CALIBRATION PROCEDURE

NI PXI-4110

This document contains instructions for writing a manual calibration procedure for the NI PXI-4110 programmable DC power supply. 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`.

Contents

NATIONAL INSTRUMENTS

Conventions

»	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-4110, you must install NI-DCPower version 1.0 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`.

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 in parentheses. 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 1.0 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`	Calibration header file, which you must include in any C program accessing calibration functions. This file automatically includes `niDCPower.h`, which defines the rest of the NI-DCPower interface.

Documentation Requirements

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

- *NI PXI-4110 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`.

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-4110 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* section for more information.

Test Equipment

Table 2 lists the equipment required to calibrate the NI PXI-4110. 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-4110 Calibration

Required Equipment	Recommended Equipment	Specifications
Digital multimeter (DMM)	NI 4071	Voltage: ≤ ±50 ppm accuracy, ≤ 30 µV resolution; Current: ≤ ±0.04% accuracy, ≤ 50 nA resolution
External load	Clarostat 240C	Power resistor decade box with a range of 4 to 48,000 Ω and an accuracy of ±10%
Auxiliary power supply	NI APS-4100	11 V to 15.5 V, 5 A
Twisted pair, shielded cabling wire	—	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`.
- Keep relative humidity between 10% and 90%, noncondensing.
- Maintain an ambient temperature of 23 °C ±10 °C.
- Allow a warm-up time of at least 15 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-4110 is at a stable operating temperature.

Calibration Procedures

The calibration process includes the following steps:

1. *Initial Setup*—Install the device and configure it in Measurement & Automation Explorer (MAX).
2. *Verification*—Verify the existing operation of the device. This step 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 steps are described in more detail in the following sections.

Note The complete external calibration procedure consists of verifying the performance of the power supply, 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.

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-4110 Channel(s)	DMM*		
	Function	Range†	Input Impedance†
0	DC Volts	10 V	10 GΩ
1, 2		100 V	10 MΩ

* Use the highest resolution available on the DMM.

† 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 Output and Measurement Verification/Adjustment

NI PXI-4110 Channel(s)	DMM*		
	Function	Range	Input Impedance
0	DC Current	1 A	10 MΩ
1, 2 (20 mA)		0.1 A	
1, 2 (1 A)		1 A	

* Use the highest resolution available on the DMM.

Verification

This section describes the program you must write to verify the published specifications for the NI PXI-4110.

Verification consists of generating and measuring a series of outputs using the NI PXI-4110, 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-4110 meets its published specifications, and adjustment is optional. If the results fall outside of the test limits, you must adjust the NI PXI-4110.

Verification tests the following NI PXI-4110 specifications:

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

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

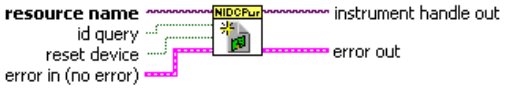
Verification of the NI PXI-4110 is complete only after you have successfully completed all tests in this section.

Note If verification fails post-adjustment, confirm that you have met the required *Test Conditions* before you return the NI PXI-4110 to NI for repair.

Verifying Voltage Programming Accuracy

Complete the following steps to verify the voltage programming accuracy of the NI PXI-4110. 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
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4110.

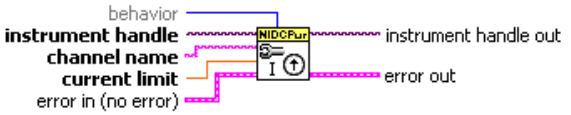
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 3.

4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

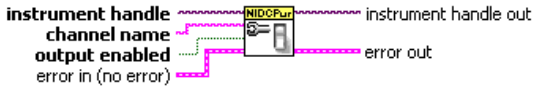
5. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 5

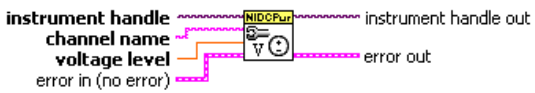
7. Enable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

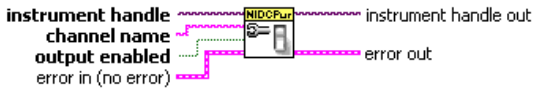
8. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

9. Wait 50 ms for the output of the NI PXI-4110 to settle.
10. Measure the output voltage with the DMM.
11. Record the measurement.
12. To calculate the output error, subtract the measurement you recorded in step 11 from the *Output* value for the iteration of channel *x*.
13. Compare the output error to the *Test Limit* for the iteration of channel *x* in Table 5. If the output error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
14. Repeat steps 4 through 13 for each iteration of channel *x* in Table 5. When you have verified all iterations of channel *x*, proceed to step 15.
15. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

17. Disconnect the DMM.
18. Repeat steps 2 through 17 for all unverified channels in Table 5. When you have verified all iterations per channel, this part of the verification is complete.
19. End the session using the niDCPower Close VI.

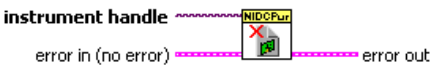
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

Table 5. NI PXI-4110 Output Parameters and Test Limits for Voltage Programming Accuracy Verification

Channel	Iteration	Output (V)	Test Limit (V)
0	1	0	±0.00400
	2	1.5	±0.00475
	3	3	±0.00550
	4	4.5	±0.00625
	5	6	±0.00700

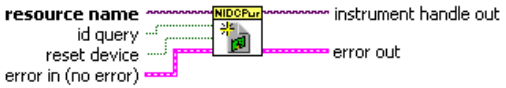
Table 5. NI PXI-4110 Output Parameters and Test Limits for Voltage Programming Accuracy Verification (Continued)

Channel	Iteration	Output (V)	Test Limit (V)
1	1	0	±0.01000
	2	5	±0.01250
	3	10	±0.01500
	4	15	±0.01750
	5	20	±0.02000
2	1	0	±0.01000
	2	–5	±0.01250
	3	–10	±0.01500
	4	–15	±0.01750
	5	–20	±0.02000

Verifying Voltage Measurement Accuracy

Complete the following steps to verify the voltage measurement accuracy of the NI PXI-4110. 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
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. Connect the DMM to the channel *x* output terminals of the NI PXI-4110.

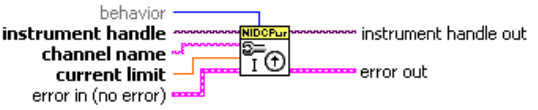
Note Channel *x* represents the channel under test. Replace the variable *x* in the program with the actual channel name.

3. Configure the DMM using the configuration settings for channel *x* in Table 3.

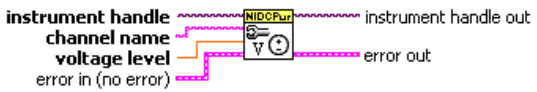
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

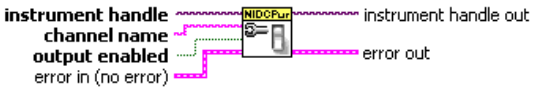
5. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 6

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

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

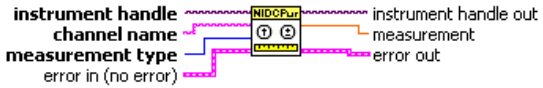
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **value**: `300`

9. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

10. Wait 50 ms for the output of the NI PXI-4110 to settle.
11. Measure the output voltage with the DMM.
12. Record the measurement.

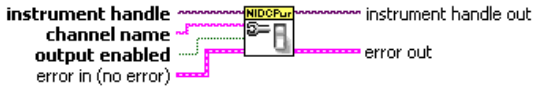
13. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_VOLTAGE`

14. Record the measurement.
15. To calculate the measurement error, subtract the measurement you recorded in step 12 from the measurement you recorded in step 14.
16. Verify that the measurement falls between the tolerances listed in the *Test Limit* for the iteration of channel *x* in Table 6. Tolerances are provided instead of absolute limits because the DMM measures a unique value. If the measurement error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
17. Repeat steps 4 through 16 for each iteration of channel *x* in Table 6. When you have verified all iterations of channel *x*, proceed to step 18.
18. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unverified channels in Table 6. When you have verified all iterations per channel, this part of the verification is complete.
22. End the session using the niDCPower Close VI.

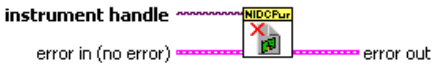
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

Table 6. NI PXI-4110 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification

Channel	Iteration	Output (V)	Test Limit (V)
0	1	0	±0.004000
	2	1.5	±0.004750
	3	3	±0.005500
	4	4.5	±0.006250
	5	6	±0.007000

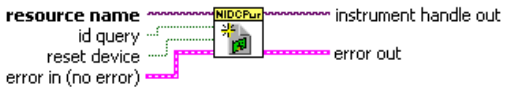
Table 6. NI PXI-4110 Output Parameters and Test Limits for Voltage Measurement Accuracy Verification (Continued)

Channel	Iteration	Output (V)	Test Limit (V)
1	1	0	±0.005000
	2	5	±0.007500
	3	10	±0.010000
	4	15	±0.012500
	5	20	±0.015000
2	1	0	±0.005000
	2	–5	±0.007500
	3	–10	±0.010000
	4	–15	±0.012500
	5	–20	±0.015000

Verifying Current Output Accuracy

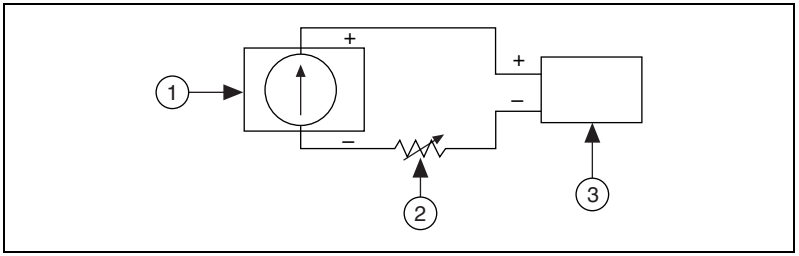
Complete the following steps to verify the current output accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range in Table 8.

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. Connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.

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



1 NI PXI-4110 2 External Load 3 DMM

Figure 1. Current Accuracy Verification Test System

3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 8.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

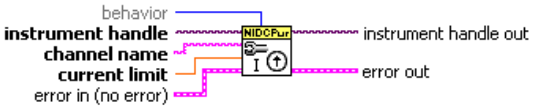
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for channel *x* in Table 7

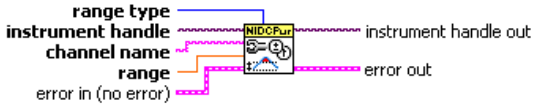
Table 7. Voltage Level Output Parameters

Channel	Output (V)
0	5
1	16
2	–16

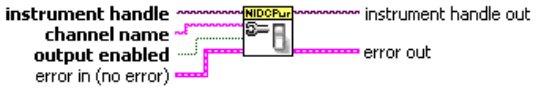
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: The *Output* value for the iteration of channel *x* in Table 8

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **rangeType**: `NIDCPOWER_VAL_RANGE_CURRENT` **range**: The *Current Range* value for the iteration of channel *x* in Table 8

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

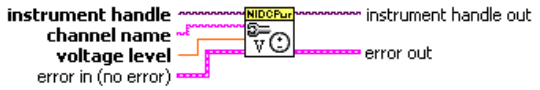
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

10. Apply the configuration using the niDCPower Initiate VI.

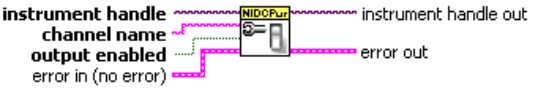
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

11. If the *Equivalent Resistance* value for this iteration of channel *x* is 48,000 Ω, wait 15 s for the output of the NI PXI-4110 to settle. For all other *Equivalent Resistance* values, wait 3 s for the output of the NI PXI-4110 to settle.
12. Measure the output current using the DMM.
13. Record the measurement.
14. To calculate the output error, subtract the measurement you recorded in step 13 from the *Output* value for the iteration of channel *x*.
15. Compare the output error to the *Test Limit* for the iteration of channel *x* in Table 8. If the output error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
16. Repeat steps 4 through 15 for all iterations of channel *x* in Table 8 per supported current range. When you have verified all iterations of channel *x*, proceed to step 17.

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

19. Disconnect the DMM and the external load.
20. Repeat steps 2 through 19 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.
21. End the session using the niDCPower Close VI.

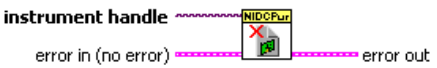
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

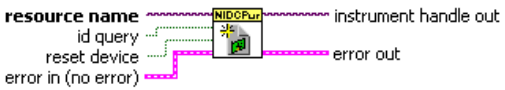
Table 8. NI PXI-4110 Output Parameters and Test Limits for Current Output Accuracy Verification

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (A)	Test Limit (A)
0	1 A	1	200	0.020000	±0.004030
		2	16	0.250000	±0.004375
		3	8	0.500000	±0.004750
		4	5	0.750000	±0.006063
		5	4	1.000000	±0.008000
1, 2	1 A	1	450	0.020000	±0.004030
		2	32	0.250000	±0.004375
		3	18	0.500000	±0.004750
		4	12	0.750000	±0.006063
		5	8	1.000000	±0.008000
1, 2	20 mA	1	24,000	0.000400	±0.000060
		2	1,900	0.005000	±0.000068
		3	900	0.010000	±0.000075
		4	600	0.015000	±0.000083
		5	480	0.020000	±0.000090

Verifying Current Measurement Accuracy

Complete the following steps to verify the current measurement accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range in Table 9.

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_init` with the following parameters: **resourceName**: The device name assigned by MAX **idQuery**: `VI_FALSE` **resetDevice**: `VI_TRUE`

2. If the *Output* value for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the external load to the channel *x* output terminals of the NI PXI-4110.

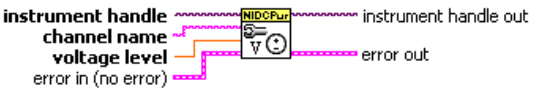
   For all other *Output* values, connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.

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

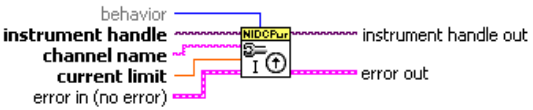
3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 9.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

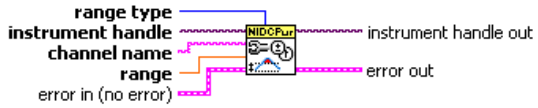
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 9

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: The *Current Range* value for the iteration of channel *x* in Table 9

- Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputRange` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **rangeType**: `NIDCPOWER_VAL_RANGE_CURRENT` **range**: The *Current Range* value for the iteration of channel *x* in Table 9

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_TRUE`

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

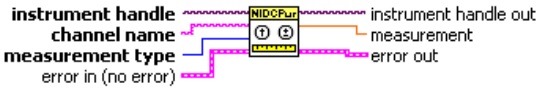
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **value**: `300`

10. Apply the configuration using the niDCPower Initiate VI.

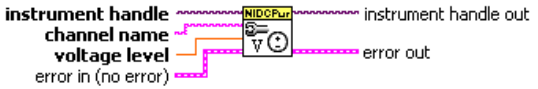
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

11. Wait 3 s for the output of the NI PXI-4110 to settle.
12. If the *Output* value for this iteration of channel *x* is 0, skip to step 13. For all other *Output* values, measure the output current using the DMM.
13. Record the measurement. If the *Output* value for this iteration of channel *x* is 0, you did not take a measurement; record "0" in place of the measurement.

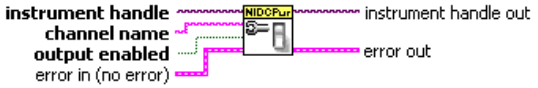
Note Measure the output voltage using the niDCPower Measure VI. To facilitate an accurate measurement using an *Output* value of 0, verify that the DMM and the external load are disconnected from the NI PXI-4110. For all other *Output* values, the DMM and the external load are connected to the NI PXI-4110.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_CURRENT`

14. Record the measurement.
15. To calculate the measurement error, subtract the measurement you recorded in step 13 from the measurement you recorded in step 14.
16. Verify that the measurement falls between the tolerances listed in the *Test Limit* for the iteration of channel *x* in Table 9. Tolerances are provided instead of absolute limits because the DMM measures a unique value. If the measurement error is outside of the test limit, adjustment of the NI PXI-4110 is necessary.
17. Repeat steps 4 through 16 for all iterations of channel *x* per supported current range in Table 9. When you have verified all iterations of channel *x*, proceed to step 18.
18. Configure the voltage level using the niDCPower Configure Voltage Level VI.

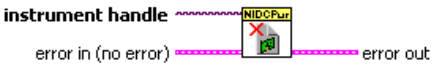
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_init` **channelName**: *x* **enabled**: `VI_FALSE`

20. Disconnect the DMM and the external load.
21. Repeat steps 1 through 20 for all unverified channels in Table 9. When you have verified all iterations per channel and range, this part of the verification is complete.

22. End the session using the niDCPower Close VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_close` with the following parameter: **vi**: The instrument handle from `niDCPower_init`

If the device has successfully passed all verification tests, the NI PXI-4110 is within the published specifications, and adjustment is optional.

Table 9. NI PXI-4110 Output Parameters and Test Limits for Current Measurement Accuracy Verification

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (V)*	Test Limit (A)	Published Specification (A)
0	1 A	1	open	0	±0.002000	±0.004
		2	5	1.25	±0.002050	±0.004375
		3	5	2.5	±0.002500	±0.00475
		4	5	3.75	±0.004000	±0.006063
		5	5	5	±0.005500	±0.008000
1, 2	1 A	1	open	0	±0.002000	±0.004
		2	20	±5	±0.002050	±0.004375
		3	20	±10	±0.002500	±0.00475
		4	20	±15	±0.004000	±0.006063
		5	20	±20	±0.005500	±0.008000
1, 2	20 mA	1	open	0	±0.000012	±0.000035
		2	1,000	±5	±0.000020	±0.000043
		3	1,000	±10	±0.000027	±0.000050
		4	1,000	±15	±0.000035	±0.000058
		5	1,000	±20	±0.000042	±0.000065

* When verifying channel 1, the *Output* is positive. When verifying channel 2, the *Output* is negative.

Adjustment

Adjustment improves the accuracy of the NI PXI-4110 and updates the calibration date and temperature in the EEPROM. Perform an adjustment once a year or when the accuracy of NI PXI-4110 is outside of the calibration test limits.

Adjustment corrects the following NI PXI-4110 specifications:

- Voltage output accuracy
- Voltage measurement accuracy
- Current output 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-4110 has passed initial verification and is within all calibration 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 (`niDCPower_InitExtCal`) and end with the niDCPower Close External Calibration VI (`niDCPower_CloseExtCal` with **action** set to `niDCPower_VAL_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-4110. 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
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

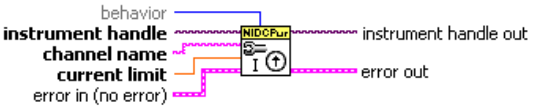
2. Connect the DMM to the channel *x* output terminals of the NI PXI-4110.

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

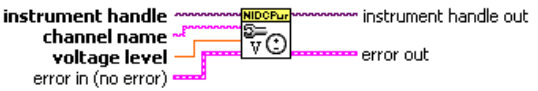
3. Configure the DMM using the configuration settings for channel *x* in Table 3.
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

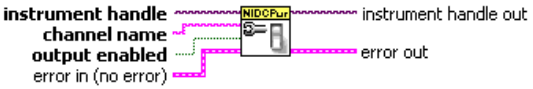
5. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 10

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

8. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

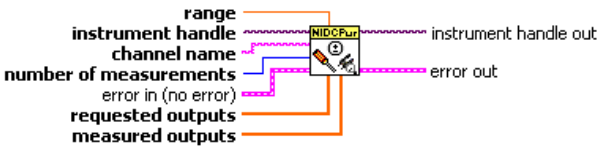
9. Wait 50 ms for the output of the NI PXI-4110 to settle.
10. Measure the output voltage with the DMM.
11. Record the measurement.
12. Repeat steps 4 through 11 for each iteration of channel *x* in Table 10. When you have measured all iterations of channel *x*, proceed to step 13.
13. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

15. Adjust the voltage level using the niDCPower Cal Adjust Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage range of channel *x* **numberOfMeasurements**: `3` **measuredOutputs**: An array composed of the measurements you recorded in step 11 for each iteration of channel *x* **requestedOutputs**: An array composed of the *Output* values for each iteration of channel *x* in Table 10

16. Disconnect the DMM.
17. Repeat steps 2 through 16 for all unadjusted channels in Table 10. When you have adjusted all channels, this part of the adjustment is complete.

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

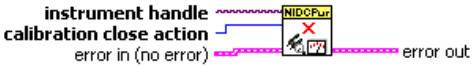
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 10. NI PXI-4110 Output Parameters for Voltage Programming Accuracy Adjustment

Channel	Iteration	Output (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	–10
	3	–20

Adjusting Voltage Measurement Accuracy

Complete the following steps to adjust the voltage measurement accuracy of the NI PXI-4110. Complete this test for each 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
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

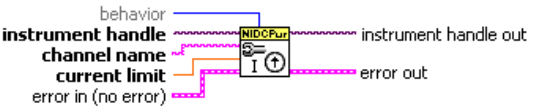
2. Connect the DMM to the channel *x* output terminals of the NI PXI-4110.

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

3. Configure the DMM using the configuration settings for channel *x* in Table 3.
4. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

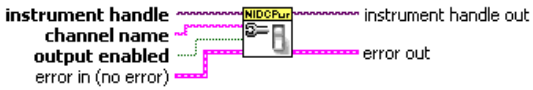
5. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: `0.5`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for the iteration of channel *x* in Table 11

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

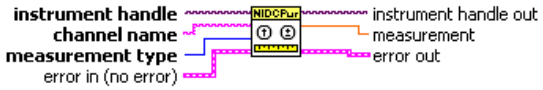
8. Specify the samples to average using the niDCPower property node.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **value**: `300`

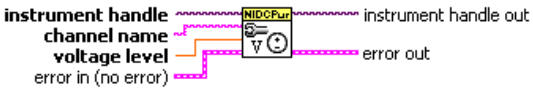
9. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

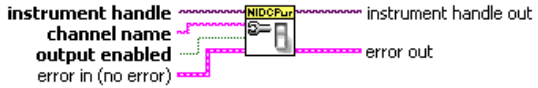
10. Wait 50 ms for the output of the NI PXI-4110 to settle.
11. Measure the output voltage with the DMM.
12. Record the measurement.
13. Measure the output voltage using the niDCPower Measure VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_VOLTAGE`

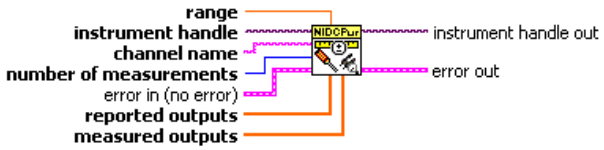
14. Record the measurement.
15. Repeat steps 4 through 14 for each iteration of channel *x* in Table 11. When you have measured all iterations of channel *x*, proceed to step 16.
16. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustVoltage Measurement` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage range of channel *x* **numberOfMeasurements**: `3` **measuredOutputs**: An array composed of the measurements you took with the DMM and recorded in step 12 for each iteration of channel *x* **requestedOutputs**: An array composed of the measurements you took with the NI PXI-4110 and recorded in step 14 for each iteration of channel *x*

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels in Table 11. When you have adjusted all channels, this part of the adjustment is complete.

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

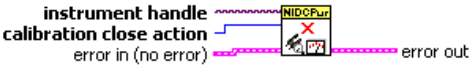
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 11. NI PXI-4110 Output Parameters for Voltage Measurement Accuracy Adjustment

Channel	Iteration	Output (V)
0	1	0
	2	3
	3	6
1	1	0
	2	10
	3	20
2	1	0
	2	–10
	3	–20

Adjusting Current Output Accuracy

Complete the following steps to adjust the current output accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range 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
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

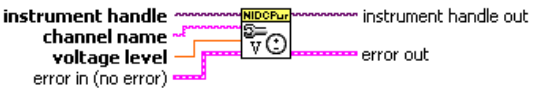
2. Connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated in Figure 1.

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

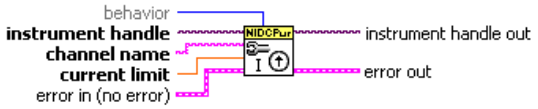
3. Configure the DMM using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 12.
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

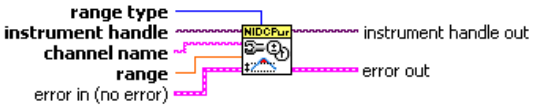
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for channel *x* in Table 7

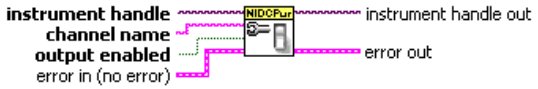
7. Configure the current limit using the niDCPower Configure Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: The *Output* value for the iteration of channel *x* in Table 12

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **rangeType**: `NIDCPOWER_VAL_RANGE_CURRENT` **range**: The *Current Range* value for the iteration of channel *x* in Table 12

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

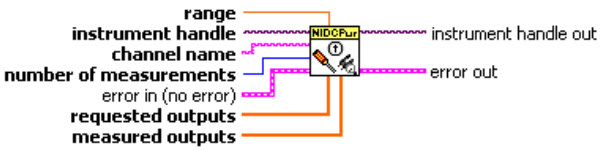
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

10. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

11. If the *Equivalent Resistance* value for this iteration of channel *x* is 48,000 Ω, wait 15 s for the output of the NI PXI-4110 to settle. For all other *Equivalent Resistance* values, wait 3 s for the output of the NI PXI-4110 to settle.
12. Measure the output current using the DMM.
13. Record the measurement.
14. Repeat steps 4 through 13 for each iteration of channel *x* of this current range. When you have measured all iterations of channel *x* in this current range, proceed to step 14.

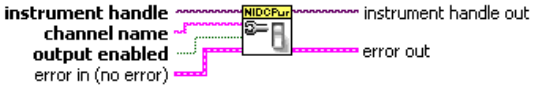
15. Adjust the current limit using the niDCPower Cal Adjust Current Limit VI.

LabVIEW VI	C/C++ Function Call
	Call `CalAdjustCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The current range of channel *x* **numberOfMeasurements**: `3` **measuredOutputs**: An array composed of the measurements you recorded in step 13 for each iteration of channel *x* **requestedOutputs**: An array composed of the *Output* values for each iteration of channel *x* in Table 12

16. Repeat steps 3 through 15 for all current ranges of channel *x* in Table 12. When you have adjusted all current ranges of channel *x*, proceed to step 17.
17. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: `0`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

19. Disconnect the DMM.
20. Repeat steps 2 through 19 for all unadjusted channels per current range in Table 12. When you have adjusted all channels per range, this part of the adjustment is complete.
21. End the session using the niDCPower Close External Calibration VI.

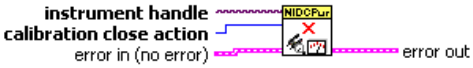
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

Table 12. NI PXI-4110 Output Parameters for Current Output Accuracy Adjustment

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (A)
0	1 A	1	80	0.050
		2	12	0.375
		3	5	0.750
1, 2	1 A	1	180	0.050
		2	24	0.375
		3	12	0.750

Table 12. NI PXI-4110 Output Parameters for Current Output Accuracy Adjustment (Continued)

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (A)
1, 2	20 mA	1	9,600	0.001
		2	900	0.010
		3	480	0.020

Adjusting Current Measurement Accuracy

Complete the following steps to adjust the current measurement accuracy of the NI PXI-4110. Complete this test for each channel iteration per supported range in Table 13.

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_InitExtCal` with the following parameters: **resourceName**: The device name assigned by MAX **password**: `NI`

2. If the *Output* value for this iteration of channel *x* is 0, skip to step 5. Do *not* connect the DMM or the external load to the channel *x* output terminals of the NI PXI-4110.

   For all other *Output* values, connect the DMM and the external load to the channel *x* output terminals of the NI PXI-4110, as illustrated 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 using the configuration settings for channel *x* in Table 4.
4. Configure the external load using the *Equivalent Resistance* value for the iteration of channel *x* in Table 13.

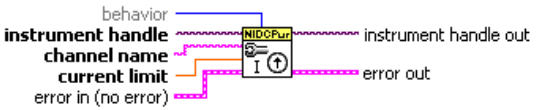
5. Place the NI PXI-4110 in delayed configuration mode using the niDCPower Abort VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Abort` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

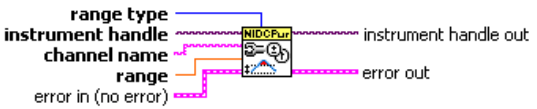
6. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: The *Output* value for channel *x* in Table 13

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureCurrentLimit` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **behavior**: `NIDCPOWER_VAL_CURRENT_REGULATE` **limit**: The *Current Range* value for the iteration of channel *x* in Table 13

8. Configure the output range using the niDCPower Configure Output Range VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputRange` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **rangeType**: `NIDCPOWER_VAL_RANGE_CURRENT` **range**: The *Current Range* value for the iteration of channel *x* in Table 13

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_TRUE`

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

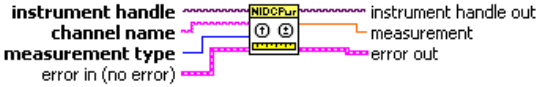
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_SetAttributeViInt32` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **value**: `300`

11. Apply the configuration using the niDCPower Initiate VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Initiate` with the following parameter: **vi**: The instrument handle from `niDCPower_InitExtCal`

12. Wait 3 s for the output of the NI PXI-4110 to settle.
13. If the *Output* value for this iteration of channel *x* is 0, skip to step 14. For all other *Output* values, measure the output current using the DMM.
14. Record the measurement. If the *Output* value for this iteration of channel *x* is 0, you did not take a measurement; record "0" in place of the measurement.

Note Measure the output voltage using the niDCPower Measure VI. To facilitate an accurate measurement using an *Output* value of 0, verify that the DMM and the external load are disconnected from the NI PXI-4110. For all other *Output* values, the DMM and the external load are connected to the NI PXI-4110.

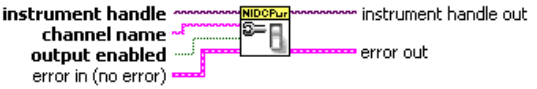
LabVIEW VI	C/C++ Function Call
	Call `niDCPower_Measure` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **measurementType**: `NIDCPOWER_VAL_MEASURE_CURRENT`

15. Record the measurement.
16. Repeat steps 4 through 15 for each iteration of channel *x* of this current range in Table 13. When you have measured all iterations of channel *x* in this current range, proceed to step 17.

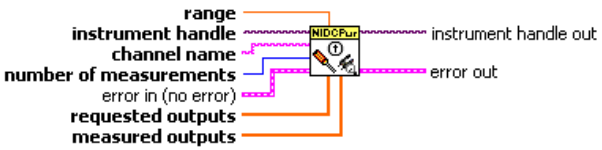
17. Configure the voltage level using the niDCPower Configure Voltage Level VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureVoltageLevel` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **level**: `0`

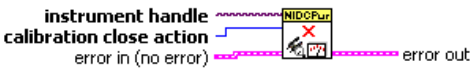
18. Disable the output using the niDCPower Configure Output Enabled VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_ConfigureOutputEnabled` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **enabled**: `VI_FALSE`

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

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CalAdjustCurrentMeasurement` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **channelName**: *x* **range**: The voltage range of channel *x* **numberOfMeasurements**: `3` **measuredOutputs**: An array composed of the measurements you took with the DMM and recorded in step 14 for each iteration of channel *x* **reportedOutputs**: An array composed of the measurements you took with the NI PXI-4110 and recorded in step 15 for each iteration of channel *x*

20. Disconnect the DMM.
21. Repeat steps 2 through 20 for all unadjusted channels in Table 13. When you have adjusted all channels, this part of the adjustment is complete.
22. End the session using the niDCPower Close External Calibration VI.

LabVIEW VI	C/C++ Function Call
	Call `niDCPower_CloseExtCal` with the following parameters: **vi**: The instrument handle from `niDCPower_InitExtCal` **action**: The instrument handle from `niDCPower_VAL_COMMIT`

When you have successfully completed all adjustment tests, adjustment of the NI PXI-4110 is complete. Repeat the *Verification* section to reverify the performance of the NI PXI-4110 post-adjustment. If the NI PXI-4110 successfully passes all verification tests, calibration is complete.

Table 13. NI PXI-4110 Output Parameters for Current Measurement Accuracy Adjustment

Channel(s)	Current Range	Iteration	Equivalent Resistance (Ω)	Output (V)*
0	1 A	1	open	0
		2	5	1.875
		3	5	3.75
1, 2	1 A	1	open	0
		2	20	±7.5
		3	20	±15
1, 2	20 mA	1	open	0
		2	1,000	±10
		3	1,000	±20

* When adjusting channel 1, the *Output* is positive. When adjusting channel 2, the *Output* is negative.

Appendix A: Calibration Options

Calibration involves verification, and, if necessary, adjustment, and reverification of the NI PXI-4110.

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-4110 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-4110 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 2 shows the programming flow for a complete calibration.

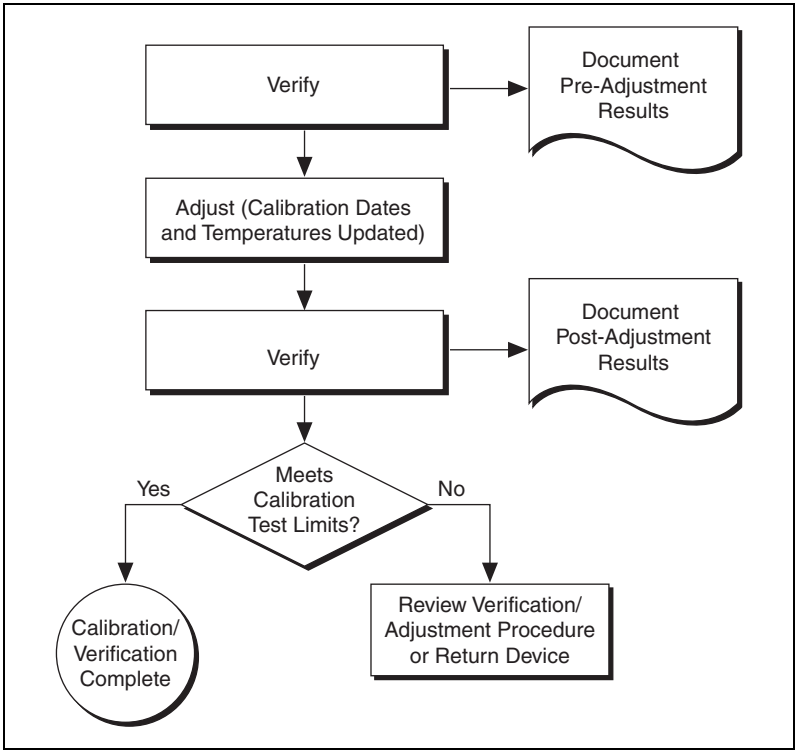


Figure 2. Complete Calibration Programming Flow

Optional Calibration

If the accuracy of the device is within the calibration test limits during the initial verification, the NI PXI-4110 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 3 shows the programming flow for the optional calibration.

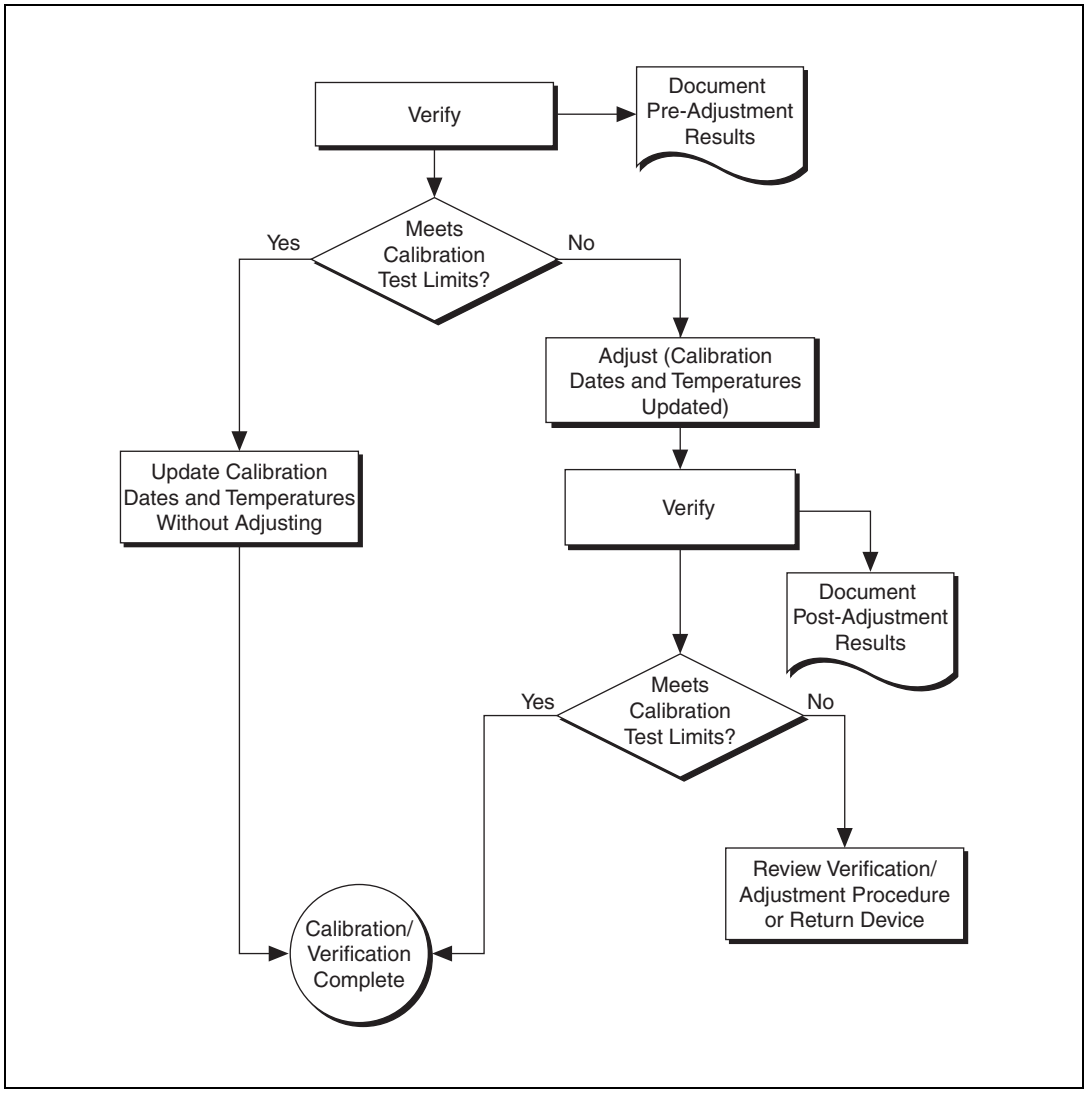


Figure 3. Optional Calibration Programming Flow

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-4110, 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 you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

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

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

Australia 1800 300 800, Austria 43 662 457990-0,
Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599,
Canada 800 433 3488, China 86 21 5050 9800,
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00,
Finland 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

371701C-01 Sep07