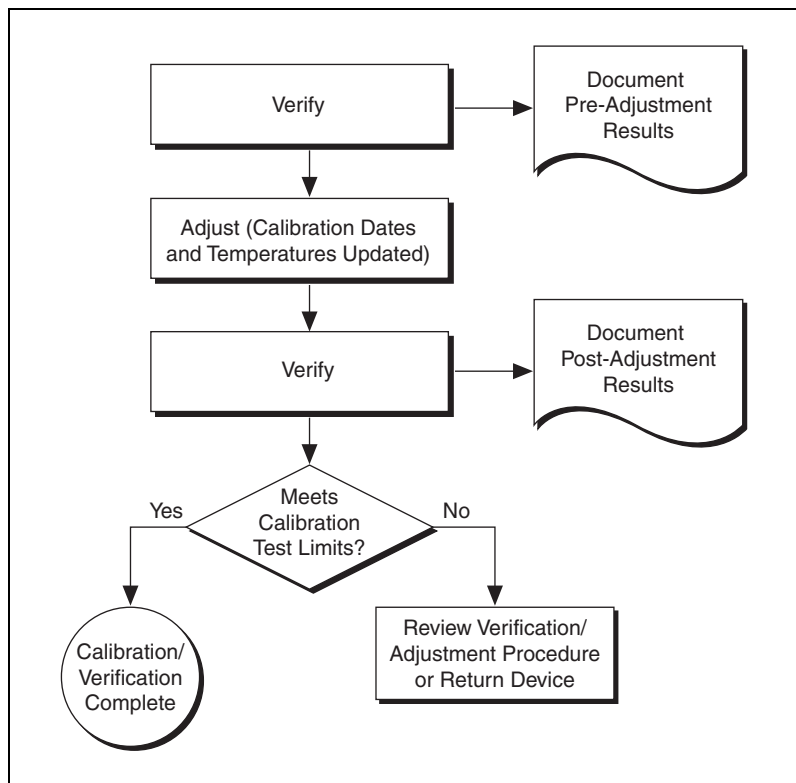
**Note** In this document, the calibration test limits are the published *NI PXI-4130 Specifications*.

*Adjustment* is the process of measuring and compensating for device performance to improve the measurement accuracy. Performing an adjustment updates the calibration date.

The *Complete Calibration* section details the recommended calibration procedure. If the device meets its calibration test limits and you prefer to skip adjustment, the *Optional Calibration* section details alternative calibration procedures.

## Complete Calibration

Perform a complete calibration to guarantee that the NI PXI-4130 meets or exceeds its published specifications for a one-year calibration interval. After adjustment, repeat verification to ensure that the device meets the calibration test limits. Figure 5 shows the programming flow for a complete calibration.



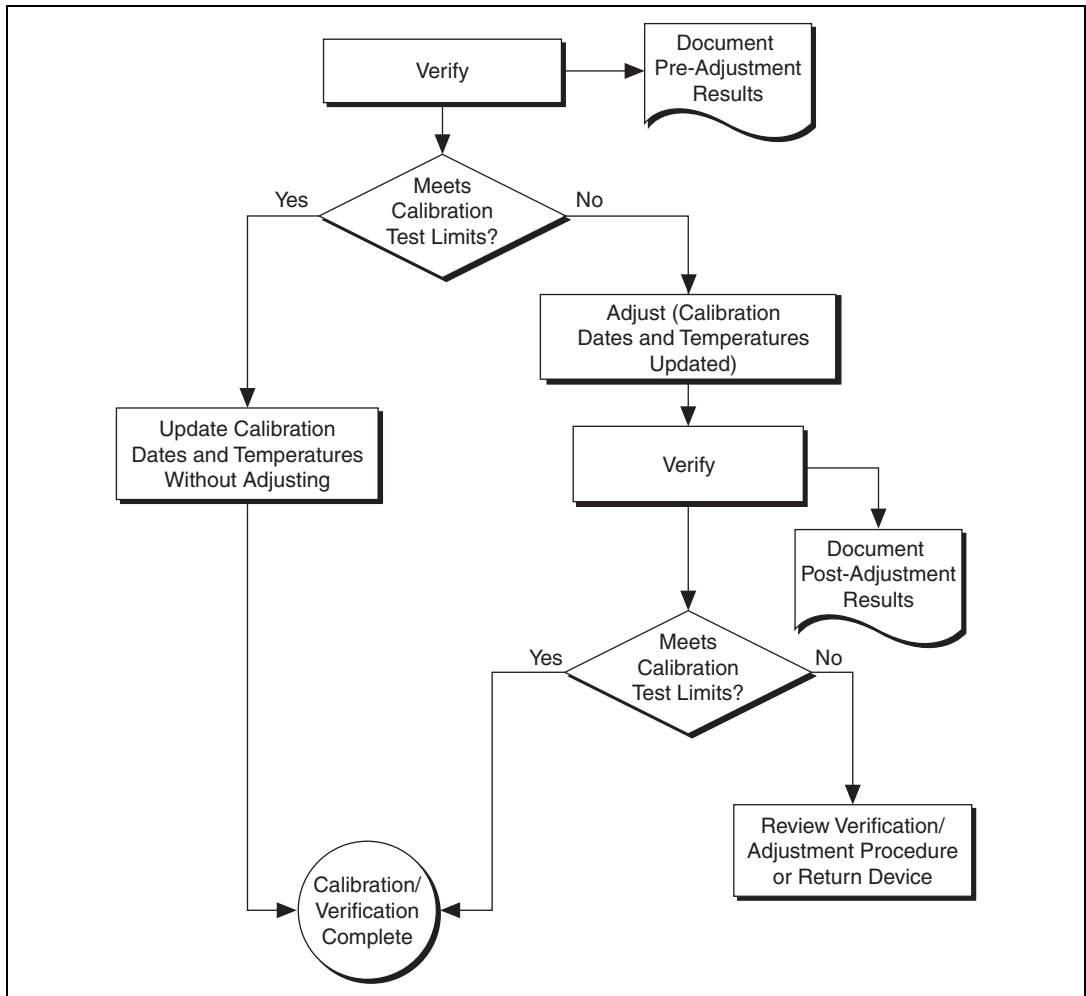
**Figure 5.** Complete Calibration Programming Flow

# Optional Calibration

If the accuracy of the device is within the calibration test limits during the initial verification, the NI PXI-4130 meets its published specifications, and you can skip the adjustment steps of the calibration procedure. If you choose to skip the adjustment, you can update the calibration date, effectively resetting the calibration interval. Refer to the [Adjustment](#) section for more information.

If you choose to perform an adjustment without verification, you must still verify that the accuracy of the device is within the calibration test limits post-adjustment.

Figure 6 shows the programming flow for the optional calibration.



**Figure 6.** Optional Calibration Programming Flow

# Appendix B: Calibration Utilities

---

NI-DCPower provides a full complement of calibration utility functions and VIs. You can use utility functions and VIs to retrieve information about adjustments performed on the NI PXI-4130, change the calibration password, and store small amounts of information in the onboard EEPROM. Refer to the *NI DC Power Supplies and SMUs Help* for the complete function reference and VI reference. The utility functions include:

- niDCPower Change Ext Cal Password VI  
(niDCPower\_ChangeExtCalPassword)
- niDCPower Get Ext Cal Recommended Interval VI  
(niDCPower\_GetExtCalRecommendedInterval)
- niDCPower Get Ext Cal Last Date And Time VI  
(niDCPower\_GetExtCalLastDateAndTime)
- niDCPower Get Cal User Defined Info Max Size VI  
(niDCPower\_GetCalUserDefinedInfoMaxSize)
- niDCPower Set Cal User Defined Info VI  
(niDCPower\_SetCalUserDefinedInfo)
- niDCPower Get Cal User Defined Info VI  
(niDCPower\_GetCalUserDefinedInfo)
- niDCPower Read Current Temperature VI  
(niDCPower\_ReadCurrentTemperature)
- niDCPower Get Ext Cal Last Temp VI  
(niDCPower\_GetExtCalLastTemp)

## Calibration Function References

The VIs and functions used in this procedure, including all calibration VIs and functions, are documented in the *NI-DCPower VI Reference Help* and the *NI-DCPower Function Reference Help*, both of which you can access from the *NI DC Power Supplies and SMUs Help* at **Start»All Programs»National Instruments»NI-DCPower»Documentation**.

# 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 electromagnetic 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 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851,  
Singapore 1800 226 5886, Slovenia 386 3 425 42 00,  
South Africa 27 0 11 805 8197, Spain 34 91 640 0085,  
Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151,  
Taiwan 886 02 2377 2222, Thailand 662 278 6777,  
Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

National Instruments, NI, 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, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or [ni.com/patents](http://ni.com/patents).