NI PXI-5600 RF Downconverter Calibration Procedure

Contents

August 2002
370544A-01

ni.com

NATIONAL INSTRUMENTS™

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.

♦

The ♦ symbol indicates that the following text applies only to a specific programming language.

✎

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

⚠

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

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. This font 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, and code excerpts.

`monospace bold`

Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

Introduction

This document contains step-by-step instructions for writing a calibration procedure for the NI PXI-5600 downconverter. The NI PXI-5600 downconverter module and the NI PXI-5620 digitizer module comprise the NI PXI-5660 RF Signal Analyzer. The digitizer and the downconverter modules must be calibrated individually.

What Is Calibration?

Calibration is a set of operations that compares the values indicated by a measuring instrument or measuring system to the corresponding values realized by external standards. The result of a calibration can be used to determine the measurement error and can correct for it in the adjustment process.

The calibration process consists of verifying, adjusting, and reverifying a device. During verification, you compare the measured performance to an external standard of known measurement uncertainty to confirm that the product meets or exceeds specifications. During adjustment, you correct the measurement error of the device by adjusting the calibration constants and storing the new calibration constants in the EEPROM. The host computer reads the calibration constants and the software uses them to compensate for errors in the data and to present calibrated data to the user.

Why Should You Calibrate?

The accuracy of electronic components drifts with time and temperature, which can affect measurement accuracy. Calibration restores the NI PXI-5600 to its specified accuracy and ensures that it still meets National Instruments standards.

How Often Should You Calibrate?

The accuracy requirements of your measurement application determine how often you should calibrate the NI PXI-5600. NI recommends performing a complete calibration at least once every year. You can shorten this interval based on the demands of your application.

Equipment and Other Test Requirements

This section describes the test instruments, test conditions, documentation, software, and connections required for calibration.

Test Instruments

Table 1 contains specifications for test instruments required for calibrating the NI PXI-5600. If you do not have the recommended test instruments, use the specifications listed in this table to select substitute calibration instruments.

Table 1. Required Test Instruments

Required Equipment	Recommended Make and Model	Necessary Specifications	
Spectrum Analyzer	Agilent 8563E	Frequency Range:	109 MHz to 111 MHz
		Noise Floor:	<130 dBm/Hz
		Frequency Resolution:	>0.5 Hz at 110 MHz
		Lockable to External Frequency Reference	
Signal Generator	Agilent ESG-4422B	Frequency Range:	20 MHz to 2.8 GHz
		Amplitude Range:	–30 dBm to +16 dBm
		Phase Noise/ Spectral Purity:	<95 dBc/Hz at 1 KHz
Power Meter with 2 Power Sensors	Anritsu ML2438A MA2421A MA2473A	Range of IF Output Power Sensor on CH-A:	–30 dBm to +20 dBm
		Range of Frequency of Sensor on CH-A:	3 MHz to 27 MHz
		Absolute Accuracy (Standard Uncertainty):	0.07 dB
		Range of RF Input Power Sensor on CH-B:	–40 dBm to +20 dBm
		Range of Frequency of Sensor on CH-B:	20 MHz to 2.8 GHz
		Absolute Accuracy[1] (Standard Uncertainty):	0.11 dB
Frequency Reference	Datum 8040 Rubidium Frequency Standard	Frequency:	10 MHz
		Frequency Accuracy:	1 ppb (typically ±6E-10 within 1 year)
		Amplitude Range:	>5 dBm (typ: 7 dBm)

Table 1. Required Test Instruments (Continued)

Required Equipment	Recommended Make and Model	Necessary Specifications	
Broadband Precision Power Splitter	Weinschel 1507R	Frequency Range:	DC to 3.0 GHz
		Output Maximum Voltage Standing-Wave Ratio (VSWR):	1.15
		Input Maximum VSWR:	1.25
		Amplitude Tracking:	<0.1 dB[2]
		Phase Tracking:	<4 Deg
		Insertion Loss:	Nominal 6 dB

[1] Including splitter and adapter uncertainty: 0.11 dB standard uncertainty when unit under test (UUT) on other side of the splitter has SWR of up to 2.5.

[2] Screened for <0.1 dB tracking.

Test Conditions

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

- Keep connections to the NI PXI-5600 short. Long cables and wires act as antennae, picking up noise that can affect measurements.
- Use 50 Ω coaxial cable for all connections to the NI PXI-5600.
- Keep relative humidity between 10 and 90%, noncondensing, or consult the hardware documentation for the optimum relative humidity.

Documentation

The following documents contain additional information you might find helpful when calibrating the NI PXI-5600:

- *NI-TUNER Reference Help* at **Start»Programs»National Instruments»RF Signal Analyzer»Documentation»Low Level**
- *PXI-5660 RF Signal Analyzer Specifications* at **Start»Programs» National Instruments»RF Signal Analyzer»Documentation**
- *PXI-5660 RF Signal Analyzer User Manual* at **Start»Programs» National Instruments»RF Signal Analyzer»Documentation**
- *Spectral Measurements Toolset User Guide* at **Start»Programs» National Instruments»RF Signal Analyzer»Documentation**

Software

The following software is required for calibrating the NI PXI-5600:

- NI RFSA 1.01—available on the RF Signal Analyzer software CD. This CD installs NI-SCOPE 2.12, NI-TUNER 1.01, and the Spectral Measurements Toolset.
- Calibration DLL—Calibration API and examples are available for download at `ni.com/support/calibrat/mancal`.

The calibration procedure in this document provides step-by-step instructions on calling the appropriate calibration functions, which are described in the *NI-TUNER Calibration API Function Reference* section. You can also program in LabVIEW using the example VIs that are included with the calibration DLL.

Connections

Figure 1 shows the appropriate connections for the NI PXI-5600 calibration procedure.

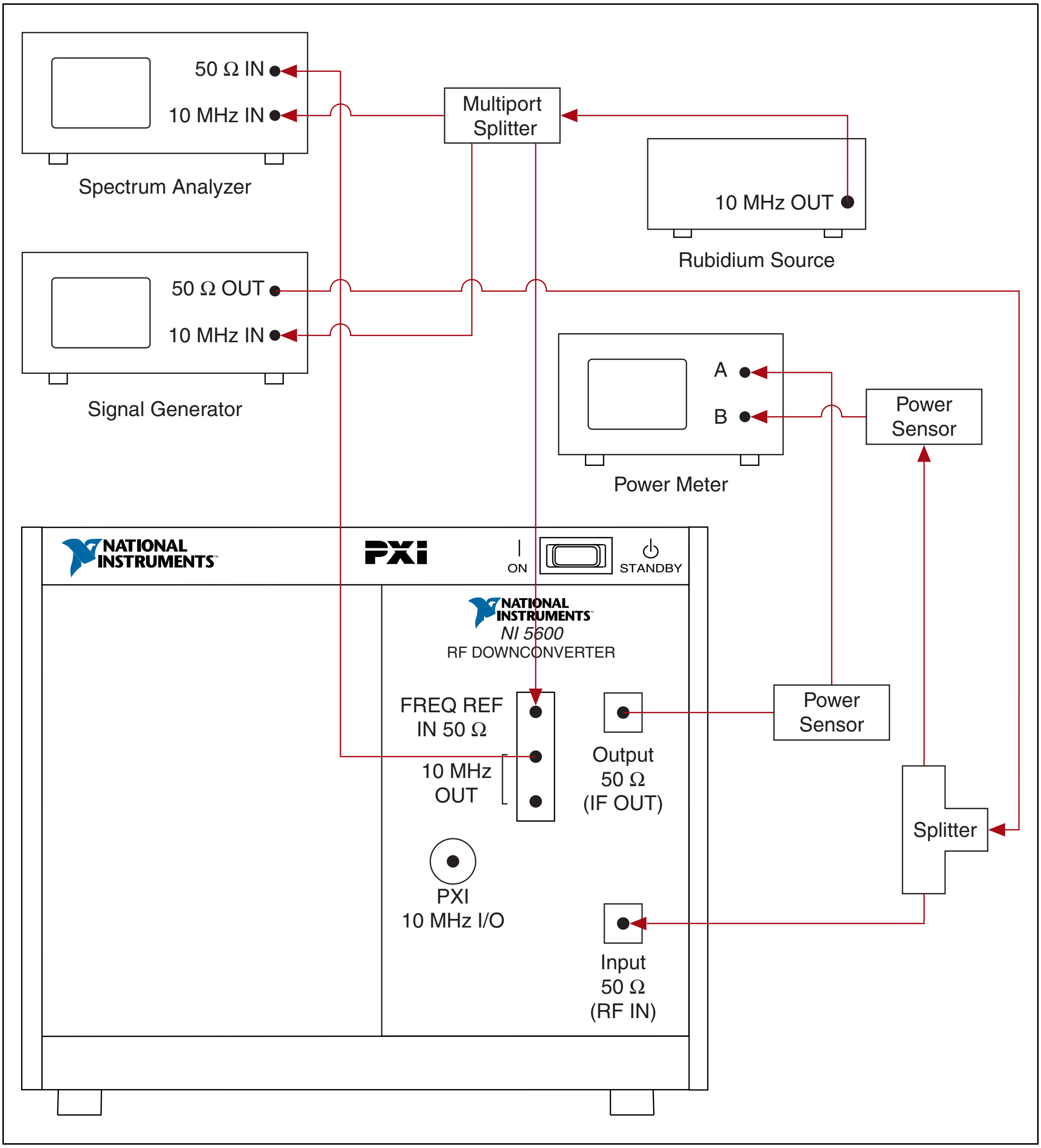

Figure 1. Connections

Calibration Flowchart

Figure 2 shows the main steps in the NI PXI-5600 calibration process.

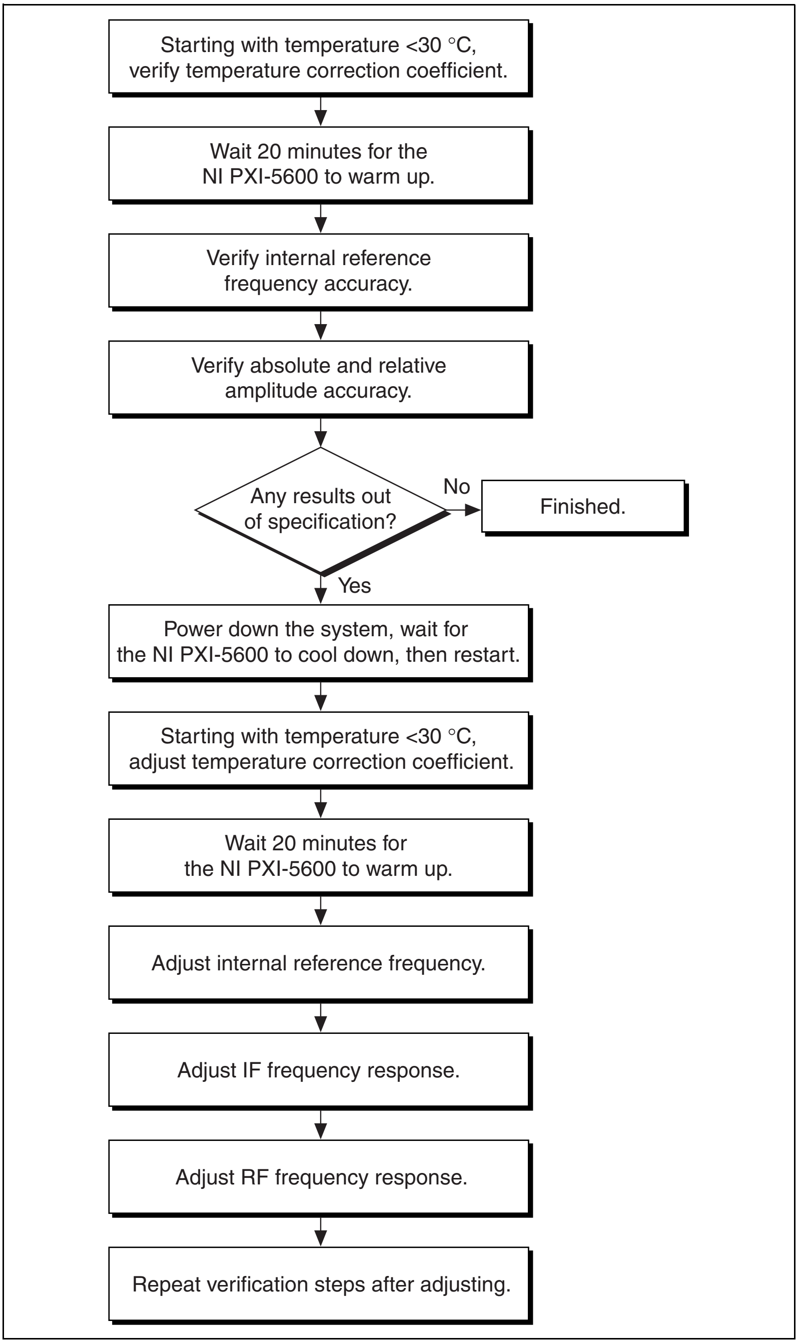
Starting with temperature <30 °C, verify temperature correction coefficient.
Wait 20 minutes for the NI PXI-5600 to warm up.
Verify internal reference frequency accuracy.
Verify absolute and relative amplitude accuracy.
Any results out of specification?
No
Finished.
Yes
Power down the system, wait for the NI PXI-5600 to cool down, then restart.
Starting with temperature <30 °C, adjust temperature correction coefficient.
Wait 20 minutes for the NI PXI-5600 to warm up.
Adjust internal reference frequency.
Adjust IF frequency response.
Adjust RF frequency response.
Repeat verification steps after adjusting.

Figure 2. Calibration Process Flowchart

Verification Procedures

Calibration includes both verification and adjustment procedures. To determine if the NI PXI-5600 requires adjustment, you must verify the temperature correction coefficient, the internal reference frequency, and the absolute and relative amplitude accuracy.

If the NI PXI-5600 fails any of the verification tests, complete the procedures described in the *Adjustment Procedures* section.

Note Refer to the *Connections* section for a diagram of how to set up the test instruments for these verification procedures.

Verifying the Temperature Correction Coefficient

Note Before you begin this verification procedure, you must ensure that the temperature of the NI PXI-5600 is below 30 °C and that the temperature does not reach 30 °C before step 7.

To verify the temperature correction coefficient, write a program that includes the following steps:

1. Call `niTuner_CalInitializeExternalCalibration` to open the calibration session.
2. Call `niTuner_CalGetCurrentTemperature` to determine the current temperature correction coefficient ($TempCo_{old}$).
3. Call `niTuner_CalSetAttenuation` with the following parameters:
   - **refLevel** = `0`
   - **mixLevel** = `-30`
4. Call `niTuner_CalSetClockSource` and set the clock source to external.
5. Wait 8 s for the NI PXI-5600 to lock to the external rubidium source.
6. Call `niTuner_CalGetLockState` to determine whether the NI PXI-5600 has locked to the external rubidium source. The lock state must be **`1`** (or **`TRUE`**).
7. Call `niTuner_CalGetCurrentTemperature` to read the temperature of the NI PXI-5600. Wait for the temperature to reach 30 °C.

Note If the temperature of the NI PXI-5600 exceeds 30 °C, abort the calibration session, power off the PXI chassis, and wait for the temperature to drop below 30 °C.

8. Set channel A on the power meter to read 15 MHz.
9. Set channel B on the power meter to read 100 MHz.
10. Set the amplitude of the signal generator to offset the loss of the power splitter so that you obtain 0 dB input to the NI PXI-5600.

Note If the power splitter you are using does not have a 6 dB loss, set the amplitude of the signal generator accordingly.

11. Wait 200 ms for channels A and B on the power meter to settle.
12. Loop 21 times. Include the following steps in each loop:
    a. Call `niTuner_CalGetStepFrequency` with the following parameters:
       - **startFrequency** = `98 MHz` (in terms of Hz)
       - **stopFrequency** = `102 MHz` (in terms of Hz)
       - **numberOf Steps** = `21`
       - **index** = current index of the loop (first iteration is `0`)

       The return value is the frequency of the current loop iteration.

    b. Set the signal generator frequency to the frequency value returned (in Hz) by `niTuner_CalGetStepFrequency` in step 12a.

    c. Wait 50 ms for the signal generator to settle.

    d. Acquire channels A and B on the power meter. Send them as inputs to `niTuner_CalAdjustExternalCalibration` with the following parameters:
       - **IFOutputPower** = channel A power (in dBm)
       - **RFInputPower** = channel B power (in dBm)
       - **calibrationProcedure** = `0` (temperature correction coefficient)
13. Repeat step 12 until the **done** value returned by `niTuner_CalAdjustExternalCalibration` on the last iteration is **`1`** (or **`TRUE`**).
14. Call `niTuner_CalGetCurrentTemperature` to determine the new temperature correction coefficient ($TempCo_{new}$).
15. Compare $TempCo_{new}$ with $TempCo_{old}$ to determine whether the NI PXI-5600 meets its specifications. Refer to Table 2 for the acceptable limits.
16. Call `niTuner_CalCloseExternalCalibration` to end the calibration session. Set **action** to `1` to cancel; do not save the result.

Table 2. Temperature Correction Coefficient Specifications

Verification Results	What to Do
$TempCo_{new} < -0.12$ OR $TempCo_{new} > -0.03$	Return the NI PXI-5600 to NI for repair.
$-0.12 \leq TempCo_{new} \leq -0.03$ AND $\{TempCo_{old} - 0.02 > TempCo_{new}$ OR $TempCo_{old} + 0.02 < TempCo_{new}\}$	Adjust the temperature correction coefficient (refer to the *Adjusting the Temperature Correction Coefficient* section for detailed instructions).
$-0.12 \leq TempCo_{new} \leq -0.03$ AND $TempCo_{old} - 0.02 \leq TempCo_{new} \leq TempCo_{old} + 0.02$	There is no need to adjust the temperature correction coefficient or to repair the NI PXI-5600.

Verifying Internal Reference Frequency Accuracy

Caution You must allow a warm-up time of at least 20 minutes before you begin the remaining verification procedures.

Note The NI PXI-5600 internal reference frequency is accurate to within 0.5 Hz of 10 MHz (50 ppb initial accuracy).

To verify the internal reference frequency accuracy, write a program that includes the following steps:

1. Call `niTuner_init` to open the session.
2. Call `niTuner_setAttenuation` and set the NI PXI-5600 with the following parameters:
   - **refLevel** = `0`
   - **mixLevel** = `-30`
3. Lock the spectrum analyzer to the 10 MHz OUT square wave output of the rubidium source.
4. Set the following parameters on the spectrum analyzer:
   - **center frequency** = `110 MHz` (examine the 11th harmonic of 10 MHz)
   - **span** = `600 Hz`

- **RBW** = `1 Hz`
- **input reference level** = `10 dBm`

5. Wait 1 s for the spectrum analyzer to settle.
6. Record the frequency output as measured by the spectrum analyzer. Verify that this value meets the specification listed in Table 3.

Table 3. Internal Reference Frequency Accuracy Specifications

Internal Reference Frequency Accuracy	
Accuracy After Adjustment (50 ppb)	**One Year Accuracy (150 ppb)**
109.9999945 – 110.0000055 MHz	109.9999835 MHz – 110.0000165 MHz

7. Call `niTuner_Close` to end the session.

Note The limits in Table 3 are for accuracy immediately after calibration, and for one year aging accuracy. Interpolate these limits appropriately for your situation.

Verifying Absolute and Relative Amplitude Accuracy

To verify absolute and relative amplitude accuracy, write a program that includes the following steps:

1. Call `niTuner_Init` to open the session.
2. Call `niTuner_ConfigReferenceClock` and set the reference clock to internal.
3. Set channel A on the power meter to read 15 MHz.

   Perform six verification iterations, as listed in Table 4, for each of the following seven frequencies: 100 MHz, followed by 25 MHz, 105 MHz, 805 MHz, 1.605 GHz, 2.115 GHz, and 2.695 GHz.

Note You can select any frequencies between 9 kHz and 2.7 GHz—you are not limited to the frequencies listed here. However, you must test 100 MHz, and you must test it before you test the remaining six frequencies to determine the relative amplitude accuracy.

For each verification iteration, complete steps 4 through 19:

4. For the first test frequency, set the signal generator amplitude as shown in Table 4 for the current iteration.
5. Call `niTuner_setAttenuation` and set the NI PXI-5600 to the **Reference Level** and **Mixer Level** shown in Table 4 for this iteration.

Table 4. Absolute and Relative Amplitude Accuracy Settings

Calibration Iteration	Signal Generator Amplitude	NI PXI-5600 Parameters	
		Reference Level	**Mixer Level**
1	–24 dBm	–30 dBm	–30 dBm
2	–14 dBm	–20 dBm	–30 dBm
3	–4 dBm	–10 dBm	–30 dBm
4	6 dBm	0 dBm	–30 dBm
5	16 dBm	10 dBm	–30 dBm
6	16 dBm	20 dBm	–30 dBm

6. Call `niTuner_setFreq` and set the NI PXI-5600 to the frequency you are testing.
7. Set the signal generator frequency to the frequency you are testing.
8. Set channel B on the power meter to the frequency you are testing.
9. Wait 200 ms for the signal generator and the power meter to settle.
10. Read channel A (*PWRA*) and channel B (*PWRB*) from the power meter.

Note *PWRA* corresponds to IF OUTPUT and *PWRB* corresponds to RF INPUT on the NI PXI-5600.

11. Call `niTuner_getTemperature` to obtain the temperature correction factor.

Caution Step 12 requires different actions depending on the programming language you are using. Steps 12a, 12b, and 12c apply to LabVIEW programming only. Steps 12d and 12e apply to C/C++ programming. Make sure to use the appropriate steps for your programming language.

12. Obtain the calibration factor to use in step 13:

♦ LabVIEW Programming

   a. Use the `niTuner GetCal` VI with the **correct for attenuation** parameter set to `TRUE`.

   b. Call `SmtCalculateCorrection` with the following inputs:

      - **actual spectrum settings**—contains the adjusted values for the following parameters:

        – **center frequency** = actual IF frequency returned by `niTuner_setFreq` in step 6

        – **number of spectral lines** = `1`

        – **span** = `-1.00`

        – **RBW** = `-1.00`

        – **window type** = `7 Term Blackman-Harris`

      - **units** = `dBm`

      - **calibration array** = the calibration factors returned by `niTuner_getCal` in step 12a

      The output of this function is an array of correction factors.

   c. Use the value at index 0 from the array of correction factors as the calibration factor in step 13.

♦ C/C++ Programming

   d. Call `niTunerVerify_GetCorrectionFactor` with the following inputs:

      - **taskId**—the task ID returned by `niTuner_init` in step 1

      - **actualRFTunedFrequency**—the actual adjusted RF center frequency returned by `niTuner_setFreq` in step 6

   e. Use the **correction factor** output of the function from the step 12d as the calibration factor in step 13.

Note You can find the function `niTunerVerify_GetCorrectionFactor` under `CVI\niTunerVerify.c` in the calibration files you downloaded with this document.

13. Add *PWRA* (from step 10), the temperature correction factor (from step 11), and the calibration factor (from step 12), to obtain $PWRA_{adjusted}$.

14. Calculate the difference between $PWRA_{adjusted}$ and *PWRB*.

Note If the frequency you are testing is 100 MHz, store the result from step 14 as *PWR100*.

15. Compare the result from step 14 to the specifications in Table 5 to determine absolute accuracy.

Table 5. Absolute Accuracy Specifications

Absolute Accuracy (dB) ≤2 GHz		**Absolute Accuracy (dB) >2 GHz**	
Specification	**Typical**	**Specification**	**Typical**
0.75	0.5	1.25	1

16. Calculate the difference between the result from step 14 and *PWR100*.
17. Compare the result from step 16 to the specifications in Table 6 to determine the relative accuracy.

Table 6. Relative Accuracy Specifications

Relative Accuracy (dB) ≤2 GHz		**Relative Accuracy (dB) >2 GHz**	
Specification	**Typical**	**Specification**	**Typical**
0.65	0.5	1.15	0.9

18. Repeat steps 4 through 18 for the remaining calibration iterations for this frequency, changing the signal generator amplitude and the **Reference Level** and **Mixer Level** settings as listed in Table 4.
19. Repeat steps 4 through 19 for each of the test frequencies.
20. Call `niTuner_Close` to end the session.

You have completed the verification procedures for the NI PXI-5600.

Adjustment Procedures

If the results you obtained in the *Verification Procedures* section indicate that the NI PXI-5600 does not meet its specifications, complete the adjustment procedures described in this section.

Note Refer to the *Connections* section for a diagram of how to set up the test instruments for these adjustment procedures.

Caution Always complete the adjustment procedures in the order specified in this document. Varying the order of the procedures can cause inaccurate results.

Adjusting the Temperature Correction Coefficient

Note Before you begin this adjustment procedure, you must ensure that the temperature of the NI PXI-5600 is below 30 °C and that the temperature does not reach 30 °C before step 6.

To adjust the temperature correction coefficient, write a program that includes the following steps:

1. Call `niTuner_CalInitializeExternalCalibration` to open the calibration session.
2. Call `niTuner_CalSetAttenuation` and set the following parameters:
   - **refLevel** = `0`
   - **mixLevel** = `-30`
3. Call `niTuner_CalSetClockSource` and set the clock source to external.
4. Wait 8 s for the NI PXI-5600 to lock to the external rubidium source.
5. Call `niTuner_CalGetLockState` to determine whether the NI PXI-5600 has locked to the external clock source. The lock state must be **`1`** (or **`TRUE`**).
6. Call `niTuner_CalGetCurrentTemperature` to read the temperature of the NI PXI-5600. Wait for the temperature to reach 30 °C.

Note If the temperature of the NI PXI-5600 exceeds 30 °C, abort the calibration session, turn off the PXI chassis, and wait for the temperature to drop below 30 °C.

7. Set channel A on the power meter to read 15 MHz.
8. Set channel B on the power meter to read 100 MHz.
9. Set the amplitude of the signal generator to offset the loss of the power splitter so that you obtain 0 dB input to the NI PXI-5600.

Note If the power splitter you are using does not have a 6 dB loss, set the amplitude of the signal generator accordingly.

10. Wait 200 ms for channels A and B on the power meter to settle.
11. Loop 21 times. Include the following steps in each loop:

    a. Call `niTuner_CalGetStepFrequency` with the following parameters:
       - **startFrequency** = `98 MHz` (in terms of Hz)
       - **stopFrequency** = `102 MHz` (in terms of Hz)

- **numberOf Steps** = `21`
- **index** = current index of the loop (first iteration is `0`)

The return value is the frequency of the current loop iteration.

b. Set the signal generator frequency to the frequency value returned (in Hz) by `niTuner_CalGetStepFrequency` in step 11a.

c. Wait 50 ms for the signal generator to settle.

d. Acquire channels A and B on the power meter. Send them as inputs to `niTuner_CalAdjustExternalCalibration` with the following parameters:

- **IFOutputPower** = channel A power (in dBm)
- **RFInputPower** = channel B power (in dBm)
- **calibrationProcedure** = `0` (temperature correction coefficient)

12. Repeat step 11 until the **done** value returned by `niTuner_CalAdjustExternalCalibration` on the last iteration is **`1`** (or **`TRUE`**).
13. Call `niTuner_CalCloseExternalCalibration` to end the calibration session. Set **action** to `0` to commit your changes, or to `1` if you want to cancel without saving.

Adjusting the Internal Reference Frequency

Caution You must allow a warm-up time of at least 20 minutes before you begin the remaining adjustment procedures.

To adjust the internal reference frequency accuracy on the NI PXI-5600, write a program that includes the following steps:

1. Call `niTuner_CalInitializeExternalCalibration` to open the calibration session.
2. Call `niTuner_CalSetAttenuation` with the following parameters:
   - **refLevel** = `0`
   - **mixLevel** = `–30`
3. Call `niTuner_CalSetClockSource` and set the clock source to internal.
4. Call `niTuner_CalAdjustInternalReferenceFrequency` with the following inputs:
   - **frequency** = `0`
   - **accuracy** = `0`
5. Lock the spectrum analyzer to the 10 MHz OUT square wave output of the rubidium source.

6. Set the following parameters on the spectrum analyzer:
   - **center frequency** = `110 MHz` (examine the 11th harmonic of 10 MHz)
   - **span** = `600 Hz`
   - **RBW** = `1 Hz`
   - **input level** = `10 dBm`
7. Set the spectrum analyzer to read the 10 MHz OUT of the NI PXI-5600.
8. Call `niTuner_CalAdjustInternalReferenceFrequency` with the following inputs:
   - **frequency** = the peak frequency read from the spectrum analyzer in step 7 (in terms of Hz)
   - **accuracy** = `0.55 Hz` (in terms of Hz)
9. Repeat steps 7 and 8 until the **done** value returned by `niTuner_CalAdjustInternalReferenceFrequency` is **`1`** (or **`TRUE`**).
10. Call `niTuner_CalCloseExternalCalibration` to end the calibration session. Set **action** to `0` to commit your changes, or to `1` if you want to cancel without saving.

Adjusting IF Frequency Response

To adjust the IF frequency response, write a program that includes the following steps:

1. Call `niTuner_CalInitializeExternalCalibration` to open the calibration session.
2. Call `niTuner_CalSetClockSource` and set the clock source to internal.
3. Call `niTuner_CalSetFrequency` and set the frequency of the NI PXI-5600 to 100 MHz.
4. Set the signal generator amplitude as shown in Table 7 for the first calibration iteration.
5. Call `niTuner_CalSetAttentuation` and set the RF signal attenuation to the specified **Reference Level** and **Mixer Level** listed in Table 7 for the current iteration.

Table 7. IF Adjustment Settings

Calibration Iteration	Signal Generator Amplitude	NI PXI-5600 Parameters	
		Reference Level	Mixer Level
1	–14 dBm	–20 dBm	–30 dBm
2	–4 dBm	–10 dBm	–20 dBm
3	6 dBm	0 dBm	–10 dBm
4	6 dBm	10 dBm	0 dBm

6. Loop 121 times. Include the following steps in each loop:

   a. Call `niTuner_CalGetStepFrequency` with the following inputs:

   - **startFrequency** = `88 MHz` (in terms of Hz)
   - **stopFrequency** = `112 MHz` (in terms of Hz)
   - **numberOf Steps** = `121`
   - **index** = current index of the loop (first iteration is `0`)

   The return value is the frequency of the current loop iteration.

   b. Set the signal generator frequency to the frequency (in Hz) returned by `niTuner_CalGetStepFrequency` in step 6a.

   c. Set channel A on the power meter to the frequency returned in step 6a minus 85 MHz.

   d. Set channel B on the power meter to the frequency returned in step 6a.

Note Refer to your power meter user documentation for instructions on setting the frequency.

   e. Wait 200 ms for the signal generator and the power meter to settle.

   f. Acquire channels A and B on the power meter. Send them as inputs to `niTuner_CalAdjustExternalCalibration` with the following values:

   - **IFOutputPower** = channel A power (in dBm)
   - **RFInputPower** = channel B power (in dBm)
   - **calibrationProcedure** = `1` (IF frequency response adjustment)

7. Repeat steps 4 through 6 for the remaining calibration iterations, changing the signal generator amplitude and the NI PXI-5600 attenuation parameters for each iteration as shown in Table 7.
8. Call `niTuner_CalCloseExternalCalibration` to end the calibration session. Set **action** to `0` to commit your changes, or to `1` if you want to cancel without saving.

Adjusting RF Frequency Response

To adjust the RF frequency response, write a program that includes the following steps:

1. Call `niTuner_CalInitializeExternalCalibration` to open the calibration session.
2. Call `niTuner_CalSetClockSource` and set the clock source to internal.
3. Set channel A on the power meter to read 15 MHz.
4. Set the signal generator amplitude as shown in Table 8 for the first calibration iteration.
5. Call `niTuner_CalSetAttenuation` and set the RF signal attenuation to the **Reference Level** and **Mixer Level** listed in Table 8 for the current iteration.

Table 8. RF Adjustment Settings

		NI PXI-5600 Parameters	
Calibration Iteration	**Signal Generator Amplitude**	**Reference Level**	**Mixer Level**
1	–24 dBm	–30 dBm	–30 dBm
2	–14 dBm	–20 dBm	–30 dBm
3	–4 dBm	–10 dBm	–30 dBm
4	6 dBm	0 dBm	–30 dBm
5	16 dBm	10 dBm	–30 dBm
6	16 dBm	20 dBm	–30 dBm

6. Loop 277 times. Include the following steps in each loop:
   a. Call `niTuner_CalGetStepFrequency` with the following inputs:
      - **startFrequency** = `20 MHz` (in terms of Hz)
      - **stopFrequency** = `2.78 GHz` (in terms of Hz)

   - **numberOf Steps** = `277`
   - **index** = current index of the loop (first iteration is `0`)

   The return value is the frequency of the current loop iteration.

   b. On the signal generator, set the frequency to the frequency value returned (in Hz) by `niTuner_CalGetStepFrequency` in step 6a.

   c. Call `niTuner_CalSetFrequency` and set the frequency of the NI PXI-5600 to the frequency value returned in step 6a.

   d. Set channel B on the power meter to the frequency value returned in step 6a.

   e. Wait 200 ms for the signal generator and the power meter to settle.

   f. Acquire channels A and B on the power meter. Send them as parameters to `niTuner_CalAdjustExternalCalibration` with the following inputs:

   - **IFOutputPower** = channel A power (in dBm)
   - **RFInputPower** = channel B power (in dBm)
   - **calibrationProcedure** = `2` (RF frequency response adjustment)

7. Repeat steps 4 through 6 for the remaining calibration iterations, changing the signal generator amplitude and the NI PXI-5600 parameters as shown in Table 8 for each iteration.
8. Call `niTuner_CalCloseExternalCalibration` to end the calibration session. Set **action** to `0` to commit your changes, or to `1` if you want to cancel without saving.

You have completed the adjustment procedures for the NI PXI-5600 calibration.

Reverification Procedures

After completing the adjustment procedures, repeat the procedures in the *Verification Procedures* section to ensure that the NI PXI-5600 is operating within its specifications after calibration.

NI-TUNER Calibration API Function Reference

This section contains descriptions of the specific calibration functions described in this document. For information in the NI-TUNER functions used in the *Verification Procedures* section, refer to the *NI-TUNER Reference Help*. The possible return values for the calibration functions are described in the *Calibration Error Codes* section.

niTuner_CalAdjustExternalCalibration

Function Prototype

```
long niTuner_CalAdjustExternalCalibration
(
   unsigned int calHandle,
   double IFOutputPower,
   double RFInputPower,
   unsigned int calibrationProcedure,
   unsigned long* done
)
```

Purpose

This function processes the power inputs to calculate calibration coefficients for the specified calibration procedure and stores them in a virtual EEPROM. The function `niTuner_CalCloseExternalCalibration` saves these calibration coefficients to the EEPROM when you set **action** = `0`. Use this function at the end of the iterations of the loops specified in the calibration procedure. You must call this function after `niTuner_CalGetStepFrequency`.

Parameters

Name	Description
calHandle	The calibration session reference
IFOutputPower	The value at the IF output connector on the NI PXI-5600, as measured by the power meter
RFInputPower	The value of the power at the output of the power splitter between the signal generator output and the input connector
calibrationProcedure	The calibration adjustment procedure calling this function: 0 = temperature correction coefficient 1 = IF frequency response 2 = RF frequency response
done	The status of the temperature correction coefficient procedure: 0 = not done 1 = done

niTuner_CalAdjustInternalReferenceFrequency

Function Prototype

```
long niTuner_CalAdjustInternalReferenceFrequency
(
   unsigned int calHandle,
   double frequency,
   double accuracy,
   unsigned long* done
)
```

Purpose

This function adjusts the internal reference frequency until the given frequency is within **accuracy** Hz of 110 MHz. Use this function in conjunction with the spectrum analyzer functions to calibrate the internal reference frequency. Refer to the *Adjusting the Internal Reference Frequency* section.

Parameters

Name	Description
calHandle	The calibration session reference
frequency	The frequency returned by the spectrum analyzer, representing the 11th harmonic of the internal reference frequency
accuracy	The deviation in Hz from 110 MHz that indicates when to stop adjusting the internal reference frequency
done	The status of the internal reference frequency adjustment procedure: 0 = not done 1 = done

niTuner_CalCloseExternalCalibration

Function Prototype

```
long niTuner_CalCloseExternalCalibration
(
   unsigned int calHandle,
   int action
)
```

Purpose

Closes the referenced NI-TUNER calibration session and optionally stores the results to the EEPROM of the NI PXI-5600. Call this function at the end of a calibration session to store calibration results.

Parameters

Name	Description
calHandle	The calibration session reference
action	The closing action to perform: 0 = commit; store the calibration results to the EEPROM 1 = cancel; close without storing calibration results. Use this value if an error occurs during calibration.

niTuner_CalGetCalibrationCount

Function Prototype

```
long niTuner_CalGetCalibrationCount
(
   int calType,
   char resourceName[],
   unsigned int* count
)
```

Purpose

This function returns the number of calibration procedures completed on the device. The device should have at least one calibration procedure completed during manufacturing.

Parameters

Name	Description
resourceName[]	The device number of the NI PXI-5600 obtained from Measurement & Automation Explorer (MAX) expressed as DAQ::<device number>
calType	The calibration type to read: 0 = Internal—the results of an internal, or self-calibration. Self-calibration is not supported by NI PXI-5600. 1 = External—the results of an external calibration. Always set this value to 1 for the NI PXI-5600.
count	The number of times the device has been calibrated

niTuner_CalGetCalibrationDateAndTime

Function Prototype

```
long niTuner_CalGetCalibrationDateAndTime
(
   char resourceName[],
   int calType,
   unsigned int* month,
   unsigned int* day,
   unsigned int* year,
   unsigned int* hour,
   unsigned int* minute
)
```

Purpose

This function returns the date and time of the last calibration performed on the NI PXI-5600. Pass properly allocated variables to the time information parameters of interest and some form of NULL or 0 otherwise.

Parameters

Name	Description
resourceName[]	The device number of the NI PXI-5600 obtained from MAX expressed as DAQ::<device number>
calType	The calibration type to read: 0 = Internal—the results of an internal, or self-calibration. Self-calibration is not supported by NI PXI-5600. 1 = External—the results of an external calibration. Always set this value to 1 for the NI PXI-5600.
month	The month of the last calibration
day	The day of the last calibration
year	The year of the last calibration
hour	The hour of the last calibration
minute	The minute of the last calibration

niTuner_CalGetCurrentTemperature

Function Prototype

```
long niTuner_CalGetCurrentTemperature
(
   unsigned int calHandle,
   double* temperature,
   double* correctionFactor
   double* temperatureCoefficient
)
```

Purpose

This function returns the temperature of the NI PXI-5600 downconverter in °C, the correction factor in dB, and the temperature coefficient in dB/°C. Retrieving the downconverter temperature causes a momentary disruption in the IF output signal, which may cause invalid IF data.

Parameters

Name	Description
calHandle	The calibration session reference
temperature	The current temperature of the NI PXI-5600 in °C
correctionFactor	The correction factor in dB based on the current temperature (normalized to 40 °C and multiplied by the calibrated temperature coefficient dB/°C)
temperatureCoefficient	The temperature coefficient in dB/°C used to calculate the **correctionFactor** based upon the current **temperature**

niTuner_CalGetLockState

Function Prototype

```
long niTuner_CalGetLockState
(
   unsigned int calHandle,
   unsigned long* lockState
)
```

Purpose

Returns the current PLL lock state of the NI PXI-5600. The locked state is equivalent to the STATUS light, indicating that the NI PXI-5600 is locked and settled to a frequency. This function serves as a good verification after a frequency tuning (as in `niTuner_CalSetFrequency`).

Parameters

Name	Description
calHandle	The calibration session reference
lockState	**`1`** (or **`TRUE`**) if the NI PXI-5600 is locked and settled to a frequency

niTuner_CalGetRecommendedExternalCalibrationInterval

Function Prototype

```
long niTuner_CalGetRecommendedExternalCalibrationInterval
(
   char resourceName[],
   unsigned int* months
)
```

Purpose

This function returns the recommended user calibration interval for the device in months. NI recommends that you calibrate the device after exceeding this interval.

Parameters

Name	Description
resourceName[]	The device number of the NI PXI-5600 obtained from MAX expressed as DAQ::<device number>
months	The recommended user calibration interval for the device

niTuner_CalGetStepFrequency

Function Prototype

```
long niTuner_CalGetStepFrequency
(
   unsigned int calHandle,
   double startFrequency,
   double stopFrequency,
   unsigned int numberOfSteps,
   unsigned int index,
   double* stepFrequency
)
```

Purpose

This function starts a procedural loop and returns the frequency of the current loop iteration. Use this function to adjust the temperature coefficient, IF frequency response, and RF frequency response. Set the proper values for the inputs **startFrequency**, **stopFrequency**, and **numberOfSteps** as specified in the calibration procedure.

Parameters

Name	Description
calHandle	The calibration session reference
startFrequency	The start frequency (in Hz) of the loop
stopFrequency	The stop frequency (in Hz) of the loop
numberOfSteps	The number of frequency steps in the loop, including the start and stop frequencies
index	The current value of the loop; an index of 0 refers to the first iteration of the loop
frequency	The frequency of the current loop iteration

niTuner_CalInitializeExternalCalibration

Function Prototype

```
long niTuner_CalInitExternalCalibration
(
   char   resourceName[],
   char*   password,
   unsigned int*   calHandle
)
```

Purpose

Initializes the PXI-5600 and opens a new calibration session. Call this function before any calibration adjustment to initialize the NI PXI-5600 in the same fashion as `niTuner_init`. This function returns a calHandle that serves as the session handle for subsequent NI-TUNER calibration functions.

Parameters

Name	Description
resourceName[]	The device number of the NI PXI-5600 obtained from MAX expressed as DAQ::<device number>
password	The user-defined password
calHandle	The calibration session reference

niTuner_CalSetAttenuation

Function Prototype

```
long niTuner_CalSetAttenuation
(
   unsigned int calHandle,
   int refLevel,
   int mixLevel
)
```

Purpose

This function sets the RF signal attenuation to the specified reference and mixer levels. **refLevel** must be ≤50 dBm. **mixLevel** must be ≤0 and ≥(**refLevel** – 50 dBm). Set the levels in accordance with the following formula:

(*reference level* – 50) ≤ *mixer level* ≤ Min (*reference level*, 0 dBm) ≤ *reference level*

Parameters

Name	Description
calHandle	The calibration session reference
refLevel	The desired reference level for the RF input signal
mixLevel	The desired mixer level for the RF input signal

niTuner_CalSetClockSource

Function Prototype

```
long niTuner_CalSetClockSource
(
   unsigned int calHandle,
   int source
)
```

Purpose

This function configures the reference clock source to accept an external signal or to use the internal OCXO. This function is necessary in the calibration adjustment for the reference frequency; it must be set to the internal clock source so that the signal to be calibrated is output through the 10 MHz OUT connector.

Parameters

Name	Description
calHandle	The calibration session reference
source	The clock source 0 = internal; use the internal clock source 1 = external; use an external clock source

niTuner_CalSetFrequency

Function Prototype

```
long niTuner_CalSetFrequency
(
   unsigned int calHandle,
   double frequency
)
```

Purpose

This function sets a single frequency in the scan list and triggers the NI PXI-5600 downconverter to settle on that frequency. If new center frequencies of less than 15 MHz are specified after a call to this function, attenuation is automatically adjusted. This may add an additional 20 dB attenuation.

Parameters

Name	Description
calHandle	The calibration session reference
frequency	The desired frequency between 9 kHz and 2.78 GHz to which the NI PXI-5600 tunes

niTuner_CalSetPassword

Function Prototype

```
long niTuner_CalSetPassword
(
   char resourceName[],
   char* oldPassword,
   char* newPassword
)
```

Purpose

This function sets the user calibration password. Use this function to customize the access password for initiating the calibration session. The default NI PXI-5600 user password is BLUE.

Parameters

Name	Description
resourceName[]	The device number of the NI PXI-5600 obtained from MAX expressed as DAQ::<device number>
oldPassword	The current user calibration password
newPassword	The new user calibration password

Calibration Error Codes

Table 9 list the calibration error codes that can be returned by the calibration functions described in the *NI-TUNER Calibration API Function Reference* section.

Table 9. Calibration Error Codes

Error Code	Description
–90001	The specified calibration session is already open.
–90002	The supplied password is too long; passwords are four standard characters (32 bits).
–90003	The supplied password is incorrect.
–90004	No password was supplied; a valid password is required for this operation.
–90005	NI-DAQ cannot be loaded; the Windows system directory was not found.
–90006	NI-TUNER function not found; you must have NI-DAQ 6.9.2 or later installed.
–90007	NI-DAQ was not found in the Windows system directory and could not be loaded.
–90008	This function is not supported.
–90009	This function has not been implemented.
–90010	The calibration session handle is closed.
–90011	The device number is out of range (1–64).
–90012	The specified calHandle is out of range (1–64).
–90013	The clock source is invalid.
–90014	The IF option is invalid.
–90015	The RF option is invalid.
–90016	The calibration procedure is invalid.
27*xxx*,–27*xxx*	Errors beginning with 27 or –27 are NI-TUNER driver errors. Refer to the *NI-TUNER Reference Help* for more information on these errors.

Technical Support Resources

NI Web Support

National Instruments Web support is your first stop for help in solving installation, configuration, and application problems and questions. Online problem-solving and diagnostic resources include frequently asked questions, knowledge bases, product-specific troubleshooting wizards, manuals, drivers, software updates, and more. Web support is available through the Technical Support section of `ni.com`.

Worldwide Support

National Instruments has offices located around the world to help address your support needs. You can access our branch office Web sites from the Worldwide Offices section of `ni.com`. Branch office Web sites provide up-to-date contact information, support phone numbers, email addresses, and current events.

If you have searched the technical support resources on our Web site and still cannot find the answers you need, contact your local office or National Instruments corporate. For telephone support in the United States, dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Brazil 55 11 3262 3599, Canada (Calgary) 403 274 9391,
Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085,
China 86 21 6555 7838, Czech Republic 02 2423 5774,
Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 01 42 96 427, Hong Kong 2645 3186,
India 91 80 4190000, Israel 03 6393737, Italy 02 413091,
Japan 03 5472 2970, Korea 02 3451 3400, Malaysia 603 9596711,
Mexico 001 800 010 0793, Netherlands 0348 433466,
New Zealand 09 914 0488, Norway 32 27 73 00, Poland 22 3390 150,
Portugal 210 311 210, Russia 095 238 7139, Singapore 65 6 226 5886,
Slovenia 3 425 4200, South Africa 11 805 8197, Spain 91 640 0085,
Sweden 08 587 895 00, Switzerland 056 200 51 51,
Taiwan 02 2528 7227, United Kingdom 01635 523545