nirfmxinstr

Release 26.3

National Instruments

May 13, 2026

CONTENTS

CHAPTER

ONE

ABOUT

The **nirfmx-python** repository generates Python bindings (Application Programming Interface) for interacting with the NI-RFmx drivers.

nirfmx-python follows Python Software Foundation support policy for different versions.

CHAPTER

TWO

OPERATING SYSTEM SUPPORT

nirfmxinstr supports Windows systems where the supported drivers are installed. Refer to NI Hardware and Operating System Compatibility for which versions of the driver support your hardware on a given operating system.

CHAPTER

THREE

INSTALLATION

You can use pip to download nirfmxinstr and install it.

```
$ python -m pip install nirfmxinstr
```

CHAPTER

FOUR

SUPPORT AND FEEDBACK

For support with Python API, hardware, the driver runtime or any other questions, please visit NI Community Forums.

CHAPTER

FIVE

DOCUMENTATION:

5.1 Attributes

attributes.py - Contains the ID of all attributes belongs to the module.

class nirfmxinstr.attributes.**AttributeID**(*value*)

Bases: Enum

This enum class contains the ID of all attributes belongs to the module.

ADVANCE_TRIGGER_DIGITAL_EDGE_SOURCE = 104

Specifies the source terminal for the advance trigger.

This attribute is used only when the *ADVANCE_TRIGGER_TYPE* attribute is set to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

ADVANCE_TRIGGER_EXPORT_OUTPUT_TERMINAL = 105

Specifies the destination terminal for the exported advance trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

ADVANCE_TRIGGER_TERMINAL_NAME = 106

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/AdvanceTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/AdvanceTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/AdvanceTrigger,*, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/AdvanceTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

ADVANCE_TRIGGER_TYPE = 103

Specifies whether the advance trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No advance trigger is configured.
Digital Edge (1)	The advance trigger is not asserted until a digital edge is detected. The source of the digital edge is specified with the Advance Trigger Digital Edge Source attribute.
Software (3)	The advance trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

AMPLITUDE_SETTLING = 56

Specifies the amplitude settling accuracy value. This value is expressed in decibels. RFmx waits until the RF power attains the specified accuracy level after calling the RFmx Initiate method.

Any specified amplitude settling value that is above the acceptable minimum value is coerced down to the closest valid value.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported Devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

CHANNEL_COUPLING = 11

Specifies whether the RF IN connector is AC- or DC-coupled on the downconverter.

Nam (valu	Description
—	NoteFor the PXIe-5665/5668, this attribute must be set to AC Coupled when the DC block is present, and set to DC Coupled when the DC block is not present to ensure device specifications are met and proper calibration data is used. For more information about removing or attaching the DC block, refer to the PXIe-5665 Theory of Operation or the PXIe-5668 Theory of Operation topics in the NI RF Vector Signal Analyzers Help.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **AC Coupled**.

Valid values

- PXIe-5665 (3.6 GHz): AC Coupled DC Coupled
- PXIe-5665 (14 GHz): AC Coupled, DC Coupled
- PXIe-5668: AC Coupled

Supported devices: PXIe-5665/5668

Name (Value)	Description
AC Coupled (0)	Specifies that the RF input channel is AC-coupled. For low frequencies (<10 MHz), accuracy decreases because RFmxInstr does not calibrate the configuration.
DC Coupled (1)	Specifies that the RF input channel is DC-coupled. The RFmx driver enforces a minimum RF attenuation for device protection.

CLEANER_SPECTRUM = 37

Specifies how to obtain the lowest noise floor or faster measurement speed.

Name (value)	Description
PXIe-5665	Sets the FFT Width attribute to take narrower bandwidth acquisitions and avoid digitizer spurs. Uses IF filters to reduce the noise floor for frequencies below 80 MHz.
PXIe-5644/5645/5646, PXIe-5840/5841/5842	Returns the best possible spectrum.
PXIe-5668	Returns the best possible spectrum. To provide the best spectrum measurement, the acquisition is reduced to 100 MHz segments for any center frequency.
Other devices	This attribute is ignored.

> **Note**
>
> Some measurements, such as Spurious Emissions enable the Cleaner Spectrum attribute by default. You can speed up those measurements by disabling the Cleaner Spectrum attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disable this attribute to get faster measurement speed.
Enabled (1)	Enable this attribute to get the lowest noise floor and avoid digitizer spurs.

COMMON_MODE_LEVEL = 70

Specifies the common-mode level presented at each differential input terminal. The common-mode level shifts both positive and negative terminals in the same direction. This must match the common-mode level of the device under test (DUT). This value is expressed in Volts.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5820

DEVICE_TEMPERATURE = 24

Returns the current temperature of the module. This value is expressed in degrees Celsius.

To use this attribute for PXIe-5830/5831/5832, you must first use the Selector String attribute to specify the name of the channel you are configuring. When you are reading the device temperature, you must specify the context in the Selector String input as "module::<ModuleName>". You can also use the `build_module_string()` method to build the module string. For all other devices, the only valid value for the selector string is "" (empty string).

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the selector string to read this attribute. You can use the `build_port_string()` method to build the selector string. For PXIe-5830/5831/5832, you must specify the context in the selector string input as port::<deviceName>/<channelNumber>/module::<moduleName>.

Refer to the following table to determine which strings are valid for your configuration.

Hardware Module	TRX Port Type	Selector String
PXIe-3621/3622/3623	•	if or “” (empty string)
PXIe-5820	•	fpga
First connected mmRH-5582	DIRECT TRX PORTS Only	rf0
First connected mmRH-5582	SWITCHED TRX PORTS [0-7]	rf0switch0
First connected mmRH-5582	SWITCHED TRX PORTS [8-15]	rf0switch1
Second connected mmRH-5582	DIRECT TRX PORTS Only	rf1
Second connected mmRH-5582	SWITCHED TRX PORTS [0-7]	rf1switch0
Second connected mmRH-5582	SWITCHED TRX PORTS [8-15]	rf1switch1

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

`DIGITAL_GAIN = 84`

Specifies the scaling factor applied to the time-domain voltage data in the digitizer. This value is expressed in dB. RFmx does not compensate for the specified digital gain.

You can use this attribute to account for external gain changes without changing the analog signal path.

> **Note**
>
> The PXIe-5644/5645/5646 applies this gain when the data is scaled. The raw data does not include this scaling on these devices.

Default Value : 0 dB

Supported Devices : PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5860

`DIGITIZER_DITHER_ENABLED = 21`

Specifies whether dithering is enabled on the digitizer.

Dithering adds band-limited noise in the analog signal path to help reduce the quantization effects of the ADC and improve spectral performance. On the PXIe-5622, this out-of-band noise is added at low frequencies of up to approximately 12 MHz.

PXIe-5663/5663E/5665: When you enable dithering, the maximum signal level is reduced by up to 3 dB. This signal level reduction is accounted for in the nominal input ranges of the PXIe-5622. Therefore, you can overrange the input by up to 3 dB with dither disabled. For example, the +4 dBm input range can handle signal levels up to +7 dBm with dither disabled.

For wider bandwidth acquisitions, such as 40 MHz, disable dithering to eliminate residual leakage of the dither signal into the lower frequencies of the IF passband, which starts at 12.5 MHz and ends at 62.5 MHz. This leakage can slightly raise the noise floor in the lower frequencies, thus degrading the performance in high-sensitivity applications. When performing spectral measurements, this leakage can also appear as a wide, low-amplitude signal near the 12.5 MHz and 62.5 MHz frequencies. The width and amplitude of the signal depends on your resolution bandwidth and the type of time-domain window you apply to your FFT.

PXIe-5668: When you enable dithering, the maximum signal level is reduced by up to 2 dB. For the PXIe-5624, the maximum input power with dither off is 8 dBm and the maximum input power level with dither on is 6 dBm. When acquiring an 800 MHz bandwidth signal, the I/Q data contains the dither even if the dither signal is not in the displayed spectrum. The dither can affect actions like power level triggering.

Name (value)	Description
—	Note For the PXIe-5668, disabling dithering can negatively affect absolute amplitude accuracy.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Note

For PXIe-5820/5830/5831/5832/5840/5841/5842, only **Enabled** is supported.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

DIGITIZER_TEMPERATURE = 25

Returns the current temperature of the digitizer module. This value is expressed in degrees Celsius.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Note

This attribute is not supported if you are using an external digitizer.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5840/5841/5842/5860

DONE_EVENT_OUTPUT_TERMINAL = 115

Specifies the destination terminal for the Done event.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

DONE_EVENT_TERMINAL_NAME = 116

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/DoneEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/DoneEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/DoneEvent*, where *ModuleName* is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/DoneEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

`DOWNCONVERTER_CENTER_FREQUENCY = 13`

Enables in-band retuning and specifies the current frequency of the RF downconverter. This value is expressed in Hz.

After you set this attribute, the RF downconverter is locked to that frequency until the value is changed or the attribute is reset. Locking the downconverter to a fixed value allows frequencies within the instantaneous bandwidth of the downconverter to be measured without the overhead of retuning the LO and waiting for the LO to settle. This method is called in-band retuning and it has the highest benefit on analyzers that have larger LO settling times. After setting the downconverter center frequency, you can set the center frequency to the frequencies at which you want to take the measurements.

If you want to avoid the LO leakage or DC offset of analyzers that use a direct conversion architecture, it is more convenient to use the *DOWNCONVERTER_FREQUENCY_OFFSET* or *LO_LEAKAGE_AVOIDANCE_ENABLED* attributes.

If you set this attribute, any measurements outside the instantaneous bandwidth of the device are invalid. To disable in-band retuning, reset this attribute or call the `reset_to_default()` method.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is the carrier frequency or spectrum center frequency.

Valid Values: Any supported tuning frequency of the device.

> **Note**
>
> PXIe-5820: The only valid value for this attribute is 0 Hz.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

DOWNCONVERTER_FREQUENCY_OFFSET = 53

Specifies an offset from the center frequency value for the downconverter. Use this attribute to offset the measurement away from the LO leakage or DC Offset of analyzers that use a direct conversion architecture. You must set this attribute to half the bandwidth or span of the measurement + guardband. The guardband is needed to ensure that the LO leakage is not inside the analog or digital filter rolloffs. This value is expressed in Hz.

NI recommends using the *LO_LEAKAGE_AVOIDANCE_ENABLED* attribute instead of the Downconverter Frequency Offset attribute. The LO Leakage Avoidance Enabled attribute automatically configures the Downconverter Frequency Offset attribute to an appropriate offset based on the bandwidth or span of the measurement.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values: For spectrum acquisition types, the RFmx driver automatically calculates the default value to avoid residual LO power. For I/Q acquisition types, the default value is 0 Hz. If the center frequency is set to a non-multiple of *LO_FREQUENCY_STEP_SIZE* attribute, this attribute is set to compensate for the difference.

The following valid values correspond to their respective devices:

Name (value)	Description
PXIe-5646	-100 MHz to +100 MHz
PXIe-5830/5831/5832/5840	-500 MHz to +500 MHz
PXIe-5841 (200MHz Bandwidth)	-100 MHz to +100 MHz
PXIe-5841 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (500MHz Bandwidth)	-250 MHz to +250 MHz
PXIe-5842 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (2GHz Bandwidth)	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using Standard personality	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using the 4GHz Bandwidth personality	-2 GHz to +2 GHz
Other devices	-42 MHz to +42 MHz

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

DOWNCONVERTER_GAIN = 52

Returns the net signal gain for the device at the current RFmx settings and temperature. RFmx scales the acquired I/Q and spectrum data from the digitizer using the value of this attribute.

For a vector signal analyzer (VSA), the system is defined as the RF downconverter for all interfaces between the RF IN connector on the RF downconverter front panel and the IF IN connector on the digitizer front panel. For a spectrum monitoring receiver, the system is defined as the RF preselector, RF downconverter, and IF conditioning modules including all interfaces between the RF IN connector on the RF preselector module front panel and the IF IN connector on the digitizer front panel.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is N/A.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842/5860

DOWNCONVERTER_PRESELECTOR_ENABLED = 12

Specifies whether the tunable preselector is enabled on the downconverter.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the preselector.
Enabled (1)	The preselector is automatically enabled when it is in the signal path and is automatically disabled when it is not in the signal path. Use the Preselector Present attribute to determine if the downconverter has a preselector.

END_OF_RECORD_EVENT_OUTPUT_TERMINAL = 113

Specifies the destination terminal for the End of Record event.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

END_OF_RECORD_EVENT_TERMINAL_NAME = 114

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/EndOfRecordEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/EndOfRecordEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/EndOfRecordEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/EndOfRecordEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

FFT_WIDTH = 22

Specifies the FFT width of the device. The FFT width is the effective bandwidth of the signal path during each signal acquisition.

The lower limit for all devices that support setting the FFT Width attribute is 7.325 kHz.

PXIe-5663/5663E: The FFT width upper limit for the PXIe-5663/5663E depends on the RF frequency and on the module revision of the PXIe-5601. For more information about determining which revision of the PXIe-5601 RF downconverter you have installed, refer to the Identifying Module Revision topic in the *NI RF Vector Signal Analyzers Help*.

> **Note**
>
> The maximum FFT width for your device is constrained to 50 MHz or 25 MHz, depending on the digitizer option you purchased.

> **Note**
>
> You can use the FFT Width attribute with in-band retuning. For more information about in-band retuning, refer to the *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

The RFmx driver treats the device instantaneous bandwidth as the effective real-time bandwidth of the signal path. The span specifies the frequency range of the computed spectrum. A signal analyzer can acquire a bandwidth only within the device instantaneous bandwidth. If the span you choose is greater than the device instantaneous bandwidth, the RFmx driver obtains multiple acquisitions and combines them into a single spectrum. By specifying the FFT width, you can control the specific bandwidth obtained in each signal acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

FREQUENCY_REFERENCE_EXPORTED_TERMINAL = 34

Specifies a comma-separated list of the terminals at which to export the frequency reference.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None ()	The Reference Clock is not exported. This value is not valid for the PXIe-5644/5645/5646.
RefOut (RefOut)	Export the clock on the REF IN/OUT terminal on the PXIe-5652, the REF OUT terminals on the PXIe-5653, or the REF OUT terminal on the PXIe-5694, PXIe-5644/5645/5646, or PXIe-5820/5830/5831/5832/5840/5841/5860.
RefOut2 (RefOut2)	Export the clock on the REF OUT2 terminal on the PXIe-5652. This value is valid only for the PXIe-5663E.
ClkOut (ClkOut)	Export the Reference Clock on the CLK OUT terminal on the Digitizer. This value is not valid for the PXIe-5644/5645/5646 or PXIe-5820/5830/5831/5832/5840/5841/5842/5860.

FREQUENCY_REFERENCE_FREQUENCY = 3

Specifies the Reference Clock rate, when the *FREQUENCY_REFERENCE_SOURCE* attribute is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 10 MHz.

Valid values

Name (value)	Description
PXIe-5644/5645/5646, PXIe-5663/5663E, PXIe-5820/5830/5831/5832/5840/5841/5842	10 MHz
PXIe-5665/5668	5 MHz to 100 MHz (inclusive), in increments of 1 MHz
PXIe-5860	10 MHz, 100 MHz

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

FREQUENCY_REFERENCE_SOURCE = 2

Specifies the frequency reference source.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

All other devices default value is **OnboardClock**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value	Description
On-board-Clock (On-board-Clock)	PXIe-5663/5663E: The RFmx driver locks the PXIe-5663/5663E to the PXIe-5652 LO source onboard clock. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. On versions of the PXIe-5663/5663E that lack a REF OUT2 connector on the PXIe-5652, connect the REF IN/OUT connector on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: The RFmx driver locks the PXIe-5665 to the PXIe-5653 LO source onboard clock. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN terminal. PXIe-5668: Lock the PXIe-5668 to the PXIe-5653 LO SOURCE onboard clock. Connect the LO2 OUT connector on the PXIe-5606 to the CLK IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to its onboard clock. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. PXIe-5842: Lock to the associated PXIe-5655 onboard clock. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860: Lock to the PXIe-5860 onboard clock.
RefIn (RefIn)	PXIe-5663/5663E: Connect the external signal to the PXIe-5652 REF IN/OUT connector. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: Connect the external signal to the PXIe-5653 REF IN connector. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN connector. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5668: Connect the external signal to the PXIe-5653 REF IN connector. Connect the LO2 OUT on the PXIe-5606 to the CLK IN connector on the PXIe-5622. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to the signal at the external REF IN connector. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. For PXIe-5830, lock the external signal to the PXIe-3621 REF IN connector. For PXIe-5831, lock the external signal to the PXIe-3622 REF IN connector. For PXIe-5832, lock the external signal to the PXIe-3623 REF IN connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5842: Lock to the signal at the REF IN connector on the associated PXIe-5655. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860:Lock to the signal at the REF IN connector on the PXIe-5860.
PXI_C (PXI_C	PXIe-5668: Lock the PXIe-5653 to the PXI backplane clock. Connect the PXIe-5606 LO2 OUT to the LO2 IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5663/5663E/5665, and PXIe-5820/5840/5841/5860: The RFmx driver locks the device to the PXI backplane clock. PXIe-5830/5831/5832 with PXIe-5653/5841 with PXIe-5655, PXIe-5842/5860: The RFmx

FREQUENCY_SETTLING = 10

Specifies the value used for LO frequency settling.

Specify the units and interpretation for this scalar value using the *FREQUENCY_SETTLING_UNITS* attribute.

Valid values

Frequency Settling Units Property Value	PXIe-5663/566	PXIe-5665/566	PXIe-5644/5	PXIe-5830/5831/5832/5840/5841/5 PXIe-5831 with PXIe-5653 (using PXIe-3622 LO), PXIe-5832 with PXIe-5653 (using PXIe-3623 LO)	PXIe-5831 with PXIe-5653 (using PXIe-5653 LO) and PXIe-5832 with PXIe-5653 (using PXIe-5653 LO)
Seconds After Lock	2 µs to 80 ms, resolution of approximately 2 µs	4 µs to 80 ms, resolution of approximately 4 µs	1 µs to 65 ms, resolution of 1 µs	1 µs to 10s, resolution of 1 µs	4 µs to 80 ms, resolution of approximately 4 µs
Seconds After I/O	0 µs to 80 ms, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs	1 µs to 65 ms, resolution of 1 µs	0 µs to 10s, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs
PPM	1.0, 0.1, 0.01	1.0, 0.1, 0.01, 0.001	1.0, 0.1, 0.01	1.0 to 0.01	1.0 to 0.01

Name (value)	Description
—	Note This attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.1.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

FREQUENCY_SETTLING_UNITS = 9

Specifies the delay duration units and interpretation for LO settling.

Specify the actual settling value using the *FREQUENCY_SETTLING* attribute.

Name (value)	Description
—	Note The Frequency Settling Units attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **PPM**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
PPM (0)	Specifies the frequency settling in parts per million (ppm).
Seconds After Lock (1)	Specifies the frequency settling in time after lock (seconds).
Seconds After I/O (2)	Specifies the frequency settling in time after I/O (seconds).

IF_FILTER_BANDWIDTH = 48

Specifies the IF filter path bandwidth for your device configuration.

Name (value)	Description
—	Note For composite devices, such as the PXIe-5665/5668, the IF filter path bandwidth includes all IF filters across the component modules of a composite device.

RFmx chooses an appropriate IF filter as default IF Filter based on measurement configuration, center frequency, cleaner spectrum and downconverter preselector.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5665/5668

IF_OUTPUT_POWER_LEVEL_OFFSET = 17

Specifies the power offset by which to adjust the default IF output power level. This value is expressed in dB.

This attribute does not depend on absolute IF output power levels; therefore, you can use this attribute to adjust the IF output power level on all RFmx-supported devices without knowing the exact default value. Use this attribute to increase or decrease the nominal output level to achieve better measurement results.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

INPUT_ISOLATION_ENABLED = 92

Specifies whether input isolation is enabled.

Enabling this attribute isolates the input signal at the RF IN connector on the RF downconverter from the rest of the RF downconverter signal path. Disabling this attribute reintegrates the input signal into the RF downconverter signal path.

> **Note**
>
> If you enable input isolation for your device, the device impedance is changed from the characteristic 50-ohm impedance. A change in the device impedance may increase the VSWR value higher than the device specifications.

For PXIe-5830/5831/5832, input isolation is supported for all available ports for your hardware configuration.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

INSTRUMENT_FIRMWARE_REVISION = 27

Returns a string containing the firmware revision information of the RF downconverter for the composite device you are currently using.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

INSTRUMENT_MODEL = 28

Returns a string that contains the model number or name of the RF device that you are currently using.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

LO2_EXPORT_ENABLED = 58

Specifies whether to enable the LO2 OUT terminals in the installed devices.

Set this attribute to **Enabled** to export the 4 GHz LO signal from the LO2 IN terminal to the LO2 OUT terminal. You can also export the LO2 signal by setting the *LO_EXPORT_ENABLED* attribute to TRUE.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported Devices: PXIe-5665/5668

Name (Value)	Description
Disabled (0)	Disables the LO2 OUT terminals.
Enabled (1)	Enables the LO2 OUT terminals.

LOAD_OPTIONS = 163

Specifies the configurations to skip while loading from a file using the `load_configurations()` method .

Name (value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session

The default value is an empty array.

Name (Value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session.

LO_EXPORT_ENABLED = 33

Specifies whether to enable the LO OUT terminals on the installed devices.

Name (value)	Description
TRUE	Enables the LO OUT terminals.
FALSE	Disables the LO OUT terminals

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value:

- PXIe-5663/5663E: TRUE
- PXIe-5644/5645/5646, PXIe-5665/5668, PXIe-5830/5831/5832/5840/5841/5842: FALSE

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

LO_FREQUENCY = 60

Specifies the LO signal frequency for the configured center frequency. This value is expressed in Hz.

If you are using the vector signal analyzer with an external LO, use this attribute to specify the LO frequency that the external LO source passes into the LO IN or LO1 IN connector on the RF downconverter front panel. If you are using an external LO, reading the value of this attribute after configuring the rest of the parameters returns the LO frequency needed by the device.

You can set this attribute to the actual LO frequency because RFmx corrects for any difference between expected and actual LO frequencies.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector Strings topic for information about the string syntax.

The default value is 0.

Supported Devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

LO_FREQUENCY_STEP_SIZE = 95

Specifies the step size for tuning the LO phase-locked loop (PLL).

You can only tune the LO frequency in multiples of the LO Frequency Step Size attribute. Therefore, the LO frequency can be offset from the requested center frequency by as much as half of the LO Frequency Step Size attribute. This offset is corrected by digitally frequency shifting the LO frequency to the value requested in *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if PXIe-5653 is used as the LO source.

The valid values for this attribute depend on the *LO_PLL_FRACTIONAL_MODE* attribute.

PXIe-5644/5645/5646: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value.

PXIe-5840: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value that is less than or equal to the desired step size.

LO PLL Fractional Mode Enabled Property Setting	LO Frequency Step Size Property Valid Values on PXIe-5644/5645	LO Frequency Step Size Property Valid Values on PXIe-5646	LO Frequency Step Size Property Valid Values on PXIe-5840/5841	LO Frequency Step Size Property Valid Values on PXIe-5830/5831/583	LO Frequency Step Size Property Valid Values on PXIe-5841 with PXIe-5655, PXIe-5842*
Enabled	50 kHz to 24 MHz	50 kHz to 25 MHz	50 kHz to 100 MHz	LO1: 8 Hz to 400 MHz LO2: 4 kHz to 400 MHz	1 nHz to 50 MHz
Disabled	4 MHz, 5 MHz, 6 MHz, 12 MHz, 24 MHz	2 MHz, 5 MHz, 10 MHz, 25 MHz	1 MHz, 5 MHz, 10 MHz, 25 MHz, 50 MHz, 100 MHz	LO1: – LO2: –	1 nHz to 50 MHz

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5644/5645/5646	200 kHz
PXIe-5830	2 MHz
PXIe-5831/5832 (RF port)	8 MHz
PXIe-5831/5832 (IF port)	2 MHz, 4 MHz
PXIe-5840/5841	500 kHz
PXIe-5842	1 Hz

> **Note**
>
> The default value for PXIe-5831/5832 depends on the frequency range of the selected port for your instrument configuration. Use `get_available_ports()` method to get the valid port names.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

LO_INJECTION_SIDE = 18

Specifies the LO injection side.

Name (value)	Description
PXIe-5663/5(	For frequencies below 517.5 MHz or above 6.4125 GHz, the LO injection side is fixed, and the RFmx driver returns an error if you specify an incorrect value. If you do not configure this attribute, the RFmx driver selects the default LO injection side based on the downconverter center frequency. Reset this attribute to return to automatic behavior.
PXIe-5665 (3.6 GHz)	Setting this attribute to Low Side is not supported for this device.
PXIe-5665 (14 GHz)	Setting this attribute to Low Side is supported for this device for frequencies greater than 4 GHz, but this configuration is not calibrated, and device specifications are not guaranteed.
PXIe-5668	Setting this attribute to Low Side is supported for some frequencies in high band, varying by the final IF frequency. This configuration is not calibrated and device specifications are not guaranteed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

** Default value:**

- PXIe-5663/5663E (frequencies < 3.0 GHz): **High Side**
- PXIe-5663/5663E (frequencies >= 3.0 GHz): **Low Side**
- PXIe-5665/5668: **High Side**

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
High Side (0)	Configures the LO signal that the device generates at a frequency higher than the RF signal. This LO frequency is given by the following formula: *fLO = fRF + fIF *
Low Side (1)	Configures the LO signal that the device generates at a frequency lower than the RF signal. This LO frequency is given by the following formula: *fLO = fRF – fIF *

LO_IN_POWER = 78

Specifies the power level expected at the LO IN terminal when the *LO_SOURCE* attribute is set to **LO_In**. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> For PXIe-5644/5645/5646, this attribute is always read-only.

The default value is 0.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

LO_LEAKAGE_AVOIDANCE_ENABLED = 55

Specifies whether to reduce the effects of the instrument leakage by placing the LO outside the band of acquisition.

This attribute is ignored if:

- the bandwidth required by the measurement is more than the available instrument bandwidth after offsetting the LO.
- you set the *DOWNCONVERTER_CENTER_FREQUENCY* or *DOWNCONVERTER_FREQUENCY_OFFSET* attributes.

> **Note**
>
> When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set this attribute to **False** when the *LO_SOURCE* attribute is set to **Automatic_SG_SA_Shared**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5830/5831/5832/5840/5841/5842 is **True**, else the default value is **False**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
False (0)	RFmx does not modify the Downconverter Frequency Offset attribute.
True (1)	RFmx calculates the required LO offset based on the measurement configuration and appropriately sets the Downconverter Frequency Offset attribute.

LO_OUT_POWER = 79

Specifies the power level of the signal at the LO OUT terminal when the *LO_EXPORT_ENABLED* attribute is set to TRUE. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842

LO_PLL_FRACTIONAL_MODE = 90

Specifies whether to use fractional mode for the LO phase-locked loop (PLL).

Fractional mode provides a finer frequency step resolution, but may result in non harmonic spurs. Refer to the specifications document of your device for more information about fractional mode and non harmonic spurs.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

Note

The LO PLL Fractional Mode attribute is applicable only when using the internal LO.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if the PXIe-5653 is used as the LO source.

The default value is **Enabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

LO_SHARING_MODE = 68

Specifies the RFmx session with the respective LO sharing mode.

The following figures illustrate different connection configuration topologies for different LO Sharing modes.

You must set the *NUMBER_OF_LO_SHARING_GROUPS* attribute to 1 for the following LO connection configurations.

You must set the Num LO Sharing Groups attribute to 2 for the following LO connection configurations.

The default value is **Disabled**.

Name (Value)	Description
Disabled (0)	LO Sharing is disabled.
External Star (3)	The LO connection configuration is configured as External Star.
External Daisy Chain (4)	The LO connection configuration is configured as External Daisy Chain.
Splitter and Daisy Chain (5)	The LO connection configuration is configured as Splitter and Daisy Chain. With this option, the only allowed value for the Number of LO Sharing Groups attribute is 1.

LO_SOURCE = 59

Specifies the local oscillator (LO) signal source used to downconvert the RF input signal.

If this attribute is set to "" (empty string), RFmx uses the internal LO source. For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

If no signal downconversion is required, this attribute is ignored.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Onboard**.

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
None (None)	Specifies that no LO source is required to downconvert the RF input signal.
Onboard (Onboard)	Specifies that the onboard synthesizer is used to generate the LO signal that downconverts the RF input signal.
LO_In (LO_In)	Specifies that the LO source used to downconvert the RF input signal is connected to the LO IN connector on the front panel.
Secondary (Secondary)	Specifies that the LO source uses the PXIe-5830/5831/5832/5840 internal LO. This value is valid on only the PXIe-5840 with PXIe-5653, PXIe-5831 with PXIe-5653 (LO1 stage only), or PXIe-5832 with PXIe-5653 (LO1 stage only).
SG_SA_Shared (SG_SA_Shared)	Specifies that the internal LO can be shared between RFmx and RFSG sessions. RFmx selects an internal synthesizer and the synthesizer signal is switched to both the RX and TX mixers. This value is valid only on PXIe-5830/5831/5832/5841/5842.

Automatic_SG_SA_Sha (Automatic_SG_SA_Sha	Specifies whether RFmx automatically configures the signal analyzer to use the LO utilized by the signal generator on the same vector signal transceiver (VST) based on the configured measurements.
	When using instruments that do not have LOs with excellent phase noise and to minimize the contribution of the instrument's phase noise affecting your measurements, NI recommends to share the LO between the signal generator (SG) and the signal analyzer (SA).
	This value is recommended in test setups that use a VST with NI-RFSG to generate a signal at the DUT's input and RFmx to measure the signal at the DUT's output.
	This value automatically:
	• determines whether the SG LO can be shared with SA based on the test instrument used, selected measurement, and the measurement settings.
	• configures instrument specific attributes on SA to share the LO between the generator and analyzer, whenever possible.
	To enable automatically sharing SG LO with SA, you must first setup the required device specific physical connections mentioned below and then follow the steps in the recommended order.
	PXIe-5840/5841: SG LO is shared with SA via an external path. Hence, you must connect RF Out LO Out to RF In LO In using a cable.
	PXIe-5841 with PXIe-5655/5842/PXIe-5830/5831/5832: SG LO is shared with SA via an internal path. Hence, an external cable connection is not required.
	NI recommends the following order of steps:
	• Set LO Source attribute to Automatic SG SA Shared in NI-RFSG (or enable Automatic SG SA shared LO on NI-RFSG Playback Library).
	• Set LO Source attribute to **Automatic_SG_SA_Shared** in RFmx.
	• Configure any additional settings on RFSG and RFmx, including selecting waveforms.
	• Initiate RFSG.
	• Initiate RFmx.
	When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set the LO Leakage Avoidance Enabled attribute to **False** and LO Source attribute to **Automatic_SG_SA_Shared**.
	Refer to following methods for examples in RFmx WLAN and RFmx NR that show the behavior of Automatic SG SA Shared LO.
	`<LabVIEW directory>\examples\RFmx\WLAN\RFmxWLAN FEM Test with Automatic SG SA Shared LO.vi`
	`<LabVIEW directory>\examples\RFmx\NR\RFmxNR FEM Test with Automatic SG SA Shared LO.vi`
	This value is valid only on PXIe-5830/5831/5832/5840/5841/5842.

LO_SPLITTER_LOSS = 185

Specifies an array of the insertion losses inherent to the RF Splitter. This value is expressed in dB.

You must specify the frequencies at which the losses were measured using the *LO_SPLITTER_LOSS* attribute.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

LO_SPLITTER_LOSS_FREQUENCY = 184

Specifies the frequencies corresponding to the insertion loss inherent to the RF Splitter, as specified by the *LO_SPLITTER_LOSS_FREQUENCY* attribute. This value is expressed in Hz.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

LO_TEMPERATURE = 26

Returns the current temperature of the LO module associated with the device. This value is expressed in degrees Celsius.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

> **Note**
>
> This attribute is not supported if you are using an external LO.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5840/5841/5842

LO_VCO_FREQUENCY_STEP_SIZE = 80

Specifies the step size for tuning the internal voltage-controlled oscillator (VCO) used to generate the LO signal. The valid values for LO1 include 1 Hz to 50 MHz and for LO2 include 1 Hz to 100 MHz.

> **Note**
>
> Do not set this attribute with the *LO_FREQUENCY_STEP_SIZE* attribute.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 1 MHz.

Supported devices: PXIe-5830/5831/5832

MECHANICAL_ATTENUATION_AUTO = 6

Specifies whether the RFmx driver chooses an attenuation setting based on the hardware settings.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured in the Mechanical Attenuation Value attribute.
True (1)	Specifies that the measurement computes the mechanical attenuation.

MECHANICAL_ATTENUATION_VALUE = 7

Specifies the level of mechanical attenuation for the RF path. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *MECHANICAL_ATTENUATION_AUTO* attribute to **False**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Valid values

Name (value)	Description
PXIe-5663/5663E	0, 16
PXIe-5665 (3.6 GHz)	0, 10, 20, 30
PXIe-5665 (14 GHz), PXIe-5668	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668

MIXER_LEVEL = 16

Specifies the mixer level. This value is expressed in dBm.

The mixer level represents the attenuation value to apply to the input RF signal as it reaches the first mixer in the signal chain. If you do not set this attribute, the RFmx driver automatically selects an optimal mixer level value based on the reference level.

If you set the *MIXER_LEVEL* and *MIXER_LEVEL_OFFSET* attributes at the same time, the RFmx driver returns an error.

This attribute is read-only for PXIe-5663/5663E devices.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5665/5668	-10
All other devices	N/A

The valid values for this attribute depend on your device configuration.

Supported devices: PXIe-5663/5663E/5665/5668

MIXER_LEVEL_OFFSET = 15

Specifies the number of dB by which to adjust the device mixer level.

Specifying a positive value for this attribute configures the device for moderate distortion and low noise, and specifying a negative value results in low distortion and higher noise. You cannot set the *MIXER_LEVEL_OFFSET* and *MIXER_LEVEL* attributes at the same time.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0. The default value specifies device settings that are the best compromise between distortion and noise.

Supported devices: PXIe-5663/5663E/5665/5668

MODULE_REVISION = 29

Returns the revision of the RF downconverter module.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

> **Note**
>
> For PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860, this attribute returns the revision of the VST module.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

NUMBER_OF_LO_SHARING_GROUPS = 97

Specifies the RFmx session with the number of LO sharing groups.

The default value is 1.

The valid values are 1 and 2.

NUMBER_OF_RAW_IQ_RECORDS = 128

Returns the number of raw IQ records to acquire to complete measurement averaging.

> **Note**
>
> This attribute returns a value of 0 when RFmx cannot provide I/Q data for the specified measurement configuration.

OPTIMIZE_PATH_FOR_SIGNAL_BANDWIDTH = 91

Optimizes RF path for the signal bandwidth that is centered on the IQ carrier frequency.

You can disable this attribute to avoid changes to the RF path when changing the signal bandwidth.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Automatic**.

Supported devices: PXIe-5830/5831/5832/5841/5842

Name (Value)	Description
Disabled (0)	Disables the optimized path for signal bandwidth.
Enabled (1)	Enables the optimized path for signal bandwidth.
Automatic (2)	Automatically enables the optimized path based on other configurations.

OSP_DELAY_ENABLED = 23

Specifies whether to enable the digitizer OSP block to delay Reference Triggers, along with the data samples, moving through the OSP block.

If you set this attribute to **Disabled**, the Reference Triggers bypass the OSP block and are processed immediately.

Enabling this attribute requires the following equipment configurations:

- All digitizers being used must be the same model and hardware revision.
- All digitizers must use the same firmware.
- All digitizers must be configured with the same I/Q rate.
- All devices must use the same signal path.

For more information about the digitizer OSP block and Reference Triggers, refer to the following topics in the *NI High-Speed Digitizers Help*:

- PXIe-5622 Onboard Signal Processing (OSP)
- PXIe-5142 Onboard Signal Processing (OSP)
- PXIe-5622 Trigger Sources
- PXI-5142 Trigger Sources
- PXIe-5622 Block Diagram
- PXI-5142 Trigger Sources

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

OVERFLOW_ERROR_REPORTING = 77

Configures error reporting for ADC and overflows occurred during onboard signal processing. Overflows lead to clipping of the waveform.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Warning**.

Supported devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Warning (0)	RFmx returns a warning when an ADC or an onboard signal processing (OSP) overflow occurs.
Disabled (1)	RFmx does not return an error or a warning when an ADC or OSP overflow occurs.

PHASE_OFFSET = 19

Specifies the offset to apply to the initial I and Q phases.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values are -180 degrees to 180 degrees, inclusive.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

PREAMP_ENABLED = 14

Specifies whether the RF preamplifier is enabled in the system.

PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860: If you set this attribute to **Automatic**, RFmx selects the preamplifier state based on the value of the Reference Level attribute and the center frequency. For PXIe-5830/5831/5832, the value is not coerced.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842 is **Automatic**, else the default value is **Disabled**.

Name (Value)	Description
Disabled (0)	Disables the RF preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Enabled (1)	Enables the RF preamplifier when it is in the signal path and disables it when it is not in the signal path. Only devices with an RF preamplifier on the downconverter and an RF preselector support this option. Use the RF Preamp Present attribute to determine whether the downconverter has a preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Automatic (3)	Automatically enables the RF preamplifier based on the value of the reference level. **Supported Devices:** PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

PRESELECTOR_PRESENT = 31

Indicates whether a preselector is available on the RF downconverter module.

Name (value)	Description
TRUE	A preselector is available on the downconverter.
FALSE	No preselector is available on the downconverter.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842

READY_FOR_ADVANCE_EVENT_OUTPUT_TERMINAL = 109

Specifies the destination terminal for the Ready for Advance event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

READY_FOR_ADVANCE_EVENT_TERMINAL_NAME = 110

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForAdvanceEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForAdvanceEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForAdvanceEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1)
- **All other devices**: */DigitizerName/ReadyForAdvanceEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

`READY_FOR_REFERENCE_EVENT_OUTPUT_TERMINAL = 111`

Specifies the destination terminal for the Ready for Reference event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

READY_FOR_REFERENCE_EVENT_TERMINAL_NAME = 112

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForReferenceEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForReferenceEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForReferenceEvent*, where *BasebandModule*is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/ReadyForReferenceEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

`READY_FOR_START_EVENT_OUTPUT_TERMINAL = 107`

Specifies the destination terminal for the Ready for Start event.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

READY_FOR_START_EVENT_TERMINAL_NAME = 108

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForStartEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForStartEvent*, where *BasebandModule*is the name of the baseband module for your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForStartEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/ReadyForStartEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

RECOMMENDED_ACQUISITION_TYPE = 39

Returns the recommended acquisition type for the last committed measurement configuration.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Name (Value)	Description
IQ (0)	Indicates that the recommended acquisition type is I/Q. Use the Analyze (IQ) method to perform the measurement.
Spectral (1)	Indicates that the recommended acquisition type is Spectral. Use Analyze (Spectrum) method to perform the measurement.
IQ or Spectral (2)	Indicates that the recommended acquisition type is I/Q or Spectral. Use either Analyze (IQ) or Analyze (Spectrum) method to perform the measurement.

RECOMMENDED_CENTER_FREQUENCY = 57

Returns the recommended center frequency of the RF signal. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_IQ_ACQUISITION_TIME = 42

Returns the recommended acquisition time for I/Q acquisition. This value is expressed in seconds.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_IQ_MINIMUM_SAMPLE_RATE = 43

Returns the recommended minimum sample rate for I/Q acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_IQ_PRE_TRIGGER_TIME = 44

Returns the recommended pretrigger time for I/Q acquisition. This value is expressed in seconds.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_NUMBER_OF_RECORDS = 40

Returns the recommended number of records to acquire to complete measurement averaging.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_SPECTRAL_ACQUISITION_SPAN = 45

Returns the recommended acquisition span for spectral acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_SPECTRAL_FFT_WINDOW = 46

Returns the recommended FFT window type for spectral acquisition.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Name (Value)	Description
None (0)	Indicates that the measurement does not use FFT windowing to reduce spectral leakage.
Flat Top (1)	Indicates a Flat Top FFT window type.
Hanning (2)	Indicates a Hanning FFT window type.
Hamming (3)	Indicates a Hamming FFT window type.
Gaussian (4)	Indicates a Gaussian FFT window type.
Blackman (5)	Indicates a Blackman FFT window type.
Blackman-Harris (6)	Indicates a Blackman-Harris FFT window type.
Kaiser-Bessel (7)	Indicates a Kaiser-Bessel FFT window type.

RECOMMENDED_SPECTRAL_RESOLUTION_BANDWIDTH = 47

Returns the recommended FFT bin width for spectral acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RECOMMENDED_TRIGGER_MINIMUM_QUIET_TIME = 41

Returns the recommended minimum quiet time during which the signal level must be below the trigger value for triggering to occur. This value is expressed in seconds.

> **Note**
>
> This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

RF_ATTENUATION_AUTO = 4

Specifies whether the RFmx driver computes the RF attenuation.

If you set this attribute to **True**, the RFmx driver chooses an attenuation setting based on the reference level configured on the personality.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E, PXIe-5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured using RF Attenuation Value attribute.
True (1)	Specifies that the RFmx driver computes the RF attenuation.

RF_ATTENUATION_STEP_SIZE = 54

Specifies the step size for the RF attenuation level. This value is expressed in dB. The actual RF attenuation is coerced up to the next highest multiple of the specified step size. If the mechanical attenuators are not available to implement the coerced RF attenuation, the solid state attenuators are used.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0
PXIe-5603/5665 (3.6 GHz)	1.0
PXIe-5605/5665 (14 GHz), PXIe-5606/5668	5.0

Valid values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0 to 93.0, continuous
PXIe-5603/5665 (3.6 GHz)	1.0 to 74.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (low band), PXIe-5606/5668 (low band)	1.0 to 106.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (high band), PXIe-5606/5668 (high band)	5.0 to 75.0, in 5 dB steps

Supported devices: PXIe-5663, PXIe-5665, PXIe-5668

RF_ATTENUATION_VALUE = 5

Specifies the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *RF_ATTENUATION_AUTO* attribute to **False**.

Name (value)	Description
PXIe-5663/5663E	You can change the attenuation value to modify the amount of noise and distortion. Higher attenuation levels increase the noise level but decreases distortion; lower attenuation levels decrease the noise level but increases distortion.
PXIe-5603/5605/5	Refer to the PXIe-5665 or the PXIe-5668 RF Attenuation and Signal Levels topic in the NI RF Vector Signal Analyzers Help for more information about configuring attenuation.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The valid values for this attribute depend on the device configuration.

Supported devices: PXIe-5663/5663E/5603/5605/5665/5668

RF_HIGHPASS_FILTER_FREQUENCY = 49

Specifies the maximum corner frequency of the high pass filter in the RF signal path. The device uses the highest frequency high-pass filter option below or equal to the value you specify and returns a coerced value. Specifying a value of 0 disables high pass filtering silly.

For multispan acquisitions, the device uses the appropriate filter for each subspan during acquisition, depending on the details of your application and the value you specify. In multispan acquisition spectrum applications, this attribute returns the value you specified rather than a coerced value if multiple high-pass filters are used during the acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

The valid values range from 0 to 26.5.

Supported devices: PXIe-5668

RF_PREAMP_PRESENT = 32

Indicates whether an RF preamplifier is available on the RF downconverter module.

Name (value)	Description
TRUE	A preamplifier is available on the downconverter.
FALSE	No preamplifier is available on the downconverter.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842

SELF_CALIBRATION_VALIDITY_CHECK = 117

Specifies whether the RFmx driver validates the self-calibration data.

You can specify the time interval required to perform the check using the *SELF_CALIBRATION_VALIDITY_CHECK_TIME_INTERVAL* attribute.

NI recommends to perform self-calibration using the `self_calibrate()` method when RFmx reports an invalid self-calibration data warning.

Note

The RFmx driver does not consider self-calibration range data during self calibration validity check.

The default value is **Off**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

Name (Value)	Description
Off (0)	Indicates that RFmx does not check whether device self-calibration data is valid.
Enabled (1)	Indicates that RFmx checks whether device self-calibration data is valid and reports a warning from the RFmx Commit and RFmx Initiate methods when the data is invalid.

SELF_CALIBRATION_VALIDITY_CHECK_TIME_INTERVAL = 118

Specifies the minimum time between two self calibration validity checks. This value is expressed in seconds.

When you call RFmx Commit or Initiate methods by enabling the *SELF_CALIBRATION_VALIDITY_CHECK* attribute, the RFmx driver checks if the amount of time specified by the Self Calibration Validity Check Time Interval attribute has elapsed before validating the calibration data.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

SERIAL_NUMBER = 30

Returns the serial number of the RF downconverter module.

> **Note**
>
> For PXIe-5644/5645/5646 and PXIe-5820/5840/5841/5842/5860, this attribute returns the serial number of the VST module. For PXIe-5830/5831/5832, this attribute returns the serial number of PXIe-3621/3622/3623.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the `build_port_string()` method to build the selector string.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

SMU_CHANNEL = 72

Specifies the output channel to be used for noise figure (NF) measurement in RFmx.

The default value is 0.

SMU_RESOURCE_NAME = 71

Specifies the resource name assigned by Measurement and Automation Explorer (MAX) for NI Source Measure Units (SMU) which is used as the noise source power supply for Noise Figure (NF) measurement, for example, PXI1Slot3, where PXI1Slot3 is an instrument resource name. SMU Resource Name can also be a logical IVI name.

Supported devices: PXIe-4138, PXIe-4139, PXIe-4139 (40 W), and PXIe-4143 SMUs.

START_TRIGGER_DIGITAL_EDGE = 100

Specifies the active edge for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Rising Edge**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Rising Edge (0)	The trigger asserts on the rising edge of the signal.
Falling Edge (1)	The trigger asserts on the falling edge of the signal.

START_TRIGGER_DIGITAL_EDGE_SOURCE = 99

Specifies the source terminal for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

START_TRIGGER_EXPORT_OUTPUT_TERMINAL = 101

Specifies the destination terminal for the exported start trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

START_TRIGGER_TERMINAL_NAME = 102

Returns the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/StartTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/StartTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/StartTrigger,*, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/StartTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

START_TRIGGER_TYPE = 98

Specifies whether the start trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No start trigger is configured.
Digital Edge (1)	The start trigger is not asserted until a digital edge is detected. The source of the digital edge is specified by the Start Trigger Digital Edge Source attribute.
Software (3)	The start trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

SUBSPAN_OVERLAP = 50

Use subspan overlap process to eliminate or reduce analyzer spurs. To enable this feature, specify a non-zero percentage overlap between consecutive subspans in a spectrum acquisition.

If a value greater than 0 is specified, then for each spectral line in the resulting spectrum, the driver acquires data twice with slightly different hardware settings, so that the analyzer spurs, if any, are present at different frequencies in the two acquisitions. Typically, LO frequency is shifted between the acquisitions causing analyzer spurs that are relative to the LO frequency, to move from one frequency to another. Those spurs, which are present in only one of the acquisitions for each spectral line, get removed.

The subspan overlap feature will not remove any spurs from the Device Under Test or modify the signal being measured; unlike the analyzer spurs, the spurs in the signal being measured stay at a constant frequency in the two acquisitions.

> **Note**
>
> Subspan overlap process effectively is performing minimum averaging, which might reduce the measured noise floor level. RFmx Spectrum Averaging can be enabled to minimize the effect of subspan overlap on the noise floor.

> **Note**
>
> RFmx may apply further shifts to the specified value to accommodate fixed-frequency edges of components such as preselectors.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values

Name (value)	Description
PXIe-5820/5830/5831/5832/5840/5841/5860	0
PXIe-5842	0, 50
PXIe-5665/5668	0 to <100

Supported devices: PXIe-5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

TEMPERATURE_READ_INTERVAL = 119

Specifies the minimum time difference between temperature sensor readings. This value is expressed in seconds.

When you call the RFmx Initiate method, RFmx checks if the amount of time specified by this attribute has elapsed before reading the hardware temperature.

> **Note**
>
> RFmx ignores Temperature Read Interval attribute if you read the *DOWNCONVERTER_GAIN* attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

THERMAL_CORRECTION_HEADROOM_RANGE = 94

Specifies the expected thermal operating range of the instrument from the self-calibration temperature returned from the *DEVICE_TEMPERATURE* attribute. This value is expressed in degree Celsius.

For example, if this attribute is set to 5.0, and the device is self-calibrated at 35 degrees Celsius, then you can expect to run the device from 30 degrees Celsius to 40 degrees Celsius with corrected accuracy and no overflows. Setting this attribute with a smaller value can result in improved dynamic range, but you must

ensure thermal stability while the instrument is running. Operating the instrument outside of the specified range may cause degraded performance and ADC or DSP overflows.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	5
PXIe-5840/5841	10

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

THERMAL_CORRECTION_TEMPERATURE_RESOLUTION = 120

Specifies the temperature change required before RFmx recalculates the thermal correction settings when entering the running state. This value is expressed in degree Celsius.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	0.2
PXIe-5840/5841	1.0

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

TRIGGER_EXPORT_OUTPUT_TERMINAL = 35

Specifies the destination terminal for the exported Reference Trigger. You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646 and PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists on only PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for the PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid on only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

TRIGGER_TERMINAL_NAME = 36

Returns the fully qualified signal name as a string.

> **Note**
>
> This attribute is not supported on a MIMO session.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/RefTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/RefTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/RefTrigger,*, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/RefTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

`TUNING_SPEED = 8`

Makes tradeoffs between tuning speed and phase noise.

> **Note**
>
> This attribute is not supported if you are using an external LO.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use `build_lo_string()` utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> The PXIe-5830/5831/5832/5840/5841/5842 supports only **Medium** for this attribute.

Default value: **Normal** for PXIe-5663/5663E/5665/5668, **Medium** for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Normal (0)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an underdamped response. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a narrow loop bandwidth.
Mediun (1)	Specifies that the RF downconverter module uses a medium loop bandwidth. This value is not supported on PXIe-5663/5663E/5665/5668 devices.
Fast (2)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an overdamped response. Setting this attribute to **Fast** allows the frequency to settle significantly faster for some frequency transitions at the expense of increased phase noise. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a wide loop bandwidth.

5.2 Enums

enums.py - Contains enum classes.

class nirfmxinstr.enums.**AdvanceTriggerType**(*value*)

Bases: Enum

AdvanceTriggerType.

DIGITAL_EDGE = 1

The advance trigger is not asserted until a digital edge is detected. The source of the digital edge is specified with the *ADVANCE_TRIGGER_DIGITAL_EDGE_SOURCE* attribute.

NONE = 0

No advance trigger is configured.

SOFTWARE = 3

The advance trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

class nirfmxinstr.enums.**ChannelCoupling**(*value*)

Bases: Enum

ChannelCoupling.

AC_COUPLED = 0

Specifies that the RF input channel is AC-coupled. For low frequencies (<10 MHz), accuracy decreases because RFmxInstr does not calibrate the configuration.

DC_COUPLED = 1

Specifies that the RF input channel is DC-coupled. The RFmx driver enforces a minimum RF attenuation for device protection.

class nirfmxinstr.enums.**CleanerSpectrum**(*value*)

Bases: Enum

CleanerSpectrum.

DISABLED = 0

Disable this attribute to get faster measurement speed.

ENABLED = 1

Enable this attribute to get the lowest noise floor and avoid digitizer spurs.

class nirfmxinstr.enums.**DigitizerDitherEnabled**(*value*)

Bases: Enum

DigitizerDitherEnabled.

DISABLED = 0

Disables the attribute.

ENABLED = 1

Enables the attribute.

class nirfmxinstr.enums.**DownconverterPreselectorEnabled**(*value*)

Bases: Enum

DownconverterPreselectorEnabled.

DISABLED = 0

Disables the preselector.

ENABLED = 1

The preselector is automatically enabled when it is in the signal path and is automatically disabled when it is not in the signal path. Use the *PRESELECTOR_PRESENT* attribute to determine if the downconverter has a preselector.

class nirfmxinstr.enums.**ExportSignalSource**(*value*)

Bases: Enum

ExportSignalSource.

ADVANCE_TRIGGER = 2

Advance trigger event is sourced.

DONE_EVENT = 7

Done Event is sourced.

END_OF_RECORD_EVENT = 6

End Of Record Event is sourced.

READY_FOR_ADVANCE_EVENT = 5

Ready For Advance Event is sourced.

READY_FOR_REFERENCE_EVENT = 4

Ready For Reference Event is sourced.

READY_FOR_START_EVENT = 3

Ready For Start Event is sourced.

REFERENCE_CLOCK = 8

Reference Clock is sourced.

REFERENCE_TRIGGER = 1

Reference trigger event is sourced.

START_TRIGGER = 0

Start trigger is sourced.

class nirfmxinstr.enums.**FrequencySettlingUnits**(*value*)

Bases: Enum

FrequencySettlingUnits.

PPM = 0

Specifies the frequency settling in parts per million (ppm).

SECONDS_AFTER_IO = 2

Specifies the frequency settling in time after I/O (seconds).

SECONDS_AFTER_LOCK = 1

Specifies the frequency settling in time after lock (seconds).

class nirfmxinstr.enums.**InputIsolationEnabled**(*value*)

Bases: Enum

InputIsolationEnabled.

DISABLED = 0

Indicates that the attribute is disabled.

ENABLED = 1

Indicates that the attribute is enabled.

class nirfmxinstr.enums.**LO2ExportEnabled**(*value*)

Bases: Enum

LO2ExportEnabled.

DISABLED = 0

Disables the LO2 OUT terminals.

ENABLED = 1

Enables the LO2 OUT terminals.

class nirfmxinstr.enums.**LOInjectionSide**(*value*)

Bases: Enum

LOInjectionSide.

HIGH_SIDE = 0

Configures the LO signal that the device generates at a frequency higher than the RF signal. This LO frequency is given by the following formula: *$f_{LO} = f_{RF} + f_{IF}$*

LOW_SIDE = 1

Configures the LO signal that the device generates at a frequency lower than the RF signal. This LO frequency is given by the following formula: *$f_{LO} = f_{RF} - f_{IF}$*

class nirfmxinstr.enums.**LOLeakageAvoidanceEnabled**(*value*)

Bases: Enum

LOLeakageAvoidanceEnabled.

FALSE = 0

RFmx does not modify the Downconverter Frequency Offset attribute.

TRUE = 1

RFmx calculates the required LO offset based on the measurement configuration and appropriately sets the Downconverter Frequency Offset attribute.

class nirfmxinstr.enums.**LOPllFractionalMode**(*value*)

Bases: Enum

LOPllFractionalMode.

DISABLED = 0

Indicates that the attribute is disabled.

ENABLED = 1

Indicates that the attribute is enabled.

class nirfmxinstr.enums.**LOSharingMode**(*value*)

Bases: Enum

LOSharingMode.

DISABLED = 0

LO Sharing is disabled.

EXTERNAL_DAISY_CHAIN = 4

The LO connection configuration is configured as External Daisy Chain.

EXTERNAL_STAR = 3

The LO connection configuration is configured as External Star.

SPLITTER_AND_DAISY_CHAIN = 5

The LO connection configuration is configured as Splitter and Daisy Chain.

With this option, the only allowed value for the *NUMBER_OF_LO_SHARING_GROUPS* attribute is 1.

class nirfmxinstr.enums.**LinearInterpolationFormat**(*value*)

Bases: Enum

LinearInterpolationFormat.

MAGNITUDE_AND_PHASE = 1

Results in a linear interpolation.

MAGNITUDE_DB_AND_PHASE = 2

Results in a linear interpolation.

REAL_AND_IMAGINARY = 0

Results in a linear interpolation of the real portion of the complex number and a separate linear interpolation of the complex portion.

class nirfmxinstr.enums.**LoadOptions**(*value*)

Bases: Enum

LoadOptions.

SKIP_NONE = 0

RFmx loads all the configurations to the session.

SKIP_RFINSTR = 1

RFmx skips loading the RFmxInstr configurations to the session.

class nirfmxinstr.enums.**MechanicalAttenuationAuto**(*value*)

Bases: Enum

MechanicalAttenuationAuto.

FALSE = 0

Specifies that the RFmx driver uses the value configured in the *MECHANICAL_ATTENUATION_VALUE* attribute.

TRUE = 1

Specifies that the measurement computes the mechanical attenuation.

class nirfmxinstr.enums.**OptimizePathForSignalBandwidth**(*value*)

Bases: Enum

OptimizePathForSignalBandwidth.

AUTOMATIC = 2

Automatically enables the optimized path based on other configurations.

DISABLED = 0

Disables the optimized path for signal bandwidth.

ENABLED = 1

Enables the optimized path for signal bandwidth.

class nirfmxinstr.enums.**OspDelayEnabled**(*value*)

Bases: Enum

OspDelayEnabled.

DISABLED = 0

Disables the attribute.

ENABLED = 1

Enables the attribute.

class nirfmxinstr.enums.**OverflowErrorReporting**(*value*)

Bases: Enum

OverflowErrorReporting.

DISABLED = 1

RFmx does not return an error or a warning when an ADC or OSP overflow occurs.

WARNING = 0

RFmx returns a warning when an ADC or an onboard signal processing (OSP) overflow occurs.

class nirfmxinstr.enums.**Personalities**(*value*)

Bases: IntFlag

Personalities.

ALL = 2147483647

Specifies all the personalities.

BT = 1024

Specifies the BT personality.

CDMA2K = 32

Specifies the CDMA2k personality.

DEMOD = 2

Specifies the Demod personality.

EVDO = 128

Specifies the EV-DO personality.

GSM = 8

Specifies the GSM personality.

LTE = 4

Specifies the LTE personality.

NONE = 0

Specifies that a signal does not exist.

NR = 256

Specifies the NR personality.

PULSE = 2048

Specifies the Pulse personality.

SPECAN = 1

Specifies the SpecAn personality.

TDSCDMA = 64

Specifies the TD-SCDMA personality.

UWB = 8192

Specifies the UWB personality.

VNA = 4096

Specifies the VNA personality.

WCDMA = 16

Specifies the WCDMA personality.

WLAN = 512

Specifies the WLAN personality.

class nirfmxinstr.enums.**PreampEnabled**(*value*)

Bases: Enum

PreampEnabled.

AUTOMATIC = 3

Automatically enables the RF preamplifier based on the value of the reference level.

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

DISABLED = 0

Disables the RF preamplifier.

Supported Devices: PXIe-5663/5663E/5665/5668

ENABLED = 1

Enables the RF preamplifier when it is in the signal path and disables it when it is not in the signal path. Only devices with an RF preamplifier on the downconverter and an RF preselector support this option. Use the *RF_PREAMP_PRESENT* attribute to determine whether the downconverter has a preamplifier.

Supported Devices: PXIe-5663/5663E/5665/5668

class nirfmxinstr.enums.**RFAttenuationAuto**(*value*)

Bases: Enum

RFAttenuationAuto.

FALSE = 0

Specifies that the RFmx driver uses the value configured using *RF_ATTENUATION_VALUE* attribute.

TRUE = 1

Specifies that the RFmx driver computes the RF attenuation.

class nirfmxinstr.enums.**RecommendedAcquisitionType**(*value*)

Bases: Enum

RecommendedAcquisitionType.

IQ = 0

Indicates that the recommended acquisition type is I/Q. Use the Analyze (IQ) method to perform the measurement.

IQ_OR_SPECTRAL = 2

Indicates that the recommended acquisition type is I/Q or Spectral. Use either Analyze (IQ) or Analyze (Spectrum) method to perform the measurement.

SPECTRAL = 1

Indicates that the recommended acquisition type is Spectral. Use Analyze (Spectrum) method to perform the measurement.

class nirfmxinstr.enums.**RecommendedSpectralFftWindow**(*value*)

Bases: Enum

RecommendedSpectralFftWindow.

BLACKMAN = 5

Indicates a Blackman FFT window type.

BLACKMAN_HARRIS = 6

Indicates a Blackman-Harris FFT window type.

FLAT_TOP = 1

Indicates a Flat Top FFT window type.

GAUSSIAN = 4

Indicates a Gaussian FFT window type.

HAMMING = 3

Indicates a Hamming FFT window type.

HANNING = 2

Indicates a Hanning FFT window type.

KAISER_BESSEL = 7

Indicates a Kaiser-Bessel FFT window type.

NONE = 0

Indicates that the measurement does not use FFT windowing to reduce spectral leakage.

class nirfmxinstr.enums.**SParameterOrientation**(*value*)

Bases: Enum

SParameterOrientation.

PORT1_TOWARDS_DUT = 0

Port 1 of the S2P is oriented towards the DUT.

PORT2_TOWARDS_DUT = 1

Port 2 of the S2P is oriented towards the DUT.

class nirfmxinstr.enums.**SParameterType**(*value*)

Bases: Enum

SParameterType.

SCALAR = 1

De-embeds the measurement using the gain term.

VECTOR = 2

De-embeds the measurement using the gain term and the reflection term.

class nirfmxinstr.enums.**SelfCalibrateSteps**(*value*)

Bases: IntFlag

SelfCalibrateSteps.

AMPLITUDE_ACCURACY = 32

Selects/Omits the Amplitude Accuracy self-calibration step.

DC_OFFSET = 512

Selects/Omits the DC Offset self-calibration step.

DIGITIZER_SELF_CAL = 8

Selects/Omits the Digitizer Self Cal self-calibration step.

GAIN_REFERENCE = 2

Selects/Omits the Gain Reference self-calibration step.

IF_FLATNESS = 4

Selects/Omits the IF Flatness self-calibration step.

IMAGE_SUPPRESSION = 128

Selects/Omits the Image Suppression self-calibration step.

LO_SELF_CAL = 16

Selects/Omits the LO Self Cal self-calibration step.

NONE = 0

A value of None specifies that all calibration steps are performed.

PRESELECTOR_ALIGNMENT = 1

Selects/Omits the Preselector Alignment self-calibration step.

RESIDUAL_LO_POWER = 64

Selects/Omits the Residual LO Power self-calibration step.

SYNTHESIZER_ALIGNMENT = 256

Selects/Omits the Synthesizer Alignment self-calibration step.

class nirfmxinstr.enums.**SelfCalibrationValidityCheck**(*value*)

Bases: Enum

SelfCalibrationValidityCheck.

ENABLED = 1

Indicates that RFmx checks whether device self-calibration data is valid and reports a warning from the RFmx Commit and RFmx Initiate methods when the data is invalid.

OFF = 0

Indicates that RFmx does not check whether device self-calibration data is valid.

class nirfmxinstr.enums.**StartTriggerDigitalEdge**(*value*)

Bases: Enum

StartTriggerDigitalEdge.

FALLING = 1

The trigger asserts on the falling edge of the signal.

RISING = 0

The trigger asserts on the rising edge of the signal.

class nirfmxinstr.enums.**StartTriggerType**(*value*)

Bases: Enum

StartTriggerType.

DIGITAL_EDGE = 1

The start trigger is not asserted until a digital edge is detected. The source of the digital edge is specified by the *START_TRIGGER_DIGITAL_EDGE_SOURCE* attribute.

NONE = 0

No start trigger is configured.

SOFTWARE = 3

The start trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

class nirfmxinstr.enums.**TuningSpeed**(*value*)

Bases: Enum

TuningSpeed.

FAST = 2

Adjusts the YIG main coil on the LO for an overdamped response. Setting this attribute to **Fast** allows the frequency to settle significantly faster for some frequency transitions at the expense of increased phase noise.

PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a wide loop bandwidth.

Type
PXIe-5665/5668

MEDIUM = 1

Specifies that the RF downconverter module uses a medium loop bandwidth. This value is not supported on PXIe-5663/5663E/5665/5668 devices.

NORMAL = 0

Adjusts the YIG main coil on the LO for an underdamped response.

PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a narrow loop bandwidth.

Type
PXIe-5665/5668

5.3 Errors

errors.py - Contains error classes and method(s) to handle error.

exception nirfmxinstr.errors.**DriverNotInstalledError**

Bases: *Error*

An error due to using this module without the driver runtime installed.

exception nirfmxinstr.errors.**DriverTooNewError**

Bases: *Error*

An error due to the RFmx Instr driver runtime being too new for this module.

exception nirfmxinstr.errors.**DriverTooOldError**

Bases: *Error*

An error due to using this module with an older version of the RFmx Instr driver runtime.

exception nirfmxinstr.errors.**Error**(*message*)

Bases: Exception

Base error class for RFmx Instr.

exception nirfmxinstr.errors.**RFmxError**(*code*, *description*)

Bases: *Error*

An error originating from the RFmx Instr driver.

exception nirfmxinstr.errors.**RpcError**(*rpc_code*, *description*)

Bases: *Error*

An error specific to sessions to the NI gRPC Device Server.

exception nirfmxinstr.errors.**UnsupportedConfigurationError**

Bases: *Error*

An error due to using this module in an usupported platform.

nirfmxinstr.errors.**handle_error**(*library_interpreter: Any*, *code: int*, *ignore_warnings: bool*, *is_error_handling: bool*) → None

Helper function for handling errors returned by RFmx Instr Library.

It calls back into the LibraryInterpreter to get the corresponding error description and raises if necessary.

5.4 gRPC Support

Collection of options that specifies session behaviors related to gRPC.

class nirfmxinstr.grpc_session_options.**GrpcSessionOptions**(*grpc_channel*, *session_name*, ** (Keyword-only parameters separator (PEP 3102))*, *initialization_behavior=SessionInitializationBehavior.AUTO*)

Bases: object

Collection of options that specifies session behaviors related to gRPC.

__init__(*grpc_channel*, *session_name*, ***, *initialization_behavior=SessionInitializationBehavior.AUTO*)

Collection of options that specifies session behaviors related to gRPC.

Creates and returns an object you can pass to a Session constructor.

Parameters

- **grpc_channel** (*grpc.Channel*) – Specifies the channel to the NI gRPC Device Server.
- **session_name** (*str*) – User-specified name that identifies the driver session on the NI gRPC Device Server. This is different from the resource name parameter many APIs take as a separate parameter. Specifying a name makes it easy to share sessions across multiple gRPC clients. You can use an empty string if you want to always initialize a new session on the server. To attach to an existing session, you must specify the session name it was initialized with.
- **initialization_behavior** (*enum*) – Specifies whether it is acceptable to initialize a new session or attach to an existing one, or if only one of the behaviors is desired. The driver session exists on the NI gRPC Device Server.

class nirfmxinstr.grpc_session_options.**SessionInitializationBehavior**(*value*)

Bases: IntEnum

ATTACH_TO_SERVER_SESSION = 2

Require the NI gRPC Device Server to attach to an existing session with the specified name.

Note: When using the Session as a context manager and the context exits, it will detach from the server session and leave it open.

AUTO = 0

The NI gRPC Device Server will attach to an existing session with the specified name if it exists, otherwise the server will initialize a new session.

Note: When using the Session as a context manager and the context exits, the behavior depends on what happened when the constructor was called. If it resulted in a new session being initialized on the NI gRPC Device Server, then it will automatically close the server session. If it instead attached to an existing session, then it will detach from the server session and leave it open.

INITIALIZE_SERVER_SESSION = 1

Require the NI gRPC Device Server to initialize a new session with the specified name.

Note: When using the Session as a context manager and the context exits, it will automatically close the server session.

5.5 RFInstr

session.py - Defines a root class that identifies and controls the instrument session.

class nirfmxinstr.session.**Session**(*resource_name*, *option_string=''*, *instrument_handle=None*, ***, *grpc_options=None*)

Bases: _SessionBase

Defines a root class that identifies and controls the instrument session.

__init__(*resource_name*, *option_string=''*, *instrument_handle=None*, ***, *grpc_options=None*)

Creates an RFmx session to the device you specify through the resource_name, option_string or instrument_handle parameter(s).

Parameters

- **resource_name** (*string*) – Specifies the resource name of the device to initialize.
- **option_string** (*string*) – Sets the initial value of certain properties for the session. The following attributes are used in this parameter: RFmxSetup, Simulate, AnalysisOnly. To simulate a device using the PXIe-5622 (25 MHz) digitizer, set the Digitizer field to 5622_25MHz_DDC and the Simulate field to 1. You can set the Digitizer field to 5622_25MHz_DDC only when using the PXIe-5665. To use AnalysisOnly mode, specify the string as "AnalysisOnly=1". While using this mode, you are responsible for waveform acquisition and RFmx will perform analysis on the I/Q waveform or Spectrum you specify. You must use personality specific Analyze functions to perform the measurements. To use external NI Source Measure Units (SMU) as the noise source power supply for the Noise Figure (NF) measurement, use "NoiseSourcePowerSupply" as the specifier within the RFmxSetup string. For example, "RFmxSetup= NoiseSourcePowerSupply:myDCPower[0]" configures RFmx to use channel 0 on myDCPower SMU device for powering the noise source. You should allocate a dedicated SMU channel for RFmx. RFmx supports PXIe-4138, PXIe-4139, and PXIe-4139 (40 W) SMUs. To set multiple attributes, separate their assignments with a comma.
- **instrument_handle** (*int*) – Specifies the pre-existing instrument handle used to create a new RFmx session.
- **grpc_options** (nirfmxinstr.grpc_session_options.GrpcSessionOptions) – Specifies the gRPC session options.

Returns
The RFmx session object.

Return type
session (*Session*)

static build_calibration_plane_string(*calibration_plane_name*)

Creates the selector string to use with External Attenuation Table methods.

Parameters
calibration_plane_name (*string*) – Specifies the calibration plane name for building the selector string. This input accepts the calibration plane name with or without the "calplane::" prefix.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

static build_instrument_string(*selector_string*, *instrument_number*)

Creates the instrument string to use as the Selector String for reading the recommended settings.

Parameters
instrument_number (*int*)

Returns
Contains the selector string created by this method.

Return type
string

static build_lo_string(*selector_string*, *lo_index*)

Creates the LO string to use as the Selector String for LO related attributes.

Parameters
lo_index (*int*) – This parameter specifies the LO index for building the selector string.

Returns
Contains the selector string created by this method.

Return type
string

static build_module_string(*selector_string*, *module_name*)

Configures the module string to use as the Selector String for reading temperature of specific modules of the device.

Parameters
module_name (*string*) – This parameter specifies the module for which you want the temperature to be read.

Returns
Contains the selector string created by this method.

Return type
string

static build_port_string(*selector_string*, *port_name*, *device_name*, *channel_number*)

Creates the port string to use as the selector string with External Attenuation Table methods.

On a MIMO session, this method can be used to build port string to use as a selector string for configuring or reading port-specific methods and external attenuation table methods.

Parameters

- **selector_string** (*string*) – Specifies the calibration plane string when used for building port string for the external attenuation table methods. If you do not specify the calibration plane string, the default calibration plane instance is used.

  Example:

  ””

  ”calplane::plane0”

- **port_name** (*string*) – Specifies the port for building the selector string.
- **device_name** (*string*) – Specifies the name of the initialized device for building the selector string.
- **channel_number** (*int*) – Specifies the channel for building the selector string. Specify 0 as the value for this parameter.

Returns
Contains the selector string created by this method.

Return type
int

`check_acquisition_status()`

Checks the status of the acquisition. Use this method to check for any errors that may occur during acquisition, or to check whether RFmx has completed the acquisition operation.

Returns

acquisition_done (bool):
This parameter indicates whether the acquisition is complete. The default value is FALSE.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (acquisition_done, error_code)

`check_if_signal_exists(signal_name)`

Returns whether the signal you specify in the **Signal Name** parameter exists, and also returns the corresponding personality of the signal, if the signal exists. This method does not support an empty ("") signal name.

Parameters
signal_name (*string*) – This parameter specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.

Example:

"signal::sig1"

"sig1"

Returns

signal_configuration_exists (bool):
This parameter indicates whether the signal exists or not.

Name (value)	Description
FALSE	Indicates that the signal does not exist.
TRUE	Indicates that the signal exists.

personality (enums.Personalities):
This parameter indicates the personality of the signal if the signal exists.

Name (Value)	Description
None (0)	Indicates that the given signal does not exist.
SpecAn (1)	Indicates that the signal personality is SpecAn.
Demod (2)	Indicates that the signal personality is Demod.
LTE (4)	Indicates that the signal personality is LTE.
GSM (8)	Indicates that the signal personality is GSM.
WCDMA (16)	Indicates that the signal personality is WCDMA.
CDMA2k (32)	Indicates that the signal personality is CDMA2k.
TDSCDMA (64)	Indicates that the signal personality is TD-SCDMA.
EVDO (128)	Indicates that the signal personality is EV-DO.
NR (256)	Indicates that the signal personality is NR.
WLAN (512)	Indicates that the signal personality is WLAN.
BT (1024)	Indicates that the signal personality is BT.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (signal_configuration_exists, personality, error_code)

`close()`

Closes the RFmx session.

Call this function a number of times equal to the number of times you obtained a reference to the RFmx session for a particular resource name.

> **Note**
>
> If you have used an existing instrument handle to create this session; calling the dispose, close, or force_close functions will only dispose the Python resources associated with this session. The pre-existing instrument handle will NOT be released.

`configure_external_attenuation_interpolation_linear`(*selector_string*, *table_name*, *format*)

Selects the linear interpolation method when interpolating S-parameters for the specified table. If the carrier frequency does not match a row in the S-parameter table, this method performs a linear interpolation based on the entries above and below the row in the table.

> **Note**
>
> Currently interpolation is supported only for S-parameter tables.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

> **Note**
>
> For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. Use *get_available_ports()* method to get the valid port names.

Example:

""

"calplane::plane0"

"calplane::plane0/port::if0"

"port::if0"

"calplane::plane0/port::all"

- **table_name** (*string*) – This parameter specifies the name to be associated with either the S-parameter table or the external attenuation table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

- **format** (enums.LinearInterpolationFormat, *int*) – This parameter specifies the format of parameters to interpolate. The default value is **Real and Imaginary**.

Name (Value)	Description
Real and Imaginary (0)	Results in a linear interpolation of the real portion of the complex number and a separate linear interpolation of the complex portion.
Magnitude and Phase (1)	Results in a linear interpolation.
Magnitude and Phase (dB) (2)	Results in a linear interpolation.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_external_attenuation_interpolation_nearest(*selector_string*, *table_name*)

Selects the nearest interpolation method when interpolating S-parameters for a specified table. The parameters of the table nearest to the carrier frequency are used.

> **Note**
>
> Currently interpolation is supported only for S-parameter tables.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  > **Note**
  >
  > For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

  "calplane::plane0/port::if0"

  "port::if0"

  "calplane::plane0/port::all"

- **table_name** (*string*) – This parameter specifies the name to be associated with either the S-parameter table or the external attenuation table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_external_attenuation_interpolation_spline(*selector_string*, *table_name*)

Selects the spline interpolation method when interpolating parameters for the specified table. If the carrier frequency does not match a row in the S-parameter table, this method performs a spline interpolation based on the entries above and below the row in the table.

> **Note**
>
> Currently interpolation is supported only for S-parameter tables.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  > **Note**
  >
  > For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

  "calplane::plane0/port::if0"

  "port::if0"

  "calplane::plane0/port::all"

- **table_name** (*string*) – This parameter specifies the name to be associated with either the S-parameter table or the external attenuation table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_external_attenuation_table(*selector_string*, *table_name*, *frequency*, *external_attenuation*)

Stores the external attenuation table in the calibration plane specified by the **Selector String** parameter. On a MIMO session, the external attenuation table is stored for each MIMO port in the specified calibration plane.

If there is only one table configured in any calibration plane, it is automatically selected as the active table.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which the external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  On a MIMO session if you do not specify the port name, this configuration is applied to all MIMO ports in the session for the default calibration plane instance. To configure external attenuation table for a specific MIMO port, use the port specifier with or without the calplane name.

  Example: "calplane::plane1/port::myrfsa1/0".

  Note

  For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". If you specify "port::all", all MIMO ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

  "calplane::plane0/port::if0"

  "port::if0"

  "calplane::plane0/port::all"

  "calplane::plane0/port::myrfsa1/0"

  "calplane::plane0/port::myrfsa1/0, port::myrfsa2/0"

  "calplane::plane0/port::myrfsa1/0/if0"

- **table_name** (*string*) – This parameter specifies the name to be associated with external attenuation table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

- **frequency** (*numpy.float64*) – This parameter specifies an array of frequencies in the external attenuation table. This value is expressed in Hz.

- **external_attenuation** (*numpy.float64*) – This parameter specifies an array of attenuations corresponding to the frequency specified by the **Frequency** parameter. This value is expressed in dB.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

configure_frequency_reference(*selector_string*, *frequency_reference_source*, *frequency_reference_frequency*)

Configures the Reference Clock and the frequency reference source.

Parameters

- **selector_string** (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.
- **frequency_reference_source** (*string*) – This parameter specifies the frequency reference source.

  The default value for PXIe-5840 with PXIe-5653 is **RefIn2**, else the default value is **OnboardClock**.

Name (Value	Description
Onboard-Clock (Onboard-Clock)	PXIe-5663/5663E: RFmx locks the PXIe-5663/5663E to the PXIe-5652 LO source onboard clock. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. On versions of the PXIe-5663/5663E that lack a REF OUT2 connector on the PXIe-5652, connect the REF IN/OUT connector on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: RFmx locks the PXIe-5665 to the PXIe-5653 LO source onboard clock. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN terminal. PXIe-5668: Lock the PXIe-5668 to the PXIe-5653 LO SOURCE onboard clock. Connect the LO2 OUT connector on the PXIe-5606 to the CLK IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: RFmx locks the device to its onboard clock. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. PXIe-5842: Lock to the PXIe-5655 onboard clock. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860:Lock to the PXIe-5860 onboard clock
RefIn (RefIn)	PXIe-5663/5663E: Connect the external signal to the PXIe-5652 REF IN/OUT connector. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: Connect the external signal to the PXIe-5653 REF IN connector. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN connector. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5668: Connect the external signal to the PXIe-5653 REF IN connector. Connect the LO2 OUT on the PXIe-5606 to the CLK IN connector on the PXIe-5622. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842: RFmx locks the device to the signal at the external REF IN connector. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. For PXIe-5830, lock the external signal to the PXIe-3621 REF IN connector. For PXIe-5831, lock the external signal to the PXIe-3622 REF IN connector. For PXIe-5832, lock the external signal to the PXIe-3623 REF IN connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. Lock the external signal to the

- **frequency_reference_frequency** (*float*) – This parameter specifies the Reference Clock rate when the **Frequency Reference Source** parameter is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

  The default value is 10 MHz.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_mechanical_attenuation(*selector_string*, *mechanical_attenuation_auto*, *mechanical_attenuation_value*)

Configures the mechanical attenuation and the RFmx driver attenuation hardware settings.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.
- **mechanical_attenuation_auto** (enums.MechanicalAttenuationAuto, *int*) – This parameter specifies whether RFmx automatically chooses an attenuation setting based on the hardware settings.

  The default value is **True**.

Name (Value)	Description
False (0)	Specifies that RFmx uses the value configured in the **Mechanical Attenuation** parameter.
True (1)	Specifies that the measurement computes the mechanical attenuation.

- **mechanical_attenuation_value** (*float*) – This parameter specifies the level of mechanical attenuation for the RF path. This value is expressed in dB.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_rf_attenuation(*selector_string*, *rf_attenuation_auto*, *rf_attenuation_value*)

Configures the nominal attenuation and the RFmx driver setting.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.
- **rf_attenuation_auto** (enums.RFAttenuationAuto, *int*) – This parameter specifies whether RFmx computes the RF attenuation.

  If you set this parameter to **True**, RFmx automatically chooses an attenuation setting based on the reference level configured on the personality.

The default value is **True**.

Name (Value)	Description
False (0)	Specifies that RFmx uses the value configured using **RF Attenuation** parameter.
True (1)	Specifies that RFmx computes the RF attenuation automatically.

- **rf_attenuation_value** (*float*) – This parameter specifies the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

  If you set the **RF Attenuation Auto** parameter to **True**, RFmx chooses an attenuation setting automatically.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_s_parameter_external_attenuation_table(*selector_string*, *table_name*, *frequency*, *s_parameters*, *s_parameter_orientation*)

Stores the S-parameter table in the calibration plane specified by the **Selector String** parameter. On a MIMO session, the S-parameter table is stored for each MIMO port in the specified calibration plane.

Supported devices: PXIe-5830/5831/5832/5840/5841/5860

Note

If there is only one table configured in any calibration plane, it is automatically selected as the active table.

Parameters

- **selector_string** (*string*) – Specifies the calibration plane name in which the external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used. The default value is "" (empty string).

  On a MIMO session if you do not specify the port name, this configuration is applied to all MIMO ports in the session for the default calibration plane instance. To configure S-parameter external attenuation table for a specific MIMO port, use the port specifier with or without the calplane name. Example: "calplane::plane1/port::myrfsa1/0".

  Note

  For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". If you specify "port::all", all MIMO ports are considered configured.

Example:

""

"calplane::plane0"

"calplane::plane0/port::if0"

"port::if0"

"calplane::plane0/port::all"

"calplane::plane0/port::myrfsa1/0"

"calplane::plane0/port::myrfsa1/0, port::myrfsa2/0"

"calplane::plane0/port::myrfsa1/0/if0"

- **table_name** (*string*) – Specifies the name to be associated with S-parameter table within a calibration plane. Provide a unique name, such as "table1" to configure the table.
- **frequency** (*numpy.float64*) – Specifies an array of frequencies in the S-parameter table. This value is expressed in Hz.
- **s_parameters** (*numpy.complex64*) – Specifies the S-parameters for each frequency. The first index indicates the corresponding frequency entry, the second index corresponds to the target port for the S-parameter, and the third index corresponds to the the source port. For example, to index the s21 parameter for the fourth frequency in the table, you would use {3, 1, 0} as the indexes since they are zero-based.
- **s_parameter_orientation** (enums.SParameterOrientation, *int*) – Specifies the orientation of the data in the S-parameter table relative to the port you specify. The default value is **Port2 Towards DUT**.

Name (Value)	Description
Port1 Towards DUT (0)	Port 1 of the S2P is oriented towards the DUT.
Port2 Towards DUT (1)	Port 2 of the S2P is oriented towards the DUT.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

configure_s_parameter_external_attenuation_type(*selector_string*, *s_parameter_type*)

Configures the type of S-parameter to apply to measurements on the specified port for a Calplane. You can use the Selector String input to specify the name of the Calplane and port to configure for S-parameter.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  On a MIMO session if you do not specify the port name, this configuration is applied to all MIMO ports in the session for the default calibration plane instance. To configure S-

parameter external attenuation type for a specific MIMO port, use the port specifier with or without the calplane name.

Example: “calplane::plane1/port::myrfsa1/0”.

> **Note**
>
> For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is “calplane::<calplaneName>/port::<portName>”. If you specify “port::all”, all ports are considered configured. Use *get_available_ports()* method to get the valid port names. For a MIMO port, the valid selector string is “calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>”. If you specify “port::all”, all MIMO ports are considered configured.

Example:

””

”calplane::plane0”

”calplane::plane0/port::if0”

”port::if0”

”calplane::plane0/port::all”

”calplane::plane0/port::myrfsa1/0”

”calplane::plane0/port::myrfsa1/0, port::myrfsa2/0”

”calplane::plane0/port::myrfsa1/0/if0”

- **s_parameter_type** (enums.SParameterType, *int*) – This parameter specifies the type of S-parameter which applies to measurements on the specified port for a Calplane. If you set this parameter to **Scalar** or **Vector**, RFmx adjusts the instrument settings and the returned data to remove the effects of the external network between the instrument and the DUT.

  PXIe-5831/5832: Valid values for this parameter are **Scalar** and **Vector**. **Vector** is only supported for TRX ports in a semiconductor test system (STS).

  PXIe-5840/5841/5842/5860: The only valid value for this parameter is **Scalar**.

  The default value is **Scalar**.

Name (Value)	Description
Scalar (1)	De-embeds the measurement using the gain term.
Vector (2)	De-embeds the measurement using the gain term and the reflection term.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

delete_all_external_attenuation_tables(*selector_string*)

Deletes all the external attenuation tables in the calibration plane specified by the **Selector String** parameter. On a MIMO session, this method deletes all the external attenuation tables for the specified MIMO port.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used. If you specify "calplane::all", all the calibration planes are deleted.

On a MIMO session, the default "" (empty string) deletes all the external attenuation tables for all MIMO Ports. To delete an external attenuation type for a specific MIMO port, use the port specifier with or without the calplane name.

Example: "calplane::plane1/port::myrfsa1/0".

> **Note**
>
> For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>".
> If you specify "port::all", all MIMO ports are considered configured. Use *get_available_ports()* method to get the valid port names.

Example:

""

"calplane::plane0"

"calplane::all"

"calplane::plane0/port::if0"

"port::if0"

"calplane::plane0/port::all"

"calplane::all/port::all"

"calplane::plane0/port::myrfsa1/0"

"calplane::plane0/port::myrfsa1/0, port::myrfsa2/0"

"calplane::plane0/port::myrfsa1/0/if0"

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

delete_external_attenuation_table(*selector_string*, *table_name*)

Deletes the external attenuation table set by the **Table Name** parameter in the calibration plane specified by the **Selector String** parameter. On a MIMO session, this method deletes the external attenuation table for the specified MIMO port.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which the external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  On a MIMO session, the default "" (empty string) deletes the active external attenuation table for all the MIMO Ports. To delete an external attenuation type for a specific MIMO port, use the port specifier with or without the calplane name.

  Example: "calplane::plane1/port::myrfsa1/0".

  > **Note**
  >
  > For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". If you specify "port::all", all MIMO ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

  "calplane::plane0/port::if0"

  "port::if0"

  "calplane::plane0/port::all"

  "calplane::plane0/port::myrfsa1/0"

  "calplane::plane0/port::myrfsa1/0, port::myrfsa2/0"

  "calplane::plane0/port::myrfsa1/0/if0"

- **table_name** (*string*) – This parameter specifies the name to be associated with external attenuation table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`disable_calibration_plane`(*selector_string*)

Disables the calibration plane specified by the **Selector String** parameter for amplitude correction.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – This parameter specifies the calibration plane name in which the external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used. If you specify "calplane::all", all the calibration planes are disabled.

Example:

""

"calplane::plane0"

"calplane::all"

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`dispose`()

Closes the RFmx session.

Call this function a number of times equal to the number of times you obtained a reference to the RFmx session for a particular resource name.

Note

You can call this function safely more than once, even if the session is already closed.

If you have used an existing instrument handle to create this session; calling the dispose, close, or force_close functions will only dispose the Python resources associated with this session. The pre-existing instrument handle will NOT be released.

`enable_calibration_plane`(*selector_string*)

Enables the calibration plane specified by the **Selector String** parameter for amplitude correction.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – This parameter specifies the calibration plane name in which the external attenuation table or S-parameter is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used. If "calplane::all" is specified, all the calibration planes are enabled.

Example:

""

”calplane::plane0”

”calplane::all”

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

export_signal(*export_signal_source*, *export_signal_output_terminal*)

Routes signals (triggers, clocks, and events) to the specified output terminal.

> **Note**
>
> This method is not supported on a MIMO session.

Parameters

- **export_signal_source** (enums.ExportSignalSource, *int*) – This parameter controls the source to export signals.

Name (Value)	Description
Start Trigger (0)	Start trigger is sourced.
Ref Trigger (1)	Reference trigger is sourced.
Advance Trigger (2)	Advance trigger is sourced.
Ready for Start Event (3)	Ready for Start event is sourced.
Ready for Advance Event (4)	Ready for Advance event is sourced.
Ready for Ref Event (5)	Ready for Reference event is sourced.
End of Record Event (6)	End of Record event is sourced.
Done Event (7)	Done event is sourced.
Reference Clock (8)	Reference clock is sourced.

- **export_signal_output_terminal** (*string*) – This parameter specifies the terminal where the signal is exported. You can also choose not to export any signal.

  The default value is “” (empty string).

Name (Value)	Description
Do not export signal (())	The signal is not exported.
ClkOut (ClkOut)	Export the Reference Clock on the CLK OUT terminal on the digitizer. This value is not valid for the PXIe-5644/5645/5646 or PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut (RefOut)	The signal is exported to the REF IN/OUT terminal on the PXIe-5652, the REF OUT terminals on the PXIe-5653, or the REF OUT terminal on the PXIe-5644/5645/5646, or PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	The signal is exported to the REF OUT2 terminal on the LO. This connector exists only on some versions of the PXIe-5652.
PFI0 (PFI0)	The signal is exported to the PFI 0 connector on the PXIe-5142 and PXIe-5624.
PFI1 (PFI1)	The signal is exported to the PFI 1 connector on the PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0	The signal is exported to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1	The signal is exported to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2	The signal is exported to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3	The signal is exported to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4	The signal is exported to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5	The signal is exported to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6	The signal is exported to the PXI trigger line 6.
PXIe_DSta (PXIe_DSt	The signal is exported to the PXI DStar C trigger line. This value is valid only on PXIe-5820/5830/5831/5832/5840/5841/5842/5860.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

`fetch_raw_iq_data`(*selector_string*, *timeout*, *records_to_fetch*, *samples_to_read*, *data*)

Fetches I/Q data from a single record in an acquisition.

Parameters

- **selector_string** (*string*) – Specifies a selector string. Pass an empty string.
- **timeout** (*float*) – This parameter specifies the timeout, in seconds, for fetching the raw IQ data. A value of -1 specifies that the function waits until all data is available. A value of 0 specifies the function immediately returns available data. The default value is 10.

- **records_to_fetch** (*int*) – This parameter specifies the record to retrieve. Record numbers are zero-based. The default value is 0.
- **samples_to_read** (*int*) – This parameter specifies the number of samples to fetch. A value of -1 specifies that RFmx fetches all samples. The default value is -1.

Returns
Returns the status code of this method. The status code either indicates success or describes an error or warning condition.

Return type
int

`force_close()`

Closes all RFmx sessions.

Calling this method once will destroy the session, irrespective of the many references obtained for the session for a particular resource name.

> **Note**
>
> If you have used an existing instrument handle to create this session; calling the dispose, close, or force_close functions will only dispose the Python resources associated with this session. The pre-existing instrument handle will NOT be released.

`get_advance_trigger_digital_edge_source(selector_string)`

Gets the source terminal for the advance trigger.

This attribute is used only when the *ADVANCE_TRIGGER_TYPE* attribute is set to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):

Specifies the source terminal for the advance trigger.

error_code (int):

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_advance_trigger_export_output_terminal`(*selector_string*)

Gets the destination terminal for the exported advance trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the exported advance trigger.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_advance_trigger_terminal_name`(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/AdvanceTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/AdvanceTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/AdvanceTrigger,*, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/AdvanceTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_advance_trigger_type`(*selector_string*)

Gets whether the advance trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No advance trigger is configured.
Digital Edge (1)	The advance trigger is not asserted until a digital edge is detected. The source of the digital edge is specified with the Advance Trigger Digital Edge Source attribute.
Software (3)	The advance trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.AdvanceTriggerType):
Specifies whether the advance trigger is a digital edge or a software trigger.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_amplitude_settling(*selector_string*)

Gets the amplitude settling accuracy value. This value is expressed in decibels. RFmx waits until the RF power attains the specified accuracy level after calling the RFmx Initiate method.

Any specified amplitude settling value that is above the acceptable minimum value is coerced down to the closest valid value.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported Devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the amplitude settling accuracy value. This value is expressed in decibels. RFmx waits until the RF power attains the specified accuracy level after calling the RFmx Initiate method.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_available_paths(*selector_string*)

Fetches the list of paths available for use based on your instrument configuration. On a MIMO session, this method fetches all the paths for the initialized MIMO paths.

Parameters
selector_string (*string*) – Specifies a selector string. Pass an empty string.

Returns

available_paths (string):
This parameter returns a list of available paths.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (available_paths, error_code)

get_available_ports(*selector_string*)

Fetches the list of ports available for use based on your instrument configuration. On a MIMO session, this method fetches all the ports for the initialized MIMO ports.

Parameters
selector_string (*string*) – Specifies a selector string. Pass an empty string.

Returns

available_ports (string):
This parameter returns a list of available ports.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (available_ports, error_code)

get_bluetooth_signal_configuration(*signal_name=''*)

Creates a Bluetooth signal configuration for specified signal name. Existing Bluetooth signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type Bluetooth.

Return type
bluetooth (*BT*)

get_channel_coupling(*selector_string*)

Gets whether the RF IN connector is AC- or DC-coupled on the downconverter.

Nam (valu	Description
—	NoteFor the PXIe-5665/5668, this attribute must be set to AC Coupled when the DC block is present, and set to DC Coupled when the DC block is not present to ensure device specifications are met and proper calibration data is used. For more information about removing or attaching the DC block, refer to the PXIe-5665 Theory of Operation or the PXIe-5668 Theory of Operation topics in the NI RF Vector Signal Analyzers Help.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **AC Coupled**.

Valid values

- PXIe-5665 (3.6 GHz): AC Coupled DC Coupled
- PXIe-5665 (14 GHz): AC Coupled, DC Coupled
- PXIe-5668: AC Coupled

Supported devices: PXIe-5665/5668

Name (Value)	Description
AC Coupled (0)	Specifies that the RF input channel is AC-coupled. For low frequencies (<10 MHz), accuracy decreases because RFmxInstr does not calibrate the configuration.
DC Coupled (1)	Specifies that the RF input channel is DC-coupled. The RFmx driver enforces a minimum RF attenuation for device protection.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.ChannelCoupling):
Specifies whether the RF IN connector is AC- or DC-coupled on the downconverter.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_cleaner_spectrum(*selector_string*)

Gets how to obtain the lowest noise floor or faster measurement speed.

Name (value)	Description
PXIe-5665	Sets the FFT Width attribute to take narrower bandwidth acquisitions and avoid digitizer spurs. Uses IF filters to reduce the noise floor for frequencies below 80 MHz.
PXIe-5644/5645/5646, PXIe-5840/5841/5842	Returns the best possible spectrum.
PXIe-5668	Returns the best possible spectrum. To provide the best spectrum measurement, the acquisition is reduced to 100 MHz segments for any center frequency.
Other devices	This attribute is ignored.

Note

Some measurements, such as Spurious Emissions enable the Cleaner Spectrum attribute by default. You can speed up those measurements by disabling the Cleaner Spectrum attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disable this attribute to get faster measurement speed.
Enabled (1)	Enable this attribute to get the lowest noise floor and avoid digitizer spurs.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.CleanerSpectrum):
Specifies how to obtain the lowest noise floor or faster measurement speed.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_common_mode_level(*selector_string*)

Gets the common-mode level presented at each differential input terminal. The common-mode level shifts both positive and negative terminals in the same direction. This must match the common-mode level of the device under test (DUT). This value is expressed in Volts.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5820

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the common-mode level presented at each differential input terminal. The common-mode level shifts both positive and negative terminals in the same direction. This must match the common-mode level of the device under test (DUT). This value is expressed in Volts.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_demod_signal_configuration(*signal_name=''*)

Creates a Demod signal configuration for specified signal name. Existing Demod signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type Demod.

Return type
demod (Demod)

get_device_temperature(*selector_string*)

Gets the current temperature of the module. This value is expressed in degrees Celsius.

To use this attribute for PXIe-5830/5831/5832, you must first use the Selector String attribute to specify the name of the channel you are configuring. When you are reading the device temperature, you must specify the context in the Selector String input as "module::<ModuleName>". You can also use the *build_module_string()* method to build the module string. For all other devices, the only valid value for the selector string is "" (empty string).

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the selector string to read this attribute. You can use the *build_port_string()* method to build the selector string. For PXIe-5830/5831/5832, you must specify the context in the selector string input as port::<deviceName>/<channelNumber>/module::<moduleName>.

Refer to the following table to determine which strings are valid for your configuration.

Hardware Module	TRX Port Type	Selector String
PXIe-3621/3622/3623	•	if or “” (empty string)
PXIe-5820	•	fpga
First connected mmRH-5582	DIRECT TRX PORTS Only	rf0
First connected mmRH-5582	SWITCHED TRX PORTS [0-7]	rf0switch0
First connected mmRH-5582	SWITCHED TRX PORTS [8-15]	rf0switch1
Second connected mmRH-5582	DIRECT TRX PORTS Only	rf1
Second connected mmRH-5582	SWITCHED TRX PORTS [0-7]	rf1switch0
Second connected mmRH-5582	SWITCHED TRX PORTS [8-15]	rf1switch1

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the current temperature of the module. This value is expressed in degrees Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_digital_gain(*selector_string*)

Gets the scaling factor applied to the time-domain voltage data in the digitizer. This value is expressed in dB. RFmx does not compensate for the specified digital gain.

You can use this attribute to account for external gain changes without changing the analog signal path.

Note

The PXIe-5644/5645/5646 applies this gain when the data is scaled. The raw data does not include this scaling on these devices.

Default Value : 0 dB

Supported Devices : PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the scaling factor applied to the time-domain voltage data in the digitizer. This value is expressed in dB. RFmx does not compensate for the specified digital gain.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_digitizer_dither_enabled`(*selector_string*)

Gets whether dithering is enabled on the digitizer.

Dithering adds band-limited noise in the analog signal path to help reduce the quantization effects of the ADC and improve spectral performance. On the PXIe-5622, this out-of-band noise is added at low frequencies of up to approximately 12 MHz.

PXIe-5663/5663E/5665: When you enable dithering, the maximum signal level is reduced by up to 3 dB. This signal level reduction is accounted for in the nominal input ranges of the PXIe-5622. Therefore, you can overrange the input by up to 3 dB with dither disabled. For example, the +4 dBm input range can handle signal levels up to +7 dBm with dither disabled.

For wider bandwidth acquisitions, such as 40 MHz, disable dithering to eliminate residual leakage of the dither signal into the lower frequencies of the IF passband, which starts at 12.5 MHz and ends at 62.5 MHz. This leakage can slightly raise the noise floor in the lower frequencies, thus degrading the performance in high-sensitivity applications. When performing spectral measurements, this leakage can also appear as a wide, low-amplitude signal near the 12.5 MHz and 62.5 MHz frequencies. The width and amplitude of the signal depends on your resolution bandwidth and the type of time-domain window you apply to your FFT.

PXIe-5668: When you enable dithering, the maximum signal level is reduced by up to 2 dB. For the PXIe-5624, the maximum input power with dither off is 8 dBm and the maximum input power level with dither on is 6 dBm. When acquiring an 800 MHz bandwidth signal, the I/Q data contains the dither even if the dither signal is not in the displayed spectrum. The dither can affect actions like power level triggering.

Name (value)	Description
—	Note For the PXIe-5668, disabling dithering can negatively affect absolute amplitude accuracy.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Note

For PXIe-5820/5830/5831/5832/5840/5841/5842, only **Enabled** is supported.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.DigitizerDitherEnabled):
Specifies whether dithering is enabled on the digitizer.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_digitizer_temperature`(*selector_string*)

Gets the current temperature of the digitizer module. This value is expressed in degrees Celsius.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

> **Note**
> This attribute is not supported if you are using an external digitizer.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the current temperature of the digitizer module. This value is expressed in degrees Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_done_event_output_terminal`(*selector_string*)

Gets the destination terminal for the Done event.

> **Note**
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the Done event.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_done_event_terminal_name`(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/DoneEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/DoneEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/DoneEvent*, where *ModuleName* is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/DoneEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_downconverter_center_frequency`(*selector_string*)

Enables in-band retuning and specifies the current frequency of the RF downconverter. This value is expressed in Hz.

After you set this attribute, the RF downconverter is locked to that frequency until the value is changed or the attribute is reset. Locking the downconverter to a fixed value allows frequencies within the instantaneous bandwidth of the downconverter to be measured without the overhead of retuning the LO and waiting for the LO to settle. This method is called in-band retuning and it has the highest benefit on analyzers that have

larger LO settling times. After setting the downconverter center frequency, you can set the center frequency to the frequencies at which you want to take the measurements.

If you want to avoid the LO leakage or DC offset of analyzers that use a direct conversion architecture, it is more convenient to use the *DOWNCONVERTER_FREQUENCY_OFFSET* or *LO_LEAKAGE_AVOIDANCE_ENABLED* attributes.

If you set this attribute, any measurements outside the instantaneous bandwidth of the device are invalid. To disable in-band retuning, reset this attribute or call the *reset_to_default()* method.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is the carrier frequency or spectrum center frequency.

Valid Values: Any supported tuning frequency of the device.

> **Note**
>
> PXIe-5820: The only valid value for this attribute is 0 Hz.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Enables in-band retuning and specifies the current frequency of the RF downconverter. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_downconverter_frequency_offset(*selector_string*)

Gets an offset from the center frequency value for the downconverter. Use this attribute to offset the measurement away from the LO leakage or DC Offset of analyzers that use a direct conversion architecture. You must set this attribute to half the bandwidth or span of the measurement + guardband. The guardband is needed to ensure that the LO leakage is not inside the analog or digital filter rolloffs. This value is expressed in Hz.

NI recommends using the *LO_LEAKAGE_AVOIDANCE_ENABLED* attribute instead of the Downconverter Frequency Offset attribute. The LO Leakage Avoidance Enabled attribute automatically configures the Downconverter Frequency Offset attribute to an appropriate offset based on the bandwidth or span of the measurement.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values: For spectrum acquisition types, the RFmx driver automatically calculates the default value to avoid residual LO power. For I/Q acquisition types, the default value is 0 Hz. If the center frequency is set to a non-multiple of *LO_FREQUENCY_STEP_SIZE* attribute, this attribute is set to compensate for the difference.

The following valid values correspond to their respective devices:

Name (value)	Description
PXIe-5646	-100 MHz to +100 MHz
PXIe-5830/5831/5832/5840	-500 MHz to +500 MHz
PXIe-5841 (200MHz Bandwidth)	-100 MHz to +100 MHz
PXIe-5841 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (500MHz Bandwidth)	-250 MHz to +250 MHz
PXIe-5842 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (2GHz Bandwidth)	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using Standard personality	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using the 4GHz Bandwidth personality	-2 GHz to +2 GHz
Other devices	-42 MHz to +42 MHz

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies an offset from the center frequency value for the downconverter. Use this attribute to offset the measurement away from the LO leakage or DC Offset of analyzers that use a direct conversion architecture. You must set this attribute to half the bandwidth or span of the measurement + guardband. The guardband is needed to ensure that the LO leakage is not inside the analog or digital filter rolloffs. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_downconverter_gain(*selector_string*)

Gets the net signal gain for the device at the current RFmx settings and temperature. RFmx scales the acquired I/Q and spectrum data from the digitizer using the value of this attribute.

For a vector signal analyzer (VSA), the system is defined as the RF downconverter for all interfaces between the RF IN connector on the RF downconverter front panel and the IF IN connector on the digitizer front panel. For a spectrum monitoring receiver, the system is defined as the RF preselector, RF downconverter,

and IF conditioning modules including all interfaces between the RF IN connector on the RF preselector module front panel and the IF IN connector on the digitizer front panel.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is N/A.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the net signal gain for the device at the current RFmx settings and temperature. RFmx scales the acquired I/Q and spectrum data from the digitizer using the value of this attribute.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_downconverter_preselector_enabled(*selector_string*)

Gets whether the tunable preselector is enabled on the downconverter.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the preselector.
Enabled (1)	The preselector is automatically enabled when it is in the signal path and is automatically disabled when it is not in the signal path. Use the Preselector Present attribute to determine if the downconverter has a preselector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.DownconverterPreselectorEnabled):
Specifies whether the tunable preselector is enabled on the downconverter.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_end_of_record_event_output_terminal`(*selector_string*)

Gets the destination terminal for the End of Record event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the End of Record event.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_end_of_record_event_terminal_name`(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/EndOfRecordEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/EndOfRecordEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/EndOfRecordEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/EndOfRecordEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_error_string`(*error_code*)

Gets the description of a driver error code.

Parameters
error_code (*int*) – Specifies an error or warning code.

Returns
Contains the error description.

Return type
string

`get_external_attenuation_table_actual_value`(*selector_string*)

Returns the external attenuation table actual value that is applied to the measurements for a specified signal and calibration plane.

On a MIMO session, this method returns the external attenuation table actual value for a specified port. You can use the **Selector String** parameter to specify the name of the signal, calibration plane, and MIMO port to return the external attenuation table actual value.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

`selector_string` (*string*) – This parameter specifies a Selector String comprising of calibration plane name. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used. On a MIMO session, you must use "port::<deviceName>/<channelNumber>" as part of the selector string to read the external attenuation table actual value for the specified port. If you do not specify the signal name, the value is returned for the last committed signal instance.

Example:

""

"signal::sig1"

"calplane::plane0"

"signal::sig1/calplane::plane0"

"port::rfsa1/0"

"signal::sig1/port::rfsa1/0"

"calplane::plane0/port::rfsa1/0"

"signal::sig1/calplane::plane0/port::rfsa1/0"

Returns

external_attenuation (float):
This parameter returns the external attenuation table actual value applied to the measurements for a specified signal and calibration plane. This further includes interpolation of the external attenuation table based on the specified signal. On a MIMO session, this value corresponds to a specified port. This value is expressed in dB.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (external_attenuation, error_code)

`get_fft_width`(*selector_string*)

Gets the FFT width of the device. The FFT width is the effective bandwidth of the signal path during each signal acquisition.

The lower limit for all devices that support setting the FFT Width attribute is 7.325 kHz.

PXIe-5663/5663E: The FFT width upper limit for the PXIe-5663/5663E depends on the RF frequency and on the module revision of the PXIe-5601. For more information about determining which revision of the

PXIe-5601 RF downconverter you have installed, refer to the Identifying Module Revision topic in the *NI RF Vector Signal Analyzers Help*.

> **Note**
>
> The maximum FFT width for your device is constrained to 50 MHz or 25 MHz, depending on the digitizer option you purchased.

> **Note**
>
> You can use the FFT Width attribute with in-band retuning. For more information about in-band retuning, refer to the *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

The RFmx driver treats the device instantaneous bandwidth as the effective real-time bandwidth of the signal path. The span specifies the frequency range of the computed spectrum. A signal analyzer can acquire a bandwidth only within the device instantaneous bandwidth. If the span you choose is greater than the device instantaneous bandwidth, the RFmx driver obtains multiple acquisitions and combines them into a single spectrum. By specifying the FFT width, you can control the specific bandwidth obtained in each signal acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the FFT width of the device. The FFT width is the effective bandwidth of the signal path during each signal acquisition.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_frequency_reference_exported_terminal(*selector_string*)

Gets a comma-separated list of the terminals at which to export the frequency reference.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None ()	The Reference Clock is not exported. This value is not valid for the PXIe-5644/5645/5646.
RefOut (RefOut)	Export the clock on the REF IN/OUT terminal on the PXIe-5652, the REF OUT terminals on the PXIe-5653, or the REF OUT terminal on the PXIe-5694, PXIe-5644/5645/5646, or PXIe-5820/5830/5831/5832/5840/5841/5860.
RefOut2 (RefOut2)	Export the clock on the REF OUT2 terminal on the PXIe-5652. This value is valid only for the PXIe-5663E.
ClkOut (ClkOut)	Export the Reference Clock on the CLK OUT terminal on the Digitizer. This value is not valid for the PXIe-5644/5645/5646 or PXIe-5820/5830/5831/5832/5840/5841/5842/5860.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies a comma-separated list of the terminals at which to export the frequency reference.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_frequency_reference_frequency(*selector_string*)

Gets the Reference Clock rate, when the *FREQUENCY_REFERENCE_SOURCE* attribute is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 10 MHz.

Valid values

Name (value)	Description
PXIe-5644/5645/5646, PXIe-5663/5663E, PXIe-5820/5830/5831/5832/5840/5841/5842	10 MHz
PXIe-5665/5668	5 MHz to 100 MHz (inclusive), in increments of 1 MHz
PXIe-5860	10 MHz, 100 MHz

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the Reference Clock rate, when the *FREQUENCY_REFERENCE_SOURCE* attribute is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_frequency_reference_source(*selector_string*)

Gets the frequency reference source.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

All other devices default value is **OnboardClock**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value	Description
Onboard-Clock (Onboard-Clock)	PXIe-5663/5663E: The RFmx driver locks the PXIe-5663/5663E to the PXIe-5652 LO source onboard clock. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. On versions of the PXIe-5663/5663E that lack a REF OUT2 connector on the PXIe-5652, connect the REF IN/OUT connector on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: The RFmx driver locks the PXIe-5665 to the PXIe-5653 LO source onboard clock. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN terminal. PXIe-5668: Lock the PXIe-5668 to the PXIe-5653 LO SOURCE onboard clock. Connect the LO2 OUT connector on the PXIe-5606 to the CLK IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to its onboard clock. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. PXIe-5842: Lock to the associated PXIe-5655 onboard clock. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860: Lock to the PXIe-5860 onboard clock.
RefIn (RefIn)	PXIe-5663/5663E: Connect the external signal to the PXIe-5652 REF IN/OUT connector. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: Connect the external signal to the PXIe-5653 REF IN connector. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN connector. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5668: Connect the external signal to the PXIe-5653 REF IN connector. Connect the LO2 OUT on the PXIe-5606 to the CLK IN connector on the PXIe-5622. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to the signal at the external REF IN connector. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. For PXIe-5830, lock the external signal to the PXIe-3621 REF IN connector. For PXIe-5831, lock the external signal to the PXIe-3622 REF IN connector. For PXIe-5832, lock the external signal to the PXIe-3623 REF IN connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5842: Lock to the signal at the REF IN connector on the associated PXIe-5655. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860:Lock to the signal at the REF IN connector on the PXIe-5860.
PXI_C (PXI_C	PXIe-5668: Lock the PXIe-5653 to the PXI backplane clock. Connect the PXIe-5606 LO2 OUT to the LO2 IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5663/5663E/5665, and PXIe-5820/5840/5841/5860: The RFmx driver locks the device to the PXI backplane clock. PXIe-5830/5831/5832 with PXIe-5653/5841 with PXIe-5655, PXIe-5842/5860: The RFmx

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the frequency reference source.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_frequency_settling(*selector_string*)

Gets the value used for LO frequency settling.

Specify the units and interpretation for this scalar value using the *FREQUENCY_SETTLING_UNITS* attribute.

Valid values

Frequency Settling Units Property Value	PXIe-5663/566	PXIe-5665/566	PXIe-5644/5	PXIe-5830/5831/5832/5840/5841/5 PXIe-5831 with PXIe-5653 (using PXIe-3622 LO), PXIe-5832 with PXIe-5653 (using PXIe-3623 LO)	PXIe-5831 with PXIe-5653 (using PXIe-5653 LO) and PXIe-5832 with PXIe-5653 (using PXIe-5653 LO)
Seconds After Lock	2 µs to 80 ms, resolution of approximately 2 µs	4 µs to 80 ms, resolution of approximately 4 µs	1 µs to 65 ms, resolution of 1 µs	1 µs to 10s, resolution of 1 µs	4 µs to 80 ms, resolution of approximately 4 µs
Seconds After I/O	0 µs to 80 ms, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs	1 µs to 65 ms, resolution of 1 µs	0 µs to 10s, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs
PPM	1.0, 0.1, 0.01	1.0, 0.1, 0.01, 0.001	1.0, 0.1, 0.01	1.0 to 0.01	1.0 to 0.01

Name (value)	Description
—	Note This attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.1.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the value used for LO frequency settling.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_frequency_settling_units(*selector_string*)

Gets the delay duration units and interpretation for LO settling.

Specify the actual settling value using the *FREQUENCY_SETTLING* attribute.

Name (value)	Description
—	Note The Frequency Settling Units attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **PPM**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
PPM (0)	Specifies the frequency settling in parts per million (ppm).
Seconds After Lock (1)	Specifies the frequency settling in time after lock (seconds).
Seconds After I/O (2)	Specifies the frequency settling in time after I/O (seconds).

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.FrequencySettlingUnits):
Specifies the delay duration units and interpretation for LO settling.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_if_filter_bandwidth(*selector_string*)

Gets the IF filter path bandwidth for your device configuration.

Name (value)	Description
—	Note For composite devices, such as the PXIe-5665/5668, the IF filter path bandwidth includes all IF filters across the component modules of a composite device.

RFmx chooses an appropriate IF filter as default IF Filter based on measurement configuration, center frequency, cleaner spectrum and downconverter preselector.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5665/5668

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the IF filter path bandwidth for your device configuration.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_if_output_power_level_offset(*selector_string*)

Gets the power offset by which to adjust the default IF output power level. This value is expressed in dB.

This attribute does not depend on absolute IF output power levels; therefore, you can use this attribute to adjust the IF output power level on all RFmx-supported devices without knowing the exact default value. Use this attribute to increase or decrease the nominal output level to achieve better measurement results.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the power offset by which to adjust the default IF output power level. This value is expressed in dB.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_input_isolation_enabled`(*selector_string*)

Gets whether input isolation is enabled.

Enabling this attribute isolates the input signal at the RF IN connector on the RF downconverter from the rest of the RF downconverter signal path. Disabling this attribute reintegrates the input signal into the RF downconverter signal path.

> **Note**
>
> If you enable input isolation for your device, the device impedance is changed from the characteristic 50-ohm impedance. A change in the device impedance may increase the VSWR value higher than the device specifications.

For PXIe-5830/5831/5832, input isolation is supported for all available ports for your hardware configuration.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.InputIsolationEnabled):
Specifies whether input isolation is enabled.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_instrument_firmware_revision`(*selector_string*)

Gets a string containing the firmware revision information of the RF downconverter for the composite device you are currently using.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns a string containing the firmware revision information of the RF downconverter for the composite device you are currently using.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_instrument_model(*selector_string*)

Gets a string that contains the model number or name of the RF device that you are currently using.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns a string that contains the model number or name of the RF device that you are currently using.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_lo2_export_enabled(*selector_string*)

Gets whether to enable the LO2 OUT terminals in the installed devices.

Set this attribute to **Enabled** to export the 4 GHz LO signal from the LO2 IN terminal to the LO2 OUT terminal. You can also export the LO2 signal by setting the *LO_EXPORT_ENABLED* attribute to TRUE.

> **Note**
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported Devices: PXIe-5665/5668

Name (Value)	Description
Disabled (0)	Disables the LO2 OUT terminals.
Enabled (1)	Enables the LO2 OUT terminals.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LO2ExportEnabled):
Specifies whether to enable the LO2 OUT terminals in the installed devices.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_lo_export_enabled(*selector_string*)

Gets whether to enable the LO OUT terminals on the installed devices.

Name (value)	Description
TRUE	Enables the LO OUT terminals.
FALSE	Disables the LO OUT terminals

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value:

- PXIe-5663/5663E: TRUE
- PXIe-5644/5645/5646, PXIe-5665/5668, PXIe-5830/5831/5832/5840/5841/5842: FALSE

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (bool):
Specifies whether to enable the LO OUT terminals on the installed devices.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_frequency`(*selector_string*)

Gets the LO signal frequency for the configured center frequency. This value is expressed in Hz.

If you are using the vector signal analyzer with an external LO, use this attribute to specify the LO frequency that the external LO source passes into the LO IN or LO1 IN connector on the RF downconverter front panel. If you are using an external LO, reading the value of this attribute after configuring the rest of the parameters returns the LO frequency needed by the device.

You can set this attribute to the actual LO frequency because RFmx corrects for any difference between expected and actual LO frequencies.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector Strings topic for information about the string syntax.

The default value is 0.

Supported Devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the LO signal frequency for the configured center frequency. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_frequency_step_size`(*selector_string*)

Gets the step size for tuning the LO phase-locked loop (PLL).

You can only tune the LO frequency in multiples of the LO Frequency Step Size attribute. Therefore, the LO frequency can be offset from the requested center frequency by as much as half of the LO Frequency Step Size attribute. This offset is corrected by digitally frequency shifting the LO frequency to the value requested in *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if PXIe-5653 is used as the LO source.

The valid values for this attribute depend on the *LO_PLL_FRACTIONAL_MODE* attribute.

PXIe-5644/5645/5646: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value.

PXIe-5840: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value that is less than or equal to the desired step size.

LO PLL Fractional Mode Enabled Property Setting	LO Frequency Step Size Property Valid Values on PXIe-5644/5645	LO Frequency Step Size Property Valid Values on PXIe-5646	LO Frequency Step Size Property Valid Values on PXIe-5840/5841	LO Frequency Step Size Property Valid Values on PXIe-5830/5831/583	LO Frequency Step Size Property Valid Values on PXIe-5841 with PXIe-5655, PXIe-5842*
Enabled	50 kHz to 24 MHz	50 kHz to 25 MHz	50 kHz to 100 MHz	LO1: 8 Hz to 400 MHz LO2: 4 kHz to 400 MHz	1 nHz to 50 MHz
Disabled	4 MHz, 5 MHz, 6 MHz, 12 MHz, 24 MHz	2 MHz, 5 MHz, 10 MHz, 25 MHz	1 MHz, 5 MHz, 10 MHz, 25 MHz, 50 MHz, 100 MHz	LO1: – LO2: –	1 nHz to 50 MHz

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5644/5645/5646	200 kHz
PXIe-5830	2 MHz
PXIe-5831/5832 (RF port)	8 MHz
PXIe-5831/5832 (IF port)	2 MHz, 4 MHz
PXIe-5840/5841	500 kHz
PXIe-5842	1 Hz

> **Note**
>
> The default value for PXIe-5831/5832 depends on the frequency range of the selected port for your instrument configuration. Use *get_available_ports()* method to get the valid port names.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the step size for tuning the LO phase-locked loop (PLL).

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_lo_in_power(*selector_string*)

Gets the power level expected at the LO IN terminal when the *LO_SOURCE* attribute is set to **LO_In**. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> For PXIe-5644/5645/5646, this attribute is always read-only.

The default value is 0.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the power level expected at the LO IN terminal when the *LO_SOURCE* attribute is set to **LO_In**. This value is expressed in dBm.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_injection_side`(*selector_string*)

Gets the LO injection side.

Name (value)	Description
PXIe-5663/5(	For frequencies below 517.5 MHz or above 6.4125 GHz, the LO injection side is fixed, and the RFmx driver returns an error if you specify an incorrect value. If you do not configure this attribute, the RFmx driver selects the default LO injection side based on the downconverter center frequency. Reset this attribute to return to automatic behavior.
PXIe-5665 (3.6 GHz)	Setting this attribute to Low Side is not supported for this device.
PXIe-5665 (14 GHz)	Setting this attribute to Low Side is supported for this device for frequencies greater than 4 GHz, but this configuration is not calibrated, and device specifications are not guaranteed.
PXIe-5668	Setting this attribute to Low Side is supported for some frequencies in high band, varying by the final IF frequency. This configuration is not calibrated and device specifications are not guaranteed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

** Default value:**

- PXIe-5663/5663E (frequencies < 3.0 GHz): **High Side**
- PXIe-5663/5663E (frequencies >= 3.0 GHz): **Low Side**
- PXIe-5665/5668: **High Side**

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
High Side (0)	Configures the LO signal that the device generates at a frequency higher than the RF signal. This LO frequency is given by the following formula: *fLO = fRF + fIF *
Low Side (1)	Configures the LO signal that the device generates at a frequency lower than the RF signal. This LO frequency is given by the following formula: *fLO = fRF – fIF *

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LOInjectionSide):
Specifies the LO injection side.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_leakage_avoidance_enabled`(*selector_string*)

Gets whether to reduce the effects of the instrument leakage by placing the LO outside the band of acquisition.

This attribute is ignored if:

- the bandwidth required by the measurement is more than the available instrument bandwidth after offsetting the LO.
- you set the *DOWNCONVERTER_CENTER_FREQUENCY* or *DOWNCONVERTER_FREQUENCY_OFFSET* attributes.

Note

When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set this attribute to **False** when the *LO_SOURCE* attribute is set to **Automatic_SG_SA_Shared**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5830/5831/5832/5840/5841/5842 is **True**, else the default value is **False**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
False (0)	RFmx does not modify the Downconverter Frequency Offset attribute.
True (1)	RFmx calculates the required LO offset based on the measurement configuration and appropriately sets the Downconverter Frequency Offset attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LOLeakageAvoidanceEnabled):
Specifies whether to reduce the effects of the instrument leakage by placing the LO outside the band of acquisition.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_out_power`(*selector_string*)

Gets the power level of the signal at the LO OUT terminal when the *LO_EXPORT_ENABLED* attribute is set to TRUE. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the power level of the signal at the LO OUT terminal when the *LO_EXPORT_ENABLED* attribute is set to TRUE. This value is expressed in dBm.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_pll_fractional_mode`(*selector_string*)

Gets whether to use fractional mode for the LO phase-locked loop (PLL).

Fractional mode provides a finer frequency step resolution, but may result in non harmonic spurs. Refer to the specifications document of your device for more information about fractional mode and non harmonic spurs.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> The LO PLL Fractional Mode attribute is applicable only when using the internal LO.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if the PXIe-5653 is used as the LO source.

The default value is **Enabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LOPllFractionalMode):
Specifies whether to use fractional mode for the LO phase-locked loop (PLL).

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_lo_sharing_mode(*selector_string*)

Gets the RFmx session with the respective LO sharing mode.

The following figures illustrate different connection configuration topologies for different LO Sharing modes.

You must set the *NUMBER_OF_LO_SHARING_GROUPS* attribute to 1 for the following LO connection configurations.

You must set the Num LO Sharing Groups attribute to 2 for the following LO connection configurations.

The default value is **Disabled**.

Name (Value)	Description
Disabled (0)	LO Sharing is disabled.
External Star (3)	The LO connection configuration is configured as External Star.
External Daisy Chain (4)	The LO connection configuration is configured as External Daisy Chain.
Splitter and Daisy Chain (5)	The LO connection configuration is configured as Splitter and Daisy Chain. With this option, the only allowed value for the Number of LO Sharing Groups attribute is 1.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LOSharingMode):
Specifies the RFmx session with the respective LO sharing mode.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_lo_source(*selector_string*)

Gets the local oscillator (LO) signal source used to downconvert the RF input signal.

If this attribute is set to "" (empty string), RFmx uses the internal LO source. For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

If no signal downconversion is required, this attribute is ignored.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Onboard**.

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
None (None)	Specifies that no LO source is required to downconvert the RF input signal.
Onboard (Onboard)	Specifies that the onboard synthesizer is used to generate the LO signal that downconverts the RF input signal.
LO_In (LO_In)	Specifies that the LO source used to downconvert the RF input signal is connected to the LO IN connector on the front panel.
Secondary (Secondary)	Specifies that the LO source uses the PXIe-5830/5831/5832/5840 internal LO. This value is valid on only the PXIe-5840 with PXIe-5653, PXIe-5831 with PXIe-5653 (LO1 stage only), or PXIe-5832 with PXIe-5653 (LO1 stage only).
SG_SA_Shared (SG_SA_Shared)	Specifies that the internal LO can be shared between RFmx and RFSG sessions. RFmx selects an internal synthesizer and the synthesizer signal is switched to both the RX and TX mixers. This value is valid only on PXIe-5830/5831/5832/5841/5842.

Automatic_SG_SA_Sha (Automatic_SG_SA_Sha	Specifies whether RFmx automatically configures the signal analyzer to use the LO utilized by the signal generator on the same vector signal transceiver (VST) based on the configured measurements.
	When using instruments that do not have LOs with excellent phase noise and to minimize the contribution of the instrument's phase noise affecting your measurements, NI recommends to share the LO between the signal generator (SG) and the signal analyzer (SA).
	This value is recommended in test setups that use a VST with NI-RFSG to generate a signal at the DUT's input and RFmx to measure the signal at the DUT's output.
	This value automatically:
	• determines whether the SG LO can be shared with SA based on the test instrument used, selected measurement, and the measurement settings.
	• configures instrument specific attributes on SA to share the LO between the generator and analyzer, whenever possible.
	To enable automatically sharing SG LO with SA, you must first setup the required device specific physical connections mentioned below and then follow the steps in the recommended order.
	PXIe-5840/5841: SG LO is shared with SA via an external path. Hence, you must connect RF Out LO Out to RF In LO In using a cable.
	PXIe-5841 with PXIe-5655/5842/PXIe-5830/5831/5832: SG LO is shared with SA via an internal path. Hence, an external cable connection is not required.
	NI recommends the following order of steps:
	• Set LO Source attribute to Automatic SG SA Shared in NI-RFSG (or enable Automatic SG SA shared LO on NI-RFSG Playback Library).
	• Set LO Source attribute to **Automatic_SG_SA_Shared** in RFmx.
	• Configure any additional settings on RFSG and RFmx, including selecting waveforms.
	• Initiate RFSG.
	• Initiate RFmx.
	When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set the LO Leakage Avoidance Enabled attribute to **False** and LO Source attribute to **Automatic_SG_SA_Shared**.
	Refer to following methods for examples in RFmx WLAN and RFmx NR that show the behavior of Automatic SG SA Shared LO.
	`<LabVIEW directory>\examples\RFmx\WLAN\RFmxWLAN FEM Test with Automatic SG SA Shared LO.vi`
	`<LabVIEW directory>\examples\RFmx\NR\RFmxNR FEM Test with Automatic SG SA Shared LO.vi`
	This value is valid only on PXIe-5830/5831/5832/5840/5841/5842.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the local oscillator (LO) signal source used to downconvert the RF input signal.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_lo_splitter_loss(*selector_string*)

Gets an array of the insertion losses inherent to the RF Splitter. This value is expressed in dB.

You must specify the frequencies at which the losses were measured using the *LO_SPLITTER_LOSS* attribute.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies an array of the insertion losses inherent to the RF Splitter. This value is expressed in dB.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_lo_splitter_loss_frequency(*selector_string*)

Gets the frequencies corresponding to the insertion loss inherent to the RF Splitter, as specified by the *LO_SPLITTER_LOSS_FREQUENCY* attribute. This value is expressed in Hz.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the frequencies corresponding to the insertion loss inherent to the RF Splitter, as specified by the *LO_SPLITTER_LOSS_FREQUENCY* attribute. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_temperature`(*selector_string*)

Gets the current temperature of the LO module associated with the device. This value is expressed in degrees Celsius.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Note

This attribute is not supported if you are using an external LO.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5840/5841/5842

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the current temperature of the LO module associated with the device. This value is expressed in degrees Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_lo_vco_frequency_step_size`(*selector_string*)

Gets the step size for tuning the internal voltage-controlled oscillator (VCO) used to generate the LO signal. The valid values for LO1 include 1 Hz to 50 MHz and for LO2 include 1 Hz to 100 MHz.

Note

Do not set this attribute with the *LO_FREQUENCY_STEP_SIZE* attribute.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 1 MHz.

Supported devices: PXIe-5830/5831/5832

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the step size for tuning the internal voltage-controlled oscillator (VCO) used to generate the LO signal. The valid values for LO1 include 1 Hz to 50 MHz and for LO2 include 1 Hz to 100 MHz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_load_options(*selector_string*)

Gets the configurations to skip while loading from a file using the *load_configurations()* method .

Name (value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session

The default value is an empty array.

Name (Value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.LoadOptions):
Specifies the configurations to skip while loading from a file using the *load_configurations()* method .

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_lte_signal_configuration(*signal_name=''*)

Creates a LTE signal configuration for specified signal name. Existing LTE signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

”signal::sig1”

”sig1”

Returns
Returns an object of type Lte.

Return type
lte (Lte)

`get_mechanical_attenuation_auto`(*selector_string*)

Gets whether the RFmx driver chooses an attenuation setting based on the hardware settings.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured in the Mechanical Attenuation Value attribute.
True (1)	Specifies that the measurement computes the mechanical attenuation.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.MechanicalAttenuationAuto):
Specifies whether the RFmx driver chooses an attenuation setting based on the hardware settings.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_mechanical_attenuation_value`(*selector_string*)

Gets the level of mechanical attenuation for the RF path. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *MECHANICAL_ATTENUATION_AUTO* attribute to **False**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Valid values

Name (value)	Description
PXIe-5663/5663E	0, 16
PXIe-5665 (3.6 GHz)	0, 10, 20, 30
PXIe-5665 (14 GHz), PXIe-5668	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the level of mechanical attenuation for the RF path. This value is expressed in dB.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_mixer_level(*selector_string*)

Gets the mixer level. This value is expressed in dBm.

The mixer level represents the attenuation value to apply to the input RF signal as it reaches the first mixer in the signal chain. If you do not set this attribute, the RFmx driver automatically selects an optimal mixer level value based on the reference level.

If you set the *MIXER_LEVEL* and *MIXER_LEVEL_OFFSET* attributes at the same time, the RFmx driver returns an error.

This attribute is read-only for PXIe-5663/5663E devices.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5665/5668	-10
All other devices	N/A

The valid values for this attribute depend on your device configuration.

Supported devices: PXIe-5663/5663E/5665/5668

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the mixer level. This value is expressed in dBm.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_mixer_level_offset`(*selector_string*)

Gets the number of dB by which to adjust the device mixer level.

Specifying a positive value for this attribute configures the device for moderate distortion and low noise, and specifying a negative value results in low distortion and higher noise. You cannot set the *MIXER_LEVEL_OFFSET* and *MIXER_LEVEL* attributes at the same time.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0. The default value specifies device settings that are the best compromise between distortion and noise.

Supported devices: PXIe-5663/5663E/5665/5668

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the number of dB by which to adjust the device mixer level.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_module_revision`(*selector_string*)

Gets the revision of the RF downconverter module.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

> **Note**
>
> For PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860, this attribute returns the revision of the VST module.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the revision of the RF downconverter module.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_nr_signal_configuration`(*signal_name=''*)

Creates a NR signal configuration for specified signal name. Existing NR signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type NR.

Return type
nr (*NR*)

`get_number_of_lo_sharing_groups`(*selector_string*)

Gets the RFmx session with the number of LO sharing groups.

The default value is 1.

The valid values are 1 and 2.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (int):
Specifies the RFmx session with the number of LO sharing groups.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_number_of_raw_iq_records`(*selector_string*)

Gets the number of raw IQ records to acquire to complete measurement averaging.

> **Note**
> This attribute returns a value of 0 when RFmx cannot provide I/Q data for the specified measurement configuration.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (int):
Returns the number of raw IQ records to acquire to complete measurement averaging.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_optimize_path_for_signal_bandwidth`(*selector_string*)

Optimizes RF path for the signal bandwidth that is centered on the IQ carrier frequency.

You can disable this attribute to avoid changes to the RF path when changing the signal bandwidth.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Automatic**.

Supported devices: PXIe-5830/5831/5832/5841/5842

Name (Value)	Description
Disabled (0)	Disables the optimized path for signal bandwidth.
Enabled (1)	Enables the optimized path for signal bandwidth.
Automatic (2)	Automatically enables the optimized path based on other configurations.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.OptimizePathForSignalBandwidth):
Optimizes RF path for the signal bandwidth that is centered on the IQ carrier frequency.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_osp_delay_enabled`(*selector_string*)

Gets whether to enable the digitizer OSP block to delay Reference Triggers, along with the data samples, moving through the OSP block.

If you set this attribute to **Disabled**, the Reference Triggers bypass the OSP block and are processed immediately.

Enabling this attribute requires the following equipment configurations:

- All digitizers being used must be the same model and hardware revision.
- All digitizers must use the same firmware.
- All digitizers must be configured with the same I/Q rate.
- All devices must use the same signal path.

For more information about the digitizer OSP block and Reference Triggers, refer to the following topics in the *NI High-Speed Digitizers Help*:

- PXIe-5622 Onboard Signal Processing (OSP)
- PXIe-5142 Onboard Signal Processing (OSP)
- PXIe-5622 Trigger Sources
- PXI-5142 Trigger Sources
- PXIe-5622 Block Diagram
- PXI-5142 Trigger Sources

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.OspDelayEnabled):
Specifies whether to enable the digitizer OSP block to delay Reference Triggers, along with the data samples, moving through the OSP block.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_overflow_error_reporting(*selector_string*)

Configures error reporting for ADC and overflows occurred during onboard signal processing. Overflows lead to clipping of the waveform.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Warning**.

Supported devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Warning (0)	RFmx returns a warning when an ADC or an onboard signal processing (OSP) overflow occurs.
Disabled (1)	RFmx does not return an error or a warning when an ADC or OSP overflow occurs.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.OverflowErrorReporting):
Configures error reporting for ADC and overflows occurred during onboard signal processing. Overflows lead to clipping of the waveform.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_phase_offset`(*selector_string*)

Gets the offset to apply to the initial I and Q phases.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values are -180 degrees to 180 degrees, inclusive.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the offset to apply to the initial I and Q phases.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_preamp_enabled`(*selector_string*)

Gets whether the RF preamplifier is enabled in the system.

PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860: If you set this attribute to **Automatic**, RFmx selects the preamplifier state based on the value of the Reference Level attribute and the center frequency. For PXIe-5830/5831/5832, the value is not coerced.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842 is **Automatic**, else the default value is **Disabled**.

Name (Value)	Description
Disabled (0)	Disables the RF preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Enabled (1)	Enables the RF preamplifier when it is in the signal path and disables it when it is not in the signal path. Only devices with an RF preamplifier on the downconverter and an RF preselector support this option. Use the RF Preamp Present attribute to determine whether the downconverter has a preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Automatic (3)	Automatically enables the RF preamplifier based on the value of the reference level. **Supported Devices:** PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.PreampEnabled):
Specifies whether the RF preamplifier is enabled in the system.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_preselector_present(*selector_string*)

Indicates whether a preselector is available on the RF downconverter module.

Name (value)	Description
TRUE	A preselector is available on the downconverter.
FALSE	No preselector is available on the downconverter.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (bool):
Indicates whether a preselector is available on the RF downconverter module.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_pulse_signal_configuration`(*signal_name=''*)

Creates a Pulse signal configuration for specified signal name. Existing Pulse signal configuration is returned if specified signal name exists.

Parameters
`signal_name` (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type Pulse.

Return type
pulse (Pulse)

`get_ready_for_advance_event_output_terminal`(*selector_string*)

Gets the destination terminal for the Ready for Advance event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the Ready for Advance event.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_ready_for_advance_event_terminal_name`(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForAdvanceEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForAdvanceEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForAdvanceEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1)
- **All other devices**: */DigitizerName/ReadyForAdvanceEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_ready_for_reference_event_output_terminal`(*selector_string*)

Gets the destination terminal for the Ready for Reference event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the Ready for Reference event.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_ready_for_reference_event_terminal_name(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForReferenceEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForReferenceEvent*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForReferenceEvent*, where *BasebandModule*is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/ReadyForReferenceEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

Note

This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_ready_for_start_event_output_terminal(*selector_string*)

Gets the destination terminal for the Ready for Start event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the Ready for Start event.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_ready_for_start_event_terminal_name`(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/ReadyForStartEvent*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/ReadyForStartEvent*, where *BasebandModule*is the name of the baseband module for your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/ReadyForStartEvent*, where *ModuleName * is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/ReadyForStartEvent*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_acquisition_type`(*selector_string*)

Gets the recommended acquisition type for the last committed measurement configuration.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Name (Value)	Description
IQ (0)	Indicates that the recommended acquisition type is I/Q. Use the Analyze (IQ) method to perform the measurement.
Spectral (1)	Indicates that the recommended acquisition type is Spectral. Use Analyze (Spectrum) method to perform the measurement.
IQ or Spectral (2)	Indicates that the recommended acquisition type is I/Q or Spectral. Use either Analyze (IQ) or Analyze (Spectrum) method to perform the measurement.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.RecommendedAcquisitionType):
Returns the recommended acquisition type for the last committed measurement configuration.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_recommended_center_frequency(*selector_string*)

Gets the recommended center frequency of the RF signal. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended center frequency of the RF signal. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_iq_acquisition_time`(*selector_string*)

Gets the recommended acquisition time for I/Q acquisition. This value is expressed in seconds.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended acquisition time for I/Q acquisition. This value is expressed in seconds.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_iq_minimum_sample_rate`(*selector_string*)

Gets the recommended minimum sample rate for I/Q acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended minimum sample rate for I/Q acquisition. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_iq_pre_trigger_time`(*selector_string*)

Gets the recommended pretrigger time for I/Q acquisition. This value is expressed in seconds.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended pretrigger time for I/Q acquisition. This value is expressed in seconds.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_number_of_records`(*selector_string*)

Gets the recommended number of records to acquire to complete measurement averaging.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (int):
Returns the recommended number of records to acquire to complete measurement averaging.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_spectral_acquisition_span`(*selector_string*)

Gets the recommended acquisition span for spectral acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended acquisition span for spectral acquisition. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_recommended_spectral_fft_window`(*selector_string*)

Gets the recommended FFT window type for spectral acquisition.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Name (Value)	Description
None (0)	Indicates that the measurement does not use FFT windowing to reduce spectral leakage.
Flat Top (1)	Indicates a Flat Top FFT window type.
Hanning (2)	Indicates a Hanning FFT window type.
Hamming (3)	Indicates a Hamming FFT window type.
Gaussian (4)	Indicates a Gaussian FFT window type.
Blackman (5)	Indicates a Blackman FFT window type.
Blackman-Harris (6)	Indicates a Blackman-Harris FFT window type.
Kaiser-Bessel (7)	Indicates a Kaiser-Bessel FFT window type.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.RecommendedSpectralFftWindow):
Returns the recommended FFT window type for spectral acquisition.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_recommended_spectral_resolution_bandwidth(*selector_string*)

Gets the recommended FFT bin width for spectral acquisition. This value is expressed in Hz.

Note

This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended FFT bin width for spectral acquisition. This value is expressed in Hz.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_recommended_trigger_minimum_quiet_time(*selector_string*)

Gets the recommended minimum quiet time during which the signal level must be below the trigger value for triggering to occur. This value is expressed in seconds.

> **Note**
>
> This attribute is supported only when:

- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1", or
- *nirfmxinstr.session.Session()* constructor is called with option string "AnalysisOnly=1;MaxNumWfms:<n>". Use "*instr<n>*" as the selector string to read this attribute.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Returns the recommended minimum quiet time during which the signal level must be below the trigger value for triggering to occur. This value is expressed in seconds.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_rf_attenuation_auto(*selector_string*)

Gets whether the RFmx driver computes the RF attenuation.

If you set this attribute to **True**, the RFmx driver chooses an attenuation setting based on the reference level configured on the personality.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E, PXIe-5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured using RF Attenuation Value attribute.
True (1)	Specifies that the RFmx driver computes the RF attenuation.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.RFAttenuationAuto):
Specifies whether the RFmx driver computes the RF attenuation.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_rf_attenuation_step_size`(*selector_string*)

Gets the step size for the RF attenuation level. This value is expressed in dB. The actual RF attenuation is coerced up to the next highest multiple of the specified step size. If the mechanical attenuators are not available to implement the coerced RF attenuation, the solid state attenuators are used.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0
PXIe-5603/5665 (3.6 GHz)	1.0
PXIe-5605/5665 (14 GHz), PXIe-5606/5668	5.0

Valid values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0 to 93.0, continuous
PXIe-5603/5665 (3.6 GHz)	1.0 to 74.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (low band), PXIe-5606/5668 (low band)	1.0 to 106.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (high band), PXIe-5606/5668 (high band)	5.0 to 75.0, in 5 dB steps

Supported devices: PXIe-5663, PXIe-5665, PXIe-5668

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the step size for the RF attenuation level. This value is expressed in dB. The actual RF attenuation is coerced up to the next highest multiple of the specified step size. If the mechanical attenuators are not available to implement the coerced RF attenuation, the solid state attenuators are used.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_rf_attenuation_value`(*selector_string*)

Gets the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *RF_ATTENUATION_AUTO* attribute to **False**.

Name (value)	Description
PXIe-5663/5663E	You can change the attenuation value to modify the amount of noise and distortion. Higher attenuation levels increase the noise level but decreases distortion; lower attenuation levels decrease the noise level but increases distortion.
PXIe-5603/5605/5	Refer to the PXIe-5665 or the PXIe-5668 RF Attenuation and Signal Levels topic in the NI RF Vector Signal Analyzers Help for more information about configuring attenuation.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The valid values for this attribute depend on the device configuration.

Supported devices: PXIe-5663/5663E/5603/5605/5665/5668

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_rf_highpass_filter_frequency(*selector_string*)

Gets the maximum corner frequency of the high pass filter in the RF signal path. The device uses the highest frequency high-pass filter option below or equal to the value you specify and returns a coerced value. Specifying a value of 0 disables high pass filtering silly.

For multispan acquisitions, the device uses the appropriate filter for each subspan during acquisition, depending on the details of your application and the value you specify. In multispan acquisition spectrum applications, this attribute returns the value you specified rather than a coerced value if multiple high-pass filters are used during the acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

The valid values range from 0 to 26.5.

Supported devices: PXIe-5668

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the maximum corner frequency of the high pass filter in the RF signal path. The device uses the highest frequency high-pass filter option below or equal to the value you specify and returns a coerced value. Specifying a value of 0 disables high pass filtering silly.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_rf_preamp_present`(*selector_string*)

Indicates whether an RF preamplifier is available on the RF downconverter module.

Name (value)	Description
TRUE	A preamplifier is available on the downconverter.
FALSE	No preamplifier is available on the downconverter.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (bool):
Indicates whether an RF preamplifier is available on the RF downconverter module.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_s_parameter_external_attenuation_type`(*selector_string*)

Returns the type of S-parameter that is applied to the measurements on the specified port on a Calplane. You can use the Selector String input to specify the name of the Calplane and port to configure for S-parameter.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

On a MIMO session if you do not specify the port name, this method will return an error. To get S-parameter external attenuation type from a specific MIMO port, use the port specifier with or without the calplane name.

Example: "calplane::plane1/port::myrfsa1/0".

> **Note**
>
> For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". Use *get_available_ports()* method to get the valid port names.

Example:

""

"calplane::plane0"

"calplane::plane0/port::if0"

"port::if0"

"calplane::plane0/port::myrfsa1/0"

"calplane::plane0/port::myrfsa1/0/if0"

Returns

s_parameter_type (enums.SParameterType):
This parameter returns the type of S-parameter which is applied to measurements on the specified port of a Calplane.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (s_parameter_type, error_code)

`get_self_calibrate_last_date_and_time`(*selector_string*, *self_calibrate_step*)

Returns the date and time of the last successful self-calibration. On a MIMO session, use the **Selector String** parameter to get the last successful self-calibration date and time for a specific MIMO port.

Supported Devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

> **Note**
>
> For PXIe-5644/5645/5646 devices, you must select **Image Suppression** for the **Self Calibrate Step** parameter.

Parameters

- **selector_string** (*string*) – This parameter specifies a Selector String comprising of a MIMO port on a MIMO session. The default value is "" (empty string).

  Example:

  ""

  "port::myrfsa1/0"

  You can use the RFmxInstr Build Port String method to build the selector string.

- **self_calibrate_step** (enums.SelfCalibrateSteps, *int*) – This parameter specifies the self-calibration step to query for the last successful self-calibration date and time data. The default value is **Preselector Alignment**.

Name (Value)	Description
Preselector Alignment (1)	Selects the Preselector Alignment self-calibration step.
Gain Reference (2)	Selects the Gain Reference self-calibration step.
IF Flatness (4)	Selects the IF Flatness self-calibration step.
Digitizer Self Cal (8)	Selects the Digitizer Self Cal self-calibration step.
LO Self Cal (10)	Selects the LO Self Cal self-calibration step.
Amplitude Accuracy (20)	Selects the Amplitude Accuracy self-calibration step.
Residual LO Power (40)	Selects the Residual LO Power self-calibration step.
Image Suppression (80)	Selects the Image Suppression self-calibration step.
Synthesizer Alignment (100)	Selects the Synthesizer Alignment self-calibration step.
DC Offset (200)	Selects the DC Offset self-calibration step.

Returns

timestamp (hightime.datetime):
This parameter returns the date and time of the last successful self-calibration.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (timestamp, error_code)

get_self_calibrate_last_temperature(*selector_string*, *self_calibrate_step*)

Returns the temperature at the last successful self-calibration. On a MIMO session, use the **Selector String** parameter to get the last successful self-calibration temperature for a specific MIMO port.

Supported Devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831 (IF only)/5832 (IF only)/5840/5841/5842/5860

> **Note**
>
> For PXIe-5644/5645/5646 devices, you must select **Image Suppression** for the **Self Calibrate Step** parameter.

Parameters

- **selector_string** (*string*) – This parameter specifies a Selector String comprising of a MIMO port on a MIMO session.

  Example:

  ””

  ”port::myrfsa1/0”

  You can use the *build_port_string()* method to build the selector string.

- **self_calibrate_step** (enums.SelfCalibrateSteps, *int*) – This parameter specifies the self-calibration step to query for the last successful self-calibration temperature data. The default value is **Preselector Alignment**.

Name (Value)	Description
Preselector Alignment (1)	Selects the Preselector Alignment self-calibration step.
Gain Reference (2)	Selects the Gain Reference self-calibration step.
IF Flatness (4)	Selects the IF Flatness self-calibration step.
Digitizer Self Cal (8)	Selects the Digitizer Self Cal self-calibration step.
LO Self Cal (10)	Selects the LO Self Cal self-calibration step.
Amplitude Accuracy (20)	Selects the Amplitude Accuracy self-calibration step.
Residual LO Power (40)	Selects the Residual LO Power self-calibration step.
Image Suppression (80)	Selects the Image Suppression self-calibration step.
Synthesizer Alignment (100)	Selects the Synthesizer Alignment self-calibration step.
DC Offset (200)	Selects the DC Offset self-calibration step.

Returns

temperature (float):
This parameter returns the temperature at the last self-calibration. This value is expressed in degree Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (temperature, error_code)

get_self_calibration_validity_check(*selector_string*)

Gets whether the RFmx driver validates the self-calibration data.

You can specify the time interval required to perform the check using the *SELF_CALIBRATION_VALIDITY_CHECK_TIME_INTERVAL* attribute.

NI recommends to perform self-calibration using the *self_calibrate()* method when RFmx reports an invalid self-calibration data warning.

Note

The RFmx driver does not consider self-calibration range data during self calibration validity check.

The default value is **Off**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

Name (Value)	Description
Off (0)	Indicates that RFmx does not check whether device self-calibration data is valid.
Enabled (1)	Indicates that RFmx checks whether device self-calibration data is valid and reports a warning from the RFmx Commit and RFmx Initiate methods when the data is invalid.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.SelfCalibrationValidityCheck):
Specifies whether the RFmx driver validates the self-calibration data.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_self_calibration_validity_check_time_interval`(*selector_string*)

Gets the minimum time between two self calibration validity checks. This value is expressed in seconds.

When you call RFmx Commit or Initiate methods by enabling the *SELF_CALIBRATION_VALIDITY_CHECK* attribute, the RFmx driver checks if the amount of time specified by the Self Calibration Validity Check Time Interval attribute has elapsed before validating the calibration data.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the minimum time between two self calibration validity checks. This value is expressed in seconds.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_serial_number`(*selector_string*)

Gets the serial number of the RF downconverter module.

> **Note**
>
> For PXIe-5644/5645/5646 and PXIe-5820/5840/5841/5842/5860, this attribute returns the serial number of the VST module. For PXIe-5830/5831/5832, this attribute returns the serial number of PXIe-3621/3622/3623.

On a MIMO session, use "port::<deviceName>/<channelNumber>" as the Selector String to read this attribute. You can use the *build_port_string()* method to build the selector string.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the serial number of the RF downconverter module.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`classmethod get_session`(*resource_name*, *option_string*, *grpc_options=None*)

Gets a session, if it exists for given resource name; else, returns a new one.

Parameters

- **resource_name** (*string*) – Specifies the resource name of the device to initialize.
- **option_string** (*string*) – Sets the initial value of certain properties for the session. The following attributes are used in this parameter:

  RFmxSetup, Simulate, AnalysisOnly.

  To simulate a device using the NI 5622 (25 MHz) digitizer, set the Digitizer field to 5622_25MHz_DDC and the Simulate field to 1 You can set the Digitizerfield to 5622_25MHz_DDC only when using the NI 5665. To use AnalysisOnly mode, specify the string as "AnalysisOnly=1". In this mode, user is responsible for waveform acquisition and RFmx driver will perform analysis on user specified IQ waveform or Spectrum. Use personality specific Analyze functions to perform measurements. To set multiple attributes, separate their assignments with a comma.

Returns

session (Session):
The RFmx session object.

is_new_session (bool):
True if new session is created; otherwise False.

Return type
Tuple (session, is_new_session)

`get_signal_configuration_names`(*selector_string*, *personality_filter*)

Returns the signal names and corresponding personality type, for the personality type selected in the personalityFilter parameter.

Parameters

- **selector_string** (*string*) – Specifies the selector string. Pass an empty string.
- **personality_filter** (enums.Personalities, *int*) – Returns an array of personalities where each entry corresponds to the personality of each signal name in the signalNames array.

Returns

signal_names (string):
Returns an array of the signal names.

personality (enums.Personalities):
Returns an array of personalities where each entry corresponds to the personality of each signal name in the signalNames array.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes an error or warning condition.

Return type
Tuple (signal_names, personality, error_code)

`get_smu_channel`(*selector_string*)

Gets the output channel to be used for noise figure (NF) measurement in RFmx.

The default value is 0.

Parameters
`selector_string` (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the output channel to be used for noise figure (NF) measurement in RFmx.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_smu_resource_name`(*selector_string*)

Gets the resource name assigned by Measurement and Automation Explorer (MAX) for NI Source Measure Units (SMU) which is used as the noise source power supply for Noise Figure (NF) measurement, for example, PXI1Slot3, where PXI1Slot3 is an instrument resource name. SMU Resource Name can also be a logical IVI name.

Supported devices: PXIe-4138, PXIe-4139, PXIe-4139 (40 W), and PXIe-4143 SMUs.

Parameters
`selector_string` (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the resource name assigned by Measurement and Automation Explorer (MAX) for NI Source Measure Units (SMU) which is used as the noise source power supply for Noise Figure (NF) measurement, for example, PXI1Slot3, where PXI1Slot3 is an instrument resource name. SMU Resource Name can also be a logical IVI name.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_specan_signal_configuration`(*signal_name=''*)

Creates a SpecAn signal configuration for specified signal name. Existing SpecAn signal configuration is returned if specified signal name exists.

Parameters
`signal_name` (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type SpecAn.

Return type
specan (SpecAn)

`get_start_trigger_digital_edge`(*selector_string*)

Gets the active edge for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Rising Edge**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Rising Edge (0)	The trigger asserts on the rising edge of the signal.
Falling Edge (1)	The trigger asserts on the falling edge of the signal.

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.StartTriggerDigitalEdge):
Specifies the active edge for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_start_trigger_digital_edge_source`(*selector_string*)

Gets the source terminal for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):

Specifies the source terminal for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_start_trigger_export_output_terminal`(*selector_string*)

Gets the destination terminal for the exported start trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the exported start trigger.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_start_trigger_terminal_name(*selector_string*)

Gets the fully qualified signal name as a string.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/StartTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/StartTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/StartTrigger,*, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/StartTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

> **Note**
>
> This attribute is not supported on a MIMO session.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_start_trigger_type(*selector_string*)

Gets whether the start trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No start trigger is configured.
Digital Edge (1)	The start trigger is not asserted until a digital edge is detected. The source of the digital edge is specified by the Start Trigger Digital Edge Source attribute.
Software (3)	The start trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.StartTriggerType):
Specifies whether the start trigger is a digital edge or a software trigger.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

Tuple (attr_val, error_code)

get_subspan_overlap(*selector_string*)

Use subspan overlap process to eliminate or reduce analyzer spurs. To enable this feature, specify a non-zero percentage overlap between consecutive subspans in a spectrum acquisition.

If a value greater than 0 is specified, then for each spectral line in the resulting spectrum, the driver acquires data twice with slightly different hardware settings, so that the analyzer spurs, if any, are present at different frequencies in the two acquisitions. Typically, LO frequency is shifted between the acquisitions causing analyzer spurs that are relative to the LO frequency, to move from one frequency to another. Those spurs, which are present in only one of the acquisitions for each spectral line, get removed.

The subspan overlap feature will not remove any spurs from the Device Under Test or modify the signal being measured; unlike the analyzer spurs, the spurs in the signal being measured stay at a constant frequency in the two acquisitions.

Note

Subspan overlap process effectively is performing minimum averaging, which might reduce the measured noise floor level. RFmx Spectrum Averaging can be enabled to minimize the effect of subspan overlap on the noise floor.

> **Note**
>
> RFmx may apply further shifts to the specified value to accommodate fixed-frequency edges of components such as preselectors.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values

Name (value)	Description
PXIe-5820/5830/5831/5832/5840/5841/5860	0
PXIe-5842	0, 50
PXIe-5665/5668	0 to <100

Supported devices: PXIe-5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Use subspan overlap process to eliminate or reduce analyzer spurs. To enable this feature, specify a non-zero percentage overlap between consecutive subspans in a spectrum acquisition.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_temperature_read_interval(*selector_string*)

Gets the minimum time difference between temperature sensor readings. This value is expressed in seconds.

When you call the RFmx Initiate method, RFmx checks if the amount of time specified by this attribute has elapsed before reading the hardware temperature.

> **Note**
>
> RFmx ignores Temperature Read Interval attribute if you read the *DOWNCONVERTER_GAIN* attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the minimum time difference between temperature sensor readings. This value is expressed in seconds.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_thermal_correction_headroom_range(*selector_string*)

Gets the expected thermal operating range of the instrument from the self-calibration temperature returned from the *DEVICE_TEMPERATURE* attribute. This value is expressed in degree Celsius.

For example, if this attribute is set to 5.0, and the device is self-calibrated at 35 degrees Celsius, then you can expect to run the device from 30 degrees Celsius to 40 degrees Celsius with corrected accuracy and no overflows. Setting this attribute with a smaller value can result in improved dynamic range, but you must ensure thermal stability while the instrument is running. Operating the instrument outside of the specified range may cause degraded performance and ADC or DSP overflows.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	5
PXIe-5840/5841	10

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the expected thermal operating range of the instrument from the self-calibration temperature returned from the *DEVICE_TEMPERATURE* attribute. This value is expressed in degree Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_thermal_correction_temperature_resolution(*selector_string*)

Gets the temperature change required before RFmx recalculates the thermal correction settings when entering the running state. This value is expressed in degree Celsius.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	0.2
PXIe-5840/5841	1.0

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (float):
Specifies the temperature change required before RFmx recalculates the thermal correction settings when entering the running state. This value is expressed in degree Celsius.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_trigger_export_output_terminal(*selector_string*)

Gets the destination terminal for the exported Reference Trigger. You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646 and PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists on only PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for the PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid on only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Specifies the destination terminal for the exported Reference Trigger. You can also choose not to export any signal.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_trigger_terminal_name(*selector_string*)

Gets the fully qualified signal name as a string.

> **Note**
>
> This attribute is not supported on a MIMO session.

The standard format is as follows:

- **PXIe-5820/5840/5841/5842**: */ModuleName/ai/0/RefTrigger*, where *ModuleName* is the name of your device in MAX.
- **PXIe-5830/5831/5832**: */BasebandModule/ai/0/RefTrigger*, where *BasebandModule* is the name of your device in MAX.
- **PXIe-5860**: */ModuleName/ai/ChannelNumber/RefTrigger*,, where *ModuleName *is the name of your device in MAX and ChannelNumber is the channel number (0 or 1).
- **All other devices**: */DigitizerName/RefTrigger*, where *DigitizerName* is the name of your associated digitizer module in MAX.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters
selector_string (*string*) – Pass an empty string.

Returns

attr_val (string):
Returns the fully qualified signal name as a string.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

get_tuning_speed(*selector_string*)

Makes tradeoffs between tuning speed and phase noise.

> **Note**
>
> This attribute is not supported if you are using an external LO.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> The PXIe-5830/5831/5832/5840/5841/5842 supports only **Medium** for this attribute.

Default value: **Normal** for PXIe-5663/5663E/5665/5668, **Medium** for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Normal (0)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an underdamped response. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a narrow loop bandwidth.
Medium (1)	Specifies that the RF downconverter module uses a medium loop bandwidth. This value is not supported on PXIe-5663/5663E/5665/5668 devices.
Fast (2)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an overdamped response. Setting this attribute to **Fast** allows the frequency to settle significantly faster for some frequency transitions at the expense of increased phase noise. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a wide loop bandwidth.

Parameters

selector_string (*string*) – Pass an empty string.

Returns

attr_val (enums.TuningSpeed):
Makes tradeoffs between tuning speed and phase noise.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (attr_val, error_code)

`get_vna_signal_configuration`(*signal_name=''*)

Creates a VNA signal configuration for specified signal name. Existing VNA signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type Vna.

Return type
vna (Vna)

`get_warning`()

Retrieves and then clears the warning information for the session.

Returns

warning_code (int):
Contains the latest warning code.

warning_message (string):
Contains the latest warning description.

Return type
Tuple (warning_code, warning_message)

`get_wlan_signal_configuration`(*signal_name=''*)

Creates a WLAN signal configuration for specified signal name. Existing WLAN signal configuration is returned if specified signal name exists.

Parameters
signal_name (*string*) –

Specifies the name of the signal. This parameter accepts the signal name with or without the "signal::" prefix.
Example:

"signal::sig1"

"sig1"

Returns
Returns an object of type Wlan.

Return type
wlan (Wlan)

`is_self_calibrate_valid`(*selector_string*)

Returns an array to indicate which calibration steps contain valid calibration data. To omit steps with the valid calibration data from self-calibration, you can pass the **Valid Steps** parameter to the **Steps To Omit**

parameter of the *self_calibrate()* method. On a MIMO session, use the **Selector String** parameter to get the self-calibration validity for a specific MIMO port.

Supported devices: PXIe-5663/5663E/5665/5668

Parameters

selector_string (*string*) – This parameter specifies a Selector String comprising of a MIMO port on a MIMO session.

Example:

””

”port::myrfsa1/0”

You can use the *build_port_string()* method to build the selector string.

Returns

self_calibrate_valid (int):
This parameter returns TRUE if all the calibration data is valid and FALSE if any of the calibration data is invalid.

valid_steps (enums.SelfCalibrateSteps):
This parameter returns an array of valid steps.

Name (Value)	Description
Preselector Alignment (1)	Indicates the Preselector Alignment calibration data is valid. This step generates coefficients to align the preselector across the frequency range for your device.
Gain Reference (2)	Indicates the Gain Reference calibration data is valid. This step measures the changes in gain since the last external calibration was run.
IF Flatness (4)	Indicates the IF Flatness calibration data is valid. This step measures the IF response of the entire system for each of the supported IF filters.
Digitizer Self Cal (8)	Indicates the Digitizer Self Cal calibration data is valid. This step calls for digitizer self-calibration if the digitizer is associated with the RF downconverter.
LO Self Cal (10)	Indicates the LO Self Cal calibration data is valid. This step calls for LO self-calibration if the LO source module is associated with the RF downconverter.

error_code (int):
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
Tuple (self_calibrate_valid, valid_steps, error_code)

load_configurations(*file_path*)

Loads the attributes of an RFmx session saved in a file. This file can be generated using *save_all_configurations()* method or using the RF Signal Analyzer panel in InstrumentStudio.

You can specify the configurations to skip while loading from a file using the *LOAD_OPTIONS* attribute.

> **Note**
>
> If the file contains a named signal configuration which is already present in the session, then this method will return an error. NI recommendeds to call the *reset_entire_session()* method to delete all the named signal configurations in the session.

Parameters

file_path (*string*) – This parameter specifies the complete path to the file from which the configurations are to be loaded. Default file extension: .rfmxconfig

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

load_s_parameter_external_attenuation_table_from_s2p_file(*selector_string*, *table_name*, *s2p_file_path*, *s_parameter_orientation*)

Stores the S-parameter table from the S2P file in the calibration plane specified by the **Selector String** parameter. S-parameter tables are used for fixture de-embedding. On a MIMO session, the S-parameter table is stored for each MIMO port in the specified calibration plane.

> **Note**
>
> If there is only one table configured in any calibration plane, it is automatically selected as the active table.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which the external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  > **Note**
  >
  > For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". If you specify "port::all", all MIMO ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

"calplane::plane0/port::if0"

"port::if0"

"calplane::plane0/port::all"

"calplane::plane0/port::myrfsa1/0"

"calplane::plane0/port::myrfsa1/0, port::myrfsa2/0"

"calplane::plane0/port::myrfsa1/0/if0"

- **table_name** (*string*) – This parameter specifies the name to be associated with S-parameter table within a calibration plane. Provide a unique name, such as "table1" to configure the table. The default value is "" (empty string).

  Example:

  ""

  "table1"

- **s2p_file_path** (*string*) – This parameter specifies the path to the S2P file that contains S-parameter table information for the specified port.
- **s_parameter_orientation** (enums.SParameterOrientation, *int*) – This parameter specifies the orientation of the data in the S-parameter table relative to the port you specify. The default value is **Port2 Towards DUT**.

Name (Value)	Description
Port1 Towards DUT (0)	Port 1 of the S2P is oriented towards the DUT.
Port2 Towards DUT (1)	Port 2 of the S2P is oriented towards the DUT.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

lte_clear_noise_calibration_database(*selector_string*)

Clears the noise calibration database used for noise compensation.

Parameters
selector_string (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

reset_attribute(*selector_string*, *attribute_id*)

Resets an attribute that you specify in the **attributeID** parameter to default values.

Parameters

- **selector_string** (*string*) – Specifies the selector string for the attribute being reset. Refer to the Selector String (C or LabWindows/CVI) topic for more information about configuring the selector string.

- **attribute_id** (*enums.AttributeID, int*) – Pass the ID of an attribute.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

reset_driver()

Restores the NI-RFSA driver state to a default state to avoid RFmx using any hardware or driver state that was set by the RF toolkits or other custom NI-RFSA code.

Use this method when you switch back to using RFmx to perform measurements after you have used the NI-RFSA handle to perform measurements with RF toolkits or you have used other custom NI-RFSA code. Unlike the *reset_to_default()* method, the RfmxInstr Reset Driver method does not reset RFmx attributes configured on the RFmx session. Hence, you do not need to set RFmx attributes again when switching back to RFmx measurements. Refer to RFmx SpecAn CHP - WCDMA ModAcc - CHP Example (LabVIEW) for more information about using RFmx to perform measurements.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

reset_entire_session()

Deletes all the named signal in the session and resets all attributes for the default signal instances of already loaded personalities in the session.

This method disables all the calibration planes.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

reset_to_default()

Resets the RFmxInstr attributes to their default values.

This method disables all the calibration planes.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

save_all_configurations(*file_path*)

Saves all the configured attributes in the RFmx session to a file in the specified file path. Use this method to save the current state of the RFmx session. On a MIMO session, this method saves all the configured attributes for the specified MIMO port.

> **Note**
>
> List configurations, reference waveforms and external attenuation tables are not saved by this method.

Parameters

file_path (*string*) – This parameter specifies the complete path to the file to which the configurations are to be saved. Default file extension: .rfmxconfig

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

select_active_external_attenuation_table(*selector_string*, *table_name*)

Activates the external attenuation table set by the **Table Name** parameter in the calibration plane specified by the **Selector String** parameter. On a MIMO session, this method selects the active external attenuation table for the specified MIMO port. The specified table will be used for amplitude correction during measurement.

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – This parameter specifies the calibration plane name in which either S-parameter or external attenuation table is stored. This input accepts the calibration plane name with the "calplane::" prefix. If you do not specify the calibration plane name, the default calibration plane instance is used.

  On a MIMO session, the default "" (empty string) selects the active external attenuation table for all the MIMO Ports. To configure external attenuation type for a specific MIMO port, use the port specifier with or without the calplane name.

  Example: "calplane::plane1/port::myrfsa1/0".

  > **Note**
  >
  > For PXIe-5830/5831/5832 devices, port names should also be specified along with Calplane names. Hence, the valid selector string is "calplane::<calplaneName>/port::<portName>". If you specify "port::all", all ports are considered configured. For a MIMO port, the valid selector string is "calplane::<calplaneName>/port::<deviceName>/<channelNumber>/<portName>". If you specify "port::all", all ports are considered configured. Use *get_available_ports()* method to get the valid port names.

  Example:

  ""

  "calplane::plane0"

  "calplane::plane0/port::if0"

  "port::if0"

  "calplane::plane0/port::all"

”calplane::plane0/port::myrfsa1/0”

”calplane::plane0/port::myrfsa1/0, port::myrfsa2/0”

”calplane::plane0/port::myrfsa1/0/if0”

- **table_name** (*string*) – This parameter specifies the name to be associated with external attenuation table within a calibration plane. Provide a unique name, such as “table1” to configure the table. The default value is “” (empty string).

  Example:

  ””

  ”table1”

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

self_calibrate(*selector_string*, *steps_to_omit*)

Self-calibrates the NI-RFSA device and associated modules that support self-calibration. If self-calibration completes successfully, the new calibration constants are stored immediately in the nonvolatile memory of the module. On a MIMO session, this method self-calibrates all NI-RFSA devices and associated modules that support self-calibration.

Refer to the specifications document for your device for more information about how often to self-calibrate. For more information about Self Calibrate, refer to the *niRFSA Self Cal VI* topic for your device in the *NI RF Vector Signal Analyzers Help*.

> **Note**
>
> For PXIe-5644/5645/5646, RFmx internally closes the RFSA session, performs self-calibration and opens a new session for the same device. If the RFSA session has been accessed from RFmx, using the get nirfsa session method before calling the RFmxInstr Self Calibrate method, the RFSA session will become invalid upon calling the RFmxInstr Self Calibrate.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Specifies a selector string. Pass an empty string.
- **steps_to_omit** (enums.SelfCalibrateSteps, *int*) – This parameter specifies which calibration steps to skip during the self-calibration process. The default value is an empty array, which indicates that all calibration steps are performed. The only valid value for PXIe-5820/5830/5831/5832/5840/5841/5842/5860 is an empty array.

Name (Value)	Description
Preselector Alignment (1)	Omits the Preselector Alignment step. If you omit this step and the niRFSA Is Self Cal Valid method indicates the calibration data for this step is invalid, the preselector alignment specifications are not guaranteed. This step applies only to the PXIe-5605/5606.
Gain Reference (2)	Omits the Gain Reference step. If you omit this step and the niRFSA Is Self Cal Valid method indicates the calibration data for this step is invalid, the absolute accuracy of the device is not guaranteed.
IF Flatness (4)	Omits the IF Flatness step. If you omit this step and the niRFSA Is Self Cal, valid method indicates the calibration data for this step is invalid, the IF flatness specifications are not guaranteed.
Digitizer Self Cal (8)	Omits the Digitizer Self Cal step. If you omit this step and the niRFSA Is Self Cal Valid method indicates the calibration data for this step is invalid, the absolute accuracy of the device is not guaranteed.
LO Self Cal (10)	Omits the LO Self Cal step. If you omit this step and the niRFSA Is Self Cal Valid method indicates the calibration data for this step is invalid, the LO PLL may fail to lock.
Amplitude Accuracy (20)	Not used by this method.
Residual LO Power (40)	Not used by this method.
Image Suppression (80)	Not used by this method.
Synthesizer Alignment (100)	Not used by this method.
DC Offset (200)	Not used by this method.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

self_calibrate_range(*selector_string*, *steps_to_omit*, *minimum_frequency*, *maximum_frequency*, *minimum_reference_level*, *maximum_reference_level*)

Self-calibrates all configurations within the specified frequency and reference level limits. If there is an open session for NI-RFSG for your device, it may remain open but cannot be used while this method runs.

NI recommends that no external signals are present on the RF In port while the calibration is taking place. For more information about Self Calibrate Range, refer to the *niRFSA Self Calibrate Range* method topic for your device in the *NI RF Vector Signal Analyzers Help*. On a MIMO session, this method self-calibrates all NI-RFSA devices and associated modules that support self-calibration.

Supported devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842

Note

This method does not update self-calibration date and temperature. Self-calibration range data is not saved to your device if you restart the system.

Parameters

- **selector_string** (*string*) – Specifies a selector string. Pass an empty string.
- **steps_to_omit** (enums.SelfCalibrateSteps, *int*) – This parameter specifies which calibration steps to skip during the self-calibration process. The default value is an empty array, which indicates that all calibration steps are performed. The only valid value for the PXIe-5820/5830/5831/5832/5840/5841/5842 is an empty array.

Name (Value)	Description
Preselector Alignment (1)	Not used by this method.
Gain Reference (2)	Not used by this method.
IF Flatness (4)	Not used by this method.
Digitizer Self Cal (8)	Not used by this method.
LO Self Cal (10)	Omits the LO Self Cal step. If you omit this step and the niRFSA Is Self Cal Valid method indicates the calibration data for this step is invalid, the LO PLL may fail to lock.
Amplitude Accuracy (20)	Omits the Amplitude Accuracy step. If you omit this step, the absolute accuracy of the device is not adjusted.
Residual LO Power (40)	Omits the Residual LO Power step. If you omit this step, the Residual LO Power performance is not adjusted.
Image Suppression (80)	Omits the Image Suppression step. If you omit this step, the Residual Sideband Image performance is not adjusted.
Synthesizer Alignment (100)	Omits the VCO Alignment step. If you omit this step, the LO PLL will not get adjusted.
DC Offset (200)	Omits the DC Offset step.

- **minimum_frequency** (*float*) – This parameter specifies the minimum frequency for the custom self calibration range. This value is expressed in Hz.

> **Note**
>
> For PXIe-5830/5831/5832, only the applicable ports within the specified frequency range are calibrated.

- **maximum_frequency** (*float*) – This parameter specifies the maximum frequency for the custom self calibration range. This value is expressed in Hz.
- **minimum_reference_level** (*float*) – This parameter specifies the minimum reference level for the custom self calibration range. This value is expressed in dBm.
- **maximum_reference_level** (*float*) – This parameter specifies the maximum reference level for the custom self calibration range. This value is expressed dBm.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

send_software_edge_advance_trigger()

Sends a trigger to the waiting device when you choose a software version of the Advance trigger. You can also use this method to override a hardware trigger.

This method returns an error if:

- You configure an invalid trigger.
- You have not previously called the RFmx Initiate method.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

send_software_edge_start_trigger()

Sends a trigger to the waiting device when you choose a software version of Start trigger. You can also use this method to override a hardware trigger.

This method returns an error if:

- You configure an invalid trigger.
- You have not previously called the RFmx Initiate method.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_advance_trigger_digital_edge_source(*selector_string*, *value*)

Sets the source terminal for the advance trigger.

This attribute is used only when the *ADVANCE_TRIGGER_TYPE* attribute is set to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the source terminal for the advance trigger.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_advance_trigger_export_output_terminal`(*selector_string*, *value*)

Sets the destination terminal for the exported advance trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the exported advance trigger.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_advance_trigger_type(*selector_string*, *value*)

Sets whether the advance trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No advance trigger is configured.
Digital Edge (1)	The advance trigger is not asserted until a digital edge is detected. The source of the digital edge is specified with the Advance Trigger Digital Edge Source attribute.
Software (3)	The advance trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*enums.AdvanceTriggerType, int*) – Specifies whether the advance trigger is a digital edge or a software trigger.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_amplitude_settling(*selector_string*, *value*)

Sets the amplitude settling accuracy value. This value is expressed in decibels. RFmx waits until the RF power attains the specified accuracy level after calling the RFmx Initiate method.

Any specified amplitude settling value that is above the acceptable minimum value is coerced down to the closest valid value.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported Devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the amplitude settling accuracy value. This value is expressed in decibels. RFmx waits until the RF power attains the specified accuracy level after calling the RFmx Initiate method.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_channel_coupling(*selector_string*, *value*)

Sets whether the RF IN connector is AC- or DC-coupled on the downconverter.

Nam (valu	Description
—	NoteFor the PXIe-5665/5668, this attribute must be set to AC Coupled when the DC block is present, and set to DC Coupled when the DC block is not present to ensure device specifications are met and proper calibration data is used. For more information about removing or attaching the DC block, refer to the PXIe-5665 Theory of Operation or the PXIe-5668 Theory of Operation topics in the NI RF Vector Signal Analyzers Help.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **AC Coupled**.

Valid values

- PXIe-5665 (3.6 GHz): AC Coupled DC Coupled
- PXIe-5665 (14 GHz): AC Coupled, DC Coupled
- PXIe-5668: AC Coupled

Supported devices: PXIe-5665/5668

Name (Value)	Description
AC Coupled (0)	Specifies that the RF input channel is AC-coupled. For low frequencies (<10 MHz), accuracy decreases because RFmxInstr does not calibrate the configuration.
DC Coupled (1)	Specifies that the RF input channel is DC-coupled. The RFmx driver enforces a minimum RF attenuation for device protection.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.ChannelCoupling, *int*) – Specifies whether the RF IN connector is AC- or DC-coupled on the downconverter.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_cleaner_spectrum`(*selector_string*, *value*)

Sets how to obtain the lowest noise floor or faster measurement speed.

Name (value)	Description
PXIe-5665	Sets the FFT Width attribute to take narrower bandwidth acquisitions and avoid digitizer spurs. Uses IF filters to reduce the noise floor for frequencies below 80 MHz.
PXIe-5644/5645/5646, PXIe-5840/5841/5842	Returns the best possible spectrum.
PXIe-5668	Returns the best possible spectrum. To provide the best spectrum measurement, the acquisition is reduced to 100 MHz segments for any center frequency.
Other devices	This attribute is ignored.

Note

Some measurements, such as Spurious Emissions enable the Cleaner Spectrum attribute by default. You can speed up those measurements by disabling the Cleaner Spectrum attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5665, PXIe-5668, PXIe-5644/5645/5646, PXIe-5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disable this attribute to get faster measurement speed.
Enabled (1)	Enable this attribute to get the lowest noise floor and avoid digitizer spurs.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.CleanerSpectrum, *int*) – Specifies how to obtain the lowest noise floor or faster measurement speed.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_common_mode_level`(*selector_string*, *value*)

Sets the common-mode level presented at each differential input terminal. The common-mode level shifts

both positive and negative terminals in the same direction. This must match the common-mode level of the device under test (DUT). This value is expressed in Volts.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5820

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the common-mode level presented at each differential input terminal. The common-mode level shifts both positive and negative terminals in the same direction. This must match the common-mode level of the device under test (DUT). This value is expressed in Volts.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_digital_gain(*selector_string*, *value*)

Sets the scaling factor applied to the time-domain voltage data in the digitizer. This value is expressed in dB. RFmx does not compensate for the specified digital gain.

You can use this attribute to account for external gain changes without changing the analog signal path.

> **Note**
>
> The PXIe-5644/5645/5646 applies this gain when the data is scaled. The raw data does not include this scaling on these devices.

Default Value : 0 dB

Supported Devices : PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the scaling factor applied to the time-domain voltage data in the digitizer. This value is expressed in dB. RFmx does not compensate for the specified digital gain.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_digitizer_dither_enabled(*selector_string*, *value*)

Sets whether dithering is enabled on the digitizer.

Dithering adds band-limited noise in the analog signal path to help reduce the quantization effects of the ADC and improve spectral performance. On the PXIe-5622, this out-of-band noise is added at low frequencies of up to approximately 12 MHz.

PXIe-5663/5663E/5665: When you enable dithering, the maximum signal level is reduced by up to 3 dB. This signal level reduction is accounted for in the nominal input ranges of the PXIe-5622. Therefore, you can overrange the input by up to 3 dB with dither disabled. For example, the +4 dBm input range can handle signal levels up to +7 dBm with dither disabled.

For wider bandwidth acquisitions, such as 40 MHz, disable dithering to eliminate residual leakage of the dither signal into the lower frequencies of the IF passband, which starts at 12.5 MHz and ends at 62.5 MHz. This leakage can slightly raise the noise floor in the lower frequencies, thus degrading the performance in high-sensitivity applications. When performing spectral measurements, this leakage can also appear as a wide, low-amplitude signal near the 12.5 MHz and 62.5 MHz frequencies. The width and amplitude of the signal depends on your resolution bandwidth and the type of time-domain window you apply to your FFT.

PXIe-5668: When you enable dithering, the maximum signal level is reduced by up to 2 dB. For the PXIe-5624, the maximum input power with dither off is 8 dBm and the maximum input power level with dither on is 6 dBm. When acquiring an 800 MHz bandwidth signal, the I/Q data contains the dither even if the dither signal is not in the displayed spectrum. The dither can affect actions like power level triggering.

Name (value)	Description
—	Note For the PXIe-5668, disabling dithering can negatively affect absolute amplitude accuracy.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Note

For PXIe-5820/5830/5831/5832/5840/5841/5842, only **Enabled** is supported.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.DigitizerDitherEnabled, *int*) – Specifies whether dithering is enabled on the digitizer.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_done_event_output_terminal`(*selector_string*, *value*)

Sets the destination terminal for the Done event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the Done event.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_downconverter_center_frequency(*selector_string*, *value*)

Enables in-band retuning and specifies the current frequency of the RF downconverter. This value is expressed in Hz.

After you set this attribute, the RF downconverter is locked to that frequency until the value is changed or the attribute is reset. Locking the downconverter to a fixed value allows frequencies within the instantaneous bandwidth of the downconverter to be measured without the overhead of retuning the LO and waiting for the LO to settle. This method is called in-band retuning and it has the highest benefit on analyzers that have larger LO settling times. After setting the downconverter center frequency, you can set the center frequency to the frequencies at which you want to take the measurements.

If you want to avoid the LO leakage or DC offset of analyzers that use a direct conversion architecture, it is more convenient to use the *DOWNCONVERTER_FREQUENCY_OFFSET* or *LO_LEAKAGE_AVOIDANCE_ENABLED* attributes.

If you set this attribute, any measurements outside the instantaneous bandwidth of the device are invalid. To disable in-band retuning, reset this attribute or call the *reset_to_default()* method.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is the carrier frequency or spectrum center frequency.

Valid Values: Any supported tuning frequency of the device.

> **Note**
>
> PXIe-5820: The only valid value for this attribute is 0 Hz.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Enables in-band retuning and specifies the current frequency of the RF downconverter. This value is expressed in Hz.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_downconverter_frequency_offset`(*selector_string*, *value*)

Sets an offset from the center frequency value for the downconverter. Use this attribute to offset the measurement away from the LO leakage or DC Offset of analyzers that use a direct conversion architecture. You must set this attribute to half the bandwidth or span of the measurement + guardband. The guardband is needed to ensure that the LO leakage is not inside the analog or digital filter rolloffs. This value is expressed in Hz.

NI recommends using the *LO_LEAKAGE_AVOIDANCE_ENABLED* attribute instead of the Downconverter Frequency Offset attribute. The LO Leakage Avoidance Enabled attribute automatically configures the Downconverter Frequency Offset attribute to an appropriate offset based on the bandwidth or span of the measurement.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values: For spectrum acquisition types, the RFmx driver automatically calculates the default value to avoid residual LO power. For I/Q acquisition types, the default value is 0 Hz. If the center frequency is set to a non-multiple of *LO_FREQUENCY_STEP_SIZE* attribute, this attribute is set to compensate for the difference.

The following valid values correspond to their respective devices:

Name (value)	Description
PXIe-5646	-100 MHz to +100 MHz
PXIe-5830/5831/5832/5840	-500 MHz to +500 MHz
PXIe-5841 (200MHz Bandwidth)	-100 MHz to +100 MHz
PXIe-5841 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (500MHz Bandwidth)	-250 MHz to +250 MHz
PXIe-5842 (1GHz Bandwidth)	-500 MHz to +500 MHz
PXIe-5842 (2GHz Bandwidth)	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using Standard personality	-1 GHz to +1 GHz
PXIe-5842 (4GHz Bandwidth) using the 4GHz Bandwidth personality	-2 GHz to +2 GHz
Other devices	-42 MHz to +42 MHz

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies an offset from the center frequency value for the downconverter. Use this attribute to offset the measurement away from the LO leakage or DC Offset of analyzers that use a direct conversion architecture. You must set this attribute to half the bandwidth or span of the measurement + guardband. The guardband is needed to ensure that the LO leakage is not inside the analog or digital filter rolloffs. This value is expressed in Hz.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_downconverter_preselector_enabled`(*selector_string*, *value*)

Sets whether the tunable preselector is enabled on the downconverter.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Disables the preselector.
Enabled (1)	The preselector is automatically enabled when it is in the signal path and is automatically disabled when it is not in the signal path. Use the Preselector Present attribute to determine if the downconverter has a preselector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.DownconverterPreselectorEnabled, *int*) – Specifies whether the tunable preselector is enabled on the downconverter.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_end_of_record_event_output_terminal`(*selector_string*, *value*)

Sets the destination terminal for the End of Record event.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the End of Record event.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_fft_width(*selector_string*, *value*)

Sets the FFT width of the device. The FFT width is the effective bandwidth of the signal path during each signal acquisition.

The lower limit for all devices that support setting the FFT Width attribute is 7.325 kHz.

PXIe-5663/5663E: The FFT width upper limit for the PXIe-5663/5663E depends on the RF frequency and on the module revision of the PXIe-5601. For more information about determining which revision of the PXIe-5601 RF downconverter you have installed, refer to the Identifying Module Revision topic in the *NI RF Vector Signal Analyzers Help*.

Note

The maximum FFT width for your device is constrained to 50 MHz or 25 MHz, depending on the digitizer option you purchased.

Note

You can use the FFT Width attribute with in-band retuning. For more information about in-band retuning, refer to the *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

The RFmx driver treats the device instantaneous bandwidth as the effective real-time bandwidth of the signal path. The span specifies the frequency range of the computed spectrum. A signal analyzer can acquire a bandwidth only within the device instantaneous bandwidth. If the span you choose is greater than the device instantaneous bandwidth, the RFmx driver obtains multiple acquisitions and combines them into a single spectrum. By specifying the FFT width, you can control the specific bandwidth obtained in each signal acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the FFT width of the device. The FFT width is the effective bandwidth of the signal path during each signal acquisition.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_frequency_reference_exported_terminal`(*selector_string*, *value*)

Sets a comma-separated list of the terminals at which to export the frequency reference.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None ()	The Reference Clock is not exported. This value is not valid for the PXIe-5644/5645/5646.
RefOut (RefOut)	Export the clock on the REF IN/OUT terminal on the PXIe-5652, the REF OUT terminals on the PXIe-5653, or the REF OUT terminal on the PXIe-5694, PXIe-5644/5645/5646, or PXIe-5820/5830/5831/5832/5840/5841/5860.
RefOut2 (RefOut2)	Export the clock on the REF OUT2 terminal on the PXIe-5652. This value is valid only for the PXIe-5663E.
ClkOut (ClkOut)	Export the Reference Clock on the CLK OUT terminal on the Digitizer. This value is not valid for the PXIe-5644/5645/5646 or PXIe-5820/5830/5831/5832/5840/5841/5842/5860.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies a comma-separated list of the terminals at which to export the frequency reference.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_frequency_reference_frequency`(*selector_string*, *value*)

Sets the Reference Clock rate, when the *FREQUENCY_REFERENCE_SOURCE* attribute is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 10 MHz.

Valid values

Name (value)	Description
PXIe-5644/5645/5646, PXIe-5663/5663E, PXIe-5820/5830/5831/5832/5840/5841/5842	10 MHz
PXIe-5665/5668	5 MHz to 100 MHz (inclusive), in increments of 1 MHz
PXIe-5860	10 MHz, 100 MHz

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the Reference Clock rate, when the *FREQUENCY_REFERENCE_SOURCE* attribute is set to **ClkIn** or **RefIn**. This value is expressed in Hz.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_frequency_reference_source(*selector_string*, *value*)

Sets the frequency reference source.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

All other devices default value is **OnboardClock**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value	Description
Onboard-Clock (Onboard-Clock)	PXIe-5663/5663E: The RFmx driver locks the PXIe-5663/5663E to the PXIe-5652 LO source onboard clock. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. On versions of the PXIe-5663/5663E that lack a REF OUT2 connector on the PXIe-5652, connect the REF IN/OUT connector on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: The RFmx driver locks the PXIe-5665 to the PXIe-5653 LO source onboard clock. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN terminal. PXIe-5668: Lock the PXIe-5668 to the PXIe-5653 LO SOURCE onboard clock. Connect the LO2 OUT connector on the PXIe-5606 to the CLK IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to its onboard clock. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. PXIe-5842: Lock to the associated PXIe-5655 onboard clock. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860: Lock to the PXIe-5860 onboard clock.
RefIn (RefIn)	PXIe-5663/5663E: Connect the external signal to the PXIe-5652 REF IN/OUT connector. Connect the REF OUT2 connector (if it exists) on the PXIe-5652 to the PXIe-5622 CLK IN terminal. PXIe-5665: Connect the external signal to the PXIe-5653 REF IN connector. Connect the 100 MHz REF OUT terminal on the PXIe-5653 to the PXIe-5622 CLK IN connector. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5668: Connect the external signal to the PXIe-5653 REF IN connector. Connect the LO2 OUT on the PXIe-5606 to the CLK IN connector on the PXIe-5622. If your external clock signal frequency is set to a frequency other than 10 MHz, set the Frequency Reference Frequency attribute according to the frequency of your external clock signal. PXIe-5644/5645/5646, PXIe-5820/5840/5841/5842/5860: The RFmx driver locks the device to the signal at the external REF IN connector. PXIe-5830/5831/5832: For PXIe-5830, connect the PXIe-5820 REF IN connector to the PXIe-3621 REF OUT connector. For PXIe-5831, connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. For PXIe-5832, connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. For PXIe-5830, lock the external signal to the PXIe-3621 REF IN connector. For PXIe-5831, lock the external signal to the PXIe-3622 REF IN connector. For PXIe-5832, lock the external signal to the PXIe-3623 REF IN connector. PXIe-5831 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3622 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3622 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5832 with PXIe-5653: Connect the PXIe-5820 REF IN connector to the PXIe-3623 REF OUT connector. Connect the PXIe-5653 REF OUT (10 MHz) connector to the PXIe-3623 REF IN connector. Lock the external signal to the PXIe-5653 REF IN connector. PXIe-5842: Lock to the signal at the REF IN connector on the associated PXIe-5655. Cables between modules are required as shown in the Getting Started Guide for the instrument. PXIe-5860:Lock to the signal at the REF IN connector on the PXIe-5860.
PXI_C (PXI_C	PXIe-5668: Lock the PXIe-5653 to the PXI backplane clock. Connect the PXIe-5606 LO2 OUT to the LO2 IN connector on the PXIe-5624. PXIe-5644/5645/5646, PXIe-5663/5663E/5665, and PXIe-5820/5840/5841/5860: The RFmx driver locks the device to the PXI backplane clock. PXIe-5830/5831/5832 with PXIe-5653/5841 with PXIe-5655, PXIe-5842/5860: The RFmx

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the frequency reference source.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_frequency_settling(*selector_string*, *value*)

Sets the value used for LO frequency settling.

Specify the units and interpretation for this scalar value using the *FREQUENCY_SETTLING_UNITS* attribute.

Valid values

Frequency Settling Units Property Value	PXIe-5663/566	PXIe-5665/566	PXIe-5644/5	PXIe-5830/5831/5832/5840/5841/5 PXIe-5831 with PXIe-5653 (using PXIe-3622 LO), PXIe-5832 with PXIe-5653 (using PXIe-3623 LO)	PXIe-5831 with PXIe-5653 (using PXIe-5653 LO) and PXIe-5832 with PXIe-5653 (using PXIe-5653 LO)
Seconds After Lock	2 µs to 80 ms, resolution of approximately 2 µs	4 µs to 80 ms, resolution of approximately 4 µs	1 µs to 65 ms, resolution of 1 µs	1 µs to 10s, resolution of 1 µs	4 µs to 80 ms, resolution of approximately 4 µs
Seconds After I/O	0 µs to 80 ms, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs	1 µs to 65 ms, resolution of 1 µs	0 µs to 10s, resolution of 1 µs	0 µs to 80 ms, resolution of 1 µs
PPM	1.0, 0.1, 0.01	1.0, 0.1, 0.01, 0.001	1.0, 0.1, 0.01	1.0 to 0.01	1.0 to 0.01

Name (value)	Description
—	Note This attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.1.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the value used for LO frequency settling.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_frequency_settling_units(*selector_string*, *value*)

Sets the delay duration units and interpretation for LO settling.

Specify the actual settling value using the *FREQUENCY_SETTLING* attribute.

Name (value)	Description
—	Note The Frequency Settling Units attribute is not supported if you are using an external LO.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **PPM**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
PPM (0)	Specifies the frequency settling in parts per million (ppm).
Seconds After Lock (1)	Specifies the frequency settling in time after lock (seconds).
Seconds After I/O (2)	Specifies the frequency settling in time after I/O (seconds).

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.FrequencySettlingUnits, *int*) – Specifies the delay duration units and interpretation for LO settling.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_if_filter_bandwidth(*selector_string*, *value*)

Sets the IF filter path bandwidth for your device configuration.

Name (value)	Description
—	Note For composite devices, such as the PXIe-5665/5668, the IF filter path bandwidth includes all IF filters across the component modules of a composite device.

RFmx chooses an appropriate IF filter as default IF Filter based on measurement configuration, center frequency, cleaner spectrum and downconverter preselector.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Supported devices: PXIe-5665/5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the IF filter path bandwidth for your device configuration.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_if_output_power_level_offset(*selector_string*, *value*)

Sets the power offset by which to adjust the default IF output power level. This value is expressed in dB.

This attribute does not depend on absolute IF output power levels; therefore, you can use this attribute to adjust the IF output power level on all RFmx-supported devices without knowing the exact default value. Use this attribute to increase or decrease the nominal output level to achieve better measurement results.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5663/5663E/5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the power offset by which to adjust the default IF output power level. This value is expressed in dB.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_input_isolation_enabled(*selector_string*, *value*)

Sets whether input isolation is enabled.

Enabling this attribute isolates the input signal at the RF IN connector on the RF downconverter from the rest of the RF downconverter signal path. Disabling this attribute reintegrates the input signal into the RF downconverter signal path.

> **Note**
>
> If you enable input isolation for your device, the device impedance is changed from the characteristic 50-ohm impedance. A change in the device impedance may increase the VSWR value higher than the device specifications.

For PXIe-5830/5831/5832, input isolation is supported for all available ports for your hardware configuration.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.InputIsolationEnabled, *int*) – Specifies whether input isolation is enabled.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo2_export_enabled(*selector_string*, *value*)

Sets whether to enable the LO2 OUT terminals in the installed devices.

Set this attribute to **Enabled** to export the 4 GHz LO signal from the LO2 IN terminal to the LO2 OUT terminal. You can also export the LO2 signal by setting the *LO_EXPORT_ENABLED* attribute to TRUE.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Disabled**.

Supported Devices: PXIe-5665/5668

Name (Value)	Description
Disabled (0)	Disables the LO2 OUT terminals.
Enabled (1)	Enables the LO2 OUT terminals.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.LO2ExportEnabled, *int*) – Specifies whether to enable the LO2 OUT terminals in the installed devices.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo_export_enabled(*selector_string*, *value*)

Sets whether to enable the LO OUT terminals on the installed devices.

Name (value)	Description
TRUE	Enables the LO OUT terminals.
FALSE	Disables the LO OUT terminals

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value:

- PXIe-5663/5663E: TRUE
- PXIe-5644/5645/5646, PXIe-5665/5668, PXIe-5830/5831/5832/5840/5841/5842: FALSE

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*bool*) – Specifies whether to enable the LO OUT terminals on the installed devices.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_lo_frequency`(*selector_string*, *value*)

Sets the LO signal frequency for the configured center frequency. This value is expressed in Hz.

If you are using the vector signal analyzer with an external LO, use this attribute to specify the LO frequency that the external LO source passes into the LO IN or LO1 IN connector on the RF downconverter front panel. If you are using an external LO, reading the value of this attribute after configuring the rest of the parameters returns the LO frequency needed by the device.

You can set this attribute to the actual LO frequency because RFmx corrects for any difference between expected and actual LO frequencies.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector Strings topic for information about the string syntax.

The default value is 0.

Supported Devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the LO signal frequency for the configured center frequency. This value is expressed in Hz.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_lo_frequency_step_size`(*selector_string*, *value*)

Sets the step size for tuning the LO phase-locked loop (PLL).

You can only tune the LO frequency in multiples of the LO Frequency Step Size attribute. Therefore, the LO frequency can be offset from the requested center frequency by as much as half of the LO Frequency Step Size attribute. This offset is corrected by digitally frequency shifting the LO frequency to the value requested in *DOWNCONVERTER_CENTER_FREQUENCY* attribute.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if PXIe-5653 is used as the LO source.

The valid values for this attribute depend on the *LO_PLL_FRACTIONAL_MODE* attribute.

PXIe-5644/5645/5646: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value.

PXIe-5840: If you set the LO PLL Fractional Mode attribute to **Disabled**, the specified value is coerced to the nearest valid value that is less than or equal to the desired step size.

LO PLL Fractional Mode Enabled Property Setting	LO Frequency Step Size Property Valid Values on PXIe-5644/5645	LO Frequency Step Size Property Valid Values on PXIe-5646	LO Frequency Step Size Property Valid Values on PXIe-5840/5841	LO Frequency Step Size Property Valid Values on PXIe-5830/5831/583	LO Frequency Step Size Property Valid Values on PXIe-5841 with PXIe-5655, PXIe-5842*
Enabled	50 kHz to 24 MHz	50 kHz to 25 MHz	50 kHz to 100 MHz	LO1: 8 Hz to 400 MHz LO2: 4 kHz to 400 MHz	1 nHz to 50 MHz
Disabled	4 MHz, 5 MHz, 6 MHz, 12 MHz, 24 MHz	2 MHz, 5 MHz, 10 MHz, 25 MHz	1 MHz, 5 MHz, 10 MHz, 25 MHz, 50 MHz, 100 MHz	LO1: – LO2: –	1 nHz to 50 MHz

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5644/5645/5646	200 kHz
PXIe-5830	2 MHz
PXIe-5831/5832 (RF port)	8 MHz
PXIe-5831/5832 (IF port)	2 MHz, 4 MHz
PXIe-5840/5841	500 kHz
PXIe-5842	1 Hz

> **Note**
>
> The default value for PXIe-5831/5832 depends on the frequency range of the selected port for your instrument configuration. Use *get_available_ports()* method to get the valid port names.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the step size for tuning the LO phase-locked loop (PLL).

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo_in_power(*selector_string*, *value*)

Sets the power level expected at the LO IN terminal when the *LO_SOURCE* attribute is set to **LO_In**. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> For PXIe-5644/5645/5646, this attribute is always read-only.

The default value is 0.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the power level expected at the LO IN terminal when the *LO_SOURCE* attribute is set to **LO_In**. This value is expressed in dBm.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_injection_side(*selector_string*, *value*)

Sets the LO injection side.

Name (value)	Description
PXIe-5663/5(	For frequencies below 517.5 MHz or above 6.4125 GHz, the LO injection side is fixed, and the RFmx driver returns an error if you specify an incorrect value. If you do not configure this attribute, the RFmx driver selects the default LO injection side based on the downconverter center frequency. Reset this attribute to return to automatic behavior.
PXIe-5665 (3.6 GHz)	Setting this attribute to Low Side is not supported for this device.
PXIe-5665 (14 GHz)	Setting this attribute to Low Side is supported for this device for frequencies greater than 4 GHz, but this configuration is not calibrated, and device specifications are not guaranteed.
PXIe-5668	Setting this attribute to Low Side is supported for some frequencies in high band, varying by the final IF frequency. This configuration is not calibrated and device specifications are not guaranteed.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

** Default value:**

- PXIe-5663/5663E (frequencies < 3.0 GHz): **High Side**
- PXIe-5663/5663E (frequencies >= 3.0 GHz): **Low Side**
- PXIe-5665/5668: **High Side**

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
High Side (0)	Configures the LO signal that the device generates at a frequency higher than the RF signal. This LO frequency is given by the following formula: *fLO = fRF + fIF *
Low Side (1)	Configures the LO signal that the device generates at a frequency lower than the RF signal. This LO frequency is given by the following formula: *fLO = fRF – fIF *

Parameters

- **selector_string** (*string*) – Pass an empty string.

- **value** (enums.LOInjectionSide, *int*) – Specifies the LO injection side.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_leakage_avoidance_enabled(*selector_string*, *value*)

Sets whether to reduce the effects of the instrument leakage by placing the LO outside the band of acquisition.

This attribute is ignored if:

- the bandwidth required by the measurement is more than the available instrument bandwidth after offsetting the LO.
- you set the *DOWNCONVERTER_CENTER_FREQUENCY* or *DOWNCONVERTER_FREQUENCY_OFFSET* attributes.

Note

When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set this attribute to **False** when the *LO_SOURCE* attribute is set to **Automatic_SG_SA_Shared**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5830/5831/5832/5840/5841/5842 is **True**, else the default value is **False**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
False (0)	RFmx does not modify the Downconverter Frequency Offset attribute.
True (1)	RFmx calculates the required LO offset based on the measurement configuration and appropriately sets the Downconverter Frequency Offset attribute.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.LOLeakageAvoidanceEnabled, *int*) – Specifies whether to reduce the effects of the instrument leakage by placing the LO outside the band of acquisition.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_out_power(*selector_string*, *value*)

Sets the power level of the signal at the LO OUT terminal when the *LO_EXPORT_ENABLED* attribute is set to TRUE. This value is expressed in dBm.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Supported devices: PXIe-5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the power level of the signal at the LO OUT terminal when the *LO_EXPORT_ENABLED* attribute is set to TRUE. This value is expressed in dBm.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo_pll_fractional_mode(*selector_string*, *value*)

Sets whether to use fractional mode for the LO phase-locked loop (PLL).

Fractional mode provides a finer frequency step resolution, but may result in non harmonic spurs. Refer to the specifications document of your device for more information about fractional mode and non harmonic spurs.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> The LO PLL Fractional Mode attribute is applicable only when using the internal LO.

> **Note**
>
> For PXIe-5831 with PXIe-5653, PXIe-5832 with PXIe-5653, this attribute is ignored if the PXIe-5653 is used as the LO source.

The default value is **Enabled**.

Supported devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Disabled (0)	Indicates that the attribute is disabled.
Enabled (1)	Indicates that the attribute is enabled.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.LOPllFractionalMode, *int*) – Specifies whether to use fractional mode for the LO phase-locked loop (PLL).

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo_sharing_mode(*selector_string*, *value*)

Sets the RFmx session with the respective LO sharing mode.

The following figures illustrate different connection configuration topologies for different LO Sharing modes.

You must set the *NUMBER_OF_LO_SHARING_GROUPS* attribute to 1 for the following LO connection configurations.

You must set the Num LO Sharing Groups attribute to 2 for the following LO connection configurations.

The default value is **Disabled**.

Name (Value)	Description
Disabled (0)	LO Sharing is disabled.
External Star (3)	The LO connection configuration is configured as External Star.
External Daisy Chain (4)	The LO connection configuration is configured as External Daisy Chain.
Splitter and Daisy Chain (5)	The LO connection configuration is configured as Splitter and Daisy Chain. With this option, the only allowed value for the Number of LO Sharing Groups attribute is 1.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.LOSharingMode, *int*) – Specifies the RFmx session with the respective LO sharing mode.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_lo_source(*selector_string*, *value*)

Sets the local oscillator (LO) signal source used to downconvert the RF input signal.

If this attribute is set to "" (empty string), RFmx uses the internal LO source. For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

If no signal downconversion is required, this attribute is ignored.

Note

This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Onboard**.

Supported Devices: PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
None (None)	Specifies that no LO source is required to downconvert the RF input signal.
Onboard (Onboard)	Specifies that the onboard synthesizer is used to generate the LO signal that downconverts the RF input signal.
LO_In (LO_In)	Specifies that the LO source used to downconvert the RF input signal is connected to the LO IN connector on the front panel.
Secondary (Secondary)	Specifies that the LO source uses the PXIe-5830/5831/5832/5840 internal LO. This value is valid on only the PXIe-5840 with PXIe-5653, PXIe-5831 with PXIe-5653 (LO1 stage only), or PXIe-5832 with PXIe-5653 (LO1 stage only).
SG_SA_Shared (SG_SA_Shared)	Specifies that the internal LO can be shared between RFmx and RFSG sessions. RFmx selects an internal synthesizer and the synthesizer signal is switched to both the RX and TX mixers. This value is valid only on PXIe-5830/5831/5832/5841/5842.

Automatic_SG_SA_Sha (Automatic_SG_SA_Sha	Specifies whether RFmx automatically configures the signal analyzer to use the LO utilized by the signal generator on the same vector signal transceiver (VST) based on the configured measurements.
	When using instruments that do not have LOs with excellent phase noise and to minimize the contribution of the instrument's phase noise affecting your measurements, NI recommends to share the LO between the signal generator (SG) and the signal analyzer (SA).
	This value is recommended in test setups that use a VST with NI-RFSG to generate a signal at the DUT's input and RFmx to measure the signal at the DUT's output.
	This value automatically:
	• determines whether the SG LO can be shared with SA based on the test instrument used, selected measurement, and the measurement settings.
	• configures instrument specific attributes on SA to share the LO between the generator and analyzer, whenever possible.
	To enable automatically sharing SG LO with SA, you must first setup the required device specific physical connections mentioned below and then follow the steps in the recommended order.
	PXIe-5840/5841: SG LO is shared with SA via an external path. Hence, you must connect RF Out LO Out to RF In LO In using a cable.
	PXIe-5841 with PXIe-5655/5842/PXIe-5830/5831/5832: SG LO is shared with SA via an internal path. Hence, an external cable connection is not required.
	NI recommends the following order of steps:
	• Set LO Source attribute to Automatic SG SA Shared in NI-RFSG (or enable Automatic SG SA shared LO on NI-RFSG Playback Library).
	• Set LO Source attribute to **Automatic_SG_SA_Shared** in RFmx.
	• Configure any additional settings on RFSG and RFmx, including selecting waveforms.
	• Initiate RFSG.
	• Initiate RFmx.
	When using a DPD applied signal for performing measurements like ModAcc, PvT, or TXP, you must set the LO Leakage Avoidance Enabled attribute to **False** and LO Source attribute to **Automatic_SG_SA_Shared**.
	Refer to following methods for examples in RFmx WLAN and RFmx NR that show the behavior of Automatic SG SA Shared LO.
	`<LabVIEW directory>\examples\RFmx\WLAN\RFmxWLAN FEM Test with Automatic SG SA Shared LO.vi`
	`<LabVIEW directory>\examples\RFmx\NR\RFmxNR FEM Test with Automatic SG SA Shared LO.vi`
	This value is valid only on PXIe-5830/5831/5832/5840/5841/5842.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the local oscillator (LO) signal source used to downconvert the RF input signal.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_splitter_loss(*selector_string*, *value*)

Sets an array of the insertion losses inherent to the RF Splitter. This value is expressed in dB.

You must specify the frequencies at which the losses were measured using the *LO_SPLITTER_LOSS* attribute.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies an array of the insertion losses inherent to the RF Splitter. This value is expressed in dB.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_splitter_loss_frequency(*selector_string*, *value*)

Sets the frequencies corresponding to the insertion loss inherent to the RF Splitter, as specified by the *LO_SPLITTER_LOSS_FREQUENCY* attribute. This value is expressed in Hz.

You do not need to use a selector string to configure or read this attribute for the default signal instance. Refer to the Selector String topic for information about the string syntax for named signals.

The default value is an empty array.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the frequencies corresponding to the insertion loss inherent to the RF Splitter, as specified by the *LO_SPLITTER_LOSS_FREQUENCY* attribute. This value is expressed in Hz.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_lo_vco_frequency_step_size(*selector_string*, *value*)

Sets the step size for tuning the internal voltage-controlled oscillator (VCO) used to generate the LO signal. The valid values for LO1 include 1 Hz to 50 MHz and for LO2 include 1 Hz to 100 MHz.

> **Note**
>
> Do not set this attribute with the *LO_FREQUENCY_STEP_SIZE* attribute.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 1 MHz.

Supported devices: PXIe-5830/5831/5832

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the step size for tuning the internal voltage-controlled oscillator (VCO) used to generate the LO signal. The valid values for LO1 include 1 Hz to 50 MHz and for LO2 include 1 Hz to 100 MHz.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_load_options(*selector_string*, *value*)

Sets the configurations to skip while loading from a file using the *load_configurations()* method .

Name (value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session

The default value is an empty array.

Name (Value)	Description
Skip None (0)	RFmx loads all the configurations to the session.
Skip RFInstr (1)	RFmx skips loading the RFmxInstr configurations to the session.

Parameters

- **selector_string** (*string*) – Pass an empty string.

- **value** (enums.LoadOptions, *int*) – Specifies the configurations to skip while loading from a file using the *load_configurations()* method .

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_mechanical_attenuation_auto(*selector_string*, *value*)

Sets whether the RFmx driver chooses an attenuation setting based on the hardware settings.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E/5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured in the Mechanical Attenuation Value attribute.
True (1)	Specifies that the measurement computes the mechanical attenuation.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.MechanicalAttenuationAuto, *int*) – Specifies whether the RFmx driver chooses an attenuation setting based on the hardware settings.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_mechanical_attenuation_value(*selector_string*, *value*)

Sets the level of mechanical attenuation for the RF path. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *MECHANICAL_ATTENUATION_AUTO* attribute to **False**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Valid values

Name (value)	Description
PXIe-5663/5663E	0, 16
PXIe-5665 (3.6 GHz)	0, 10, 20, 30
PXIe-5665 (14 GHz), PXIe-5668	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75

Supported devices: PXIe-5663/5663E, PXIe-5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the level of mechanical attenuation for the RF path. This value is expressed in dB.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_mixer_level(*selector_string*, *value*)

Sets the mixer level. This value is expressed in dBm.

The mixer level represents the attenuation value to apply to the input RF signal as it reaches the first mixer in the signal chain. If you do not set this attribute, the RFmx driver automatically selects an optimal mixer level value based on the reference level.

If you set the *MIXER_LEVEL* and *MIXER_LEVEL_OFFSET* attributes at the same time, the RFmx driver returns an error.

This attribute is read-only for PXIe-5663/5663E devices.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values

Name (value)	Description
PXIe-5665/5668	-10
All other devices	N/A

The valid values for this attribute depend on your device configuration.

Supported devices: PXIe-5663/5663E/5665/5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the mixer level. This value is expressed in dBm.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_mixer_level_offset(*selector_string*, *value*)

Sets the number of dB by which to adjust the device mixer level.

Specifying a positive value for this attribute configures the device for moderate distortion and low noise, and specifying a negative value results in low distortion and higher noise. You cannot set the *MIXER_LEVEL_OFFSET* and *MIXER_LEVEL* attributes at the same time.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0. The default value specifies device settings that are the best compromise between distortion and noise.

Supported devices: PXIe-5663/5663E/5665/5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the number of dB by which to adjust the device mixer level.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_number_of_lo_sharing_groups(*selector_string*, *value*)

Sets the RFmx session with the number of LO sharing groups.

The default value is 1.

The valid values are 1 and 2.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*int*) – Specifies the RFmx session with the number of LO sharing groups.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_optimize_path_for_signal_bandwidth(*selector_string*, *value*)

Optimizes RF path for the signal bandwidth that is centered on the IQ carrier frequency.

You can disable this attribute to avoid changes to the RF path when changing the signal bandwidth.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Automatic**.

Supported devices: PXIe-5830/5831/5832/5841/5842

Name (Value)	Description
Disabled (0)	Disables the optimized path for signal bandwidth.
Enabled (1)	Enables the optimized path for signal bandwidth.
Automatic (2)	Automatically enables the optimized path based on other configurations.

Parameters

- **selector_string** (*string*) – Pass an empty string.

- **value** (enums.OptimizePathForSignalBandwidth, *int*) – Optimizes RF path for the signal bandwidth that is centered on the IQ carrier frequency.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_osp_delay_enabled(*selector_string*, *value*)

Sets whether to enable the digitizer OSP block to delay Reference Triggers, along with the data samples, moving through the OSP block.

If you set this attribute to **Disabled**, the Reference Triggers bypass the OSP block and are processed immediately.

Enabling this attribute requires the following equipment configurations:

- All digitizers being used must be the same model and hardware revision.
- All digitizers must use the same firmware.
- All digitizers must be configured with the same I/Q rate.
- All devices must use the same signal path.

For more information about the digitizer OSP block and Reference Triggers, refer to the following topics in the *NI High-Speed Digitizers Help*:

- PXIe-5622 Onboard Signal Processing (OSP)
- PXIe-5142 Onboard Signal Processing (OSP)
- PXIe-5622 Trigger Sources
- PXI-5142 Trigger Sources
- PXIe-5622 Block Diagram
- PXI-5142 Trigger Sources

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Enabled**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Disabled (0)	Disables the attribute.
Enabled (1)	Enables the attribute.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.OspDelayEnabled, *int*) – Specifies whether to enable the digitizer OSP block to delay Reference Triggers, along with the data samples, moving through the OSP block.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_overflow_error_reporting`(*selector_string*, *value*)

Configures error reporting for ADC and overflows occurred during onboard signal processing. Overflows lead to clipping of the waveform.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Warning**.

Supported devices: PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Warning (0)	RFmx returns a warning when an ADC or an onboard signal processing (OSP) overflow occurs.
Disabled (1)	RFmx does not return an error or a warning when an ADC or OSP overflow occurs.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.OverflowErrorReporting, *int*) – Configures error reporting for ADC and overflows occurred during onboard signal processing. Overflows lead to clipping of the waveform.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_phase_offset`(*selector_string*, *value*)

Sets the offset to apply to the initial I and Q phases.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values are -180 degrees to 180 degrees, inclusive.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the offset to apply to the initial I and Q phases.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_preamp_enabled(*selector_string*, *value*)

Sets whether the RF preamplifier is enabled in the system.

PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860: If you set this attribute to **Automatic**, RFmx selects the preamplifier state based on the value of the Reference Level attribute and the center frequency. For PXIe-5830/5831/5832, the value is not coerced.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842 is **Automatic**, else the default value is **Disabled**.

Name (Value	Description
Disabled (0)	Disables the RF preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Enabled (1)	Enables the RF preamplifier when it is in the signal path and disables it when it is not in the signal path. Only devices with an RF preamplifier on the downconverter and an RF preselector support this option. Use the RF Preamp Present attribute to determine whether the downconverter has a preamplifier. **Supported Devices:** PXIe-5663/5663E/5665/5668
Automatic (3)	Automatically enables the RF preamplifier based on the value of the reference level. **Supported Devices:** PXIe-5644/5645/5646, PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.PreampEnabled, *int*) – Specifies whether the RF preamplifier is enabled in the system.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_ready_for_advance_event_output_terminal(*selector_string*, *value*)

Sets the destination terminal for the Ready for Advance event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the Ready for Advance event.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_ready_for_reference_event_output_terminal(*selector_string*, *value*)

Sets the destination terminal for the Ready for Reference event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the Ready for Reference event.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_ready_for_start_event_output_terminal(*selector_string*, *value*)

Sets the destination terminal for the Ready for Start event.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on the PXIe-5652, and the REF OUT terminal on the PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the Ready for Start event.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_rf_attenuation_auto(*selector_string*, *value*)

Sets whether the RFmx driver computes the RF attenuation.

If you set this attribute to **True**, the RFmx driver chooses an attenuation setting based on the reference level configured on the personality.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **True**.

Supported devices: PXIe-5663/5663E, PXIe-5665/5668

Name (Value)	Description
False (0)	Specifies that the RFmx driver uses the value configured using RF Attenuation Value attribute.
True (1)	Specifies that the RFmx driver computes the RF attenuation.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.RFAttenuationAuto, *int*) – Specifies whether the RFmx driver computes the RF attenuation.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_rf_attenuation_step_size(*selector_string*, *value*)

Sets the step size for the RF attenuation level. This value is expressed in dB. The actual RF attenuation is coerced up to the next highest multiple of the specified step size. If the mechanical attenuators are not available to implement the coerced RF attenuation, the solid state attenuators are used.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0
PXIe-5603/5665 (3.6 GHz)	1.0
PXIe-5605/5665 (14 GHz), PXIe-5606/5668	5.0

Valid values:

Name (value)	Description
PXIe-5601/5663/5663E	0.0 to 93.0, continuous
PXIe-5603/5665 (3.6 GHz)	1.0 to 74.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (low band), PXIe-5606/5668 (low band)	1.0 to 106.0, in 1 dB steps
PXIe-5605/5665 (14 GHz) (high band), PXIe-5606/5668 (high band)	5.0 to 75.0, in 5 dB steps

Supported devices: PXIe-5663, PXIe-5665, PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the step size for the RF attenuation level. This value is expressed in dB. The actual RF attenuation is coerced up to the next highest multiple of the specified step size. If the mechanical attenuators are not available to implement the coerced RF attenuation, the solid state attenuators are used.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_rf_attenuation_value(*selector_string*, *value*)

Sets the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

The RFmx driver uses the value of this attribute as the attenuation setting when you set the *RF_ATTENUATION_AUTO* attribute to **False**.

Name (value)	Description
PXIe-5663/5663E	You can change the attenuation value to modify the amount of noise and distortion. Higher attenuation levels increase the noise level but decreases distortion; lower attenuation levels decrease the noise level but increases distortion.
PXIe-5603/5605/5	Refer to the PXIe-5665 or the PXIe-5668 RF Attenuation and Signal Levels topic in the NI RF Vector Signal Analyzers Help for more information about configuring attenuation.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The valid values for this attribute depend on the device configuration.

Supported devices: PXIe-5663/5663E/5603/5605/5665/5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the nominal attenuation setting for all attenuators before the first mixer in the RF signal chain. This value is expressed in dB.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_rf_highpass_filter_frequency(*selector_string*, *value*)

Sets the maximum corner frequency of the high pass filter in the RF signal path. The device uses the highest frequency high-pass filter option below or equal to the value you specify and returns a coerced value. Specifying a value of 0 disables high pass filtering silly.

For multispan acquisitions, the device uses the appropriate filter for each subspan during acquisition, depending on the details of your application and the value you specify. In multispan acquisition spectrum applications, this attribute returns the value you specified rather than a coerced value if multiple high-pass filters are used during the acquisition.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

The valid values range from 0 to 26.5.

Supported devices: PXIe-5668

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the maximum corner frequency of the high pass filter in the RF signal path. The device uses the highest frequency high-pass filter option below or equal to the value you specify and returns a coerced value. Specifying a value of 0 disables high pass filtering silly.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_self_calibration_validity_check(*selector_string*, *value*)

Sets whether the RFmx driver validates the self-calibration data.

You can specify the time interval required to perform the check using the *SELF_CALIBRATION_VALIDITY_CHECK_TIME_INTERVAL* attribute.

NI recommends to perform self-calibration using the *self_calibrate()* method when RFmx reports an invalid self-calibration data warning.

> **Note**
>
> The RFmx driver does not consider self-calibration range data during self calibration validity check.

The default value is **Off**.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

Name (Value)	Description
Off (0)	Indicates that RFmx does not check whether device self-calibration data is valid.
Enabled (1)	Indicates that RFmx checks whether device self-calibration data is valid and reports a warning from the RFmx Commit and RFmx Initiate methods when the data is invalid.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.SelfCalibrationValidityCheck, *int*) – Specifies whether the RFmx driver validates the self-calibration data.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_self_calibration_validity_check_time_interval(*selector_string*, *value*)

Sets the minimum time between two self calibration validity checks. This value is expressed in seconds.

When you call RFmx Commit or Initiate methods by enabling the *SELF_CALIBRATION_VALIDITY_CHECK* attribute, the RFmx driver checks if the amount of time specified by the Self Calibration Validity Check Time Interval attribute has elapsed before validating the calibration data.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5644/5645/5646, PXIe-5820/5830/5831/5832/5833/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the minimum time between two self calibration validity checks. This value is expressed in seconds.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_smu_channel(*selector_string*, *value*)

Sets the output channel to be used for noise figure (NF) measurement in RFmx.

The default value is 0.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the output channel to be used for noise figure (NF) measurement in RFmx.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_smu_resource_name(*selector_string*, *value*)

Sets the resource name assigned by Measurement and Automation Explorer (MAX) for NI Source Measure Units (SMU) which is used as the noise source power supply for Noise Figure (NF) measurement, for example, PXI1Slot3, where PXI1Slot3 is an instrument resource name. SMU Resource Name can also be a logical IVI name.

Supported devices: PXIe-4138, PXIe-4139, PXIe-4139 (40 W), and PXIe-4143 SMUs.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the resource name assigned by Measurement and Automation Explorer (MAX) for NI Source Measure Units (SMU) which is used as the noise source power supply for Noise Figure (NF) measurement, for example, PXI1Slot3, where PXI1Slot3 is an instrument resource name. SMU Resource Name can also be a logical IVI name.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_start_trigger_digital_edge(*selector_string*, *value*)

Sets the active edge for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Rising Edge**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Rising Edge (0)	The trigger asserts on the rising edge of the signal.
Falling Edge (1)	The trigger asserts on the falling edge of the signal.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.StartTriggerDigitalEdge, *int*) – Specifies the active edge for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_start_trigger_digital_edge_source(*selector_string*, *value*)

Sets the source terminal for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value of this attribute is "" (empty string).

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
PFI0 (PFI0)	The trigger is received on PFI 0. For the PXIe-5841 with PXIe-5655, the trigger is received on the PXIe-5841 PFI 0.
PFI1 (PFI1)	The trigger is received on PFI 1.
PXI_Trig0 (PXI_Trig0)	The trigger is received on PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	The trigger is received on PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	The trigger is received on PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	The trigger is received on PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	The trigger is received on PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	The trigger is received on PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	The trigger is received on PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	The trigger is received on PXI trigger line 7.
PXI_STAR (PXI_STAR)	The trigger is received on the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarB (PXIe_DStarB)	The trigger is received on the PXIe DStar B trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
TimerEvent (TimerEvent)	The trigger is received from the timer event. This value is valid only for PXIe-5820/5840/5841/5842/5860 and for digital edge advance triggers on PXIe-5663E/5665.
DIO/PFI0 (DIO/PFI0)	The trigger is received on PFI 0 of the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	The trigger is received on PFI 1 of the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	The trigger is received on PFI 2 of the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	The trigger is received on PFI 3 of the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	The trigger is received on PFI 4 of the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	The trigger is received on PFI 5 of the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	The trigger is received on PFI 6 of the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	The trigger is received on PFI 7 of the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the source terminal for the start trigger. This attribute is used only when you set the *START_TRIGGER_TYPE* attribute to **Digital Edge**.

Returns

Returns the status code of this method. The status code either indicates success or describes

a warning condition.

Return type
int

`set_start_trigger_export_output_terminal`(*selector_string*, *value*)

Sets the destination terminal for the exported start trigger.

You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists only on PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the exported start trigger.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_start_trigger_type(*selector_string*, *value*)

Sets whether the start trigger is a digital edge or a software trigger.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **None**.

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
None (0)	No start trigger is configured.
Digital Edge (1)	The start trigger is not asserted until a digital edge is detected. The source of the digital edge is specified by the Start Trigger Digital Edge Source attribute.
Software (3)	The start trigger is not asserted until a software trigger occurs. You can assert the software trigger by calling the RFmxInstr Send Software Edge Trigger method.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.StartTriggerType, *int*) – Specifies whether the start trigger is a digital edge or a software trigger.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_subspan_overlap(*selector_string*, *value*)

Use subspan overlap process to eliminate or reduce analyzer spurs. To enable this feature, specify a non-zero percentage overlap between consecutive subspans in a spectrum acquisition.

If a value greater than 0 is specified, then for each spectral line in the resulting spectrum, the driver acquires data twice with slightly different hardware settings, so that the analyzer spurs, if any, are present at different frequencies in the two acquisitions. Typically, LO frequency is shifted between the acquisitions causing analyzer spurs that are relative to the LO frequency, to move from one frequency to another. Those spurs, which are present in only one of the acquisitions for each spectral line, get removed.

The subspan overlap feature will not remove any spurs from the Device Under Test or modify the signal being measured; unlike the analyzer spurs, the spurs in the signal being measured stay at a constant frequency in the two acquisitions.

> **Note**
>
> Subspan overlap process effectively is performing minimum averaging, which might reduce the measured noise floor level. RFmx Spectrum Averaging can be enabled to minimize the effect of subspan overlap on the noise floor.

> **Note**
>
> RFmx may apply further shifts to the specified value to accommodate fixed-frequency edges of components such as preselectors.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 0.

Valid values

Name (value)	Description
PXIe-5820/5830/5831/5832/5840/5841/5860	0
PXIe-5842	0, 50
PXIe-5665/5668	0 to <100

Supported devices: PXIe-5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Use subspan overlap process to eliminate or reduce analyzer spurs. To enable this feature, specify a non-zero percentage overlap between consecutive subspans in a spectrum acquisition.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_temperature_read_interval(*selector_string*, *value*)

Sets the minimum time difference between temperature sensor readings. This value is expressed in seconds.

When you call the RFmx Initiate method, RFmx checks if the amount of time specified by this attribute has elapsed before reading the hardware temperature.

> **Note**
>
> RFmx ignores Temperature Read Interval attribute if you read the *DOWNCONVERTER_GAIN* attribute.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is 30 seconds.

Supported devices: PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the minimum time difference between temperature sensor readings. This value is expressed in seconds.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

set_thermal_correction_headroom_range(*selector_string*, *value*)

Sets the expected thermal operating range of the instrument from the self-calibration temperature returned from the *DEVICE_TEMPERATURE* attribute. This value is expressed in degree Celsius.

For example, if this attribute is set to 5.0, and the device is self-calibrated at 35 degrees Celsius, then you can expect to run the device from 30 degrees Celsius to 40 degrees Celsius with corrected accuracy and no overflows. Setting this attribute with a smaller value can result in improved dynamic range, but you must ensure thermal stability while the instrument is running. Operating the instrument outside of the specified range may cause degraded performance and ADC or DSP overflows.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	5
PXIe-5840/5841	10

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the expected thermal operating range of the instrument from the self-calibration temperature returned from the *DEVICE_TEMPERATURE* attribute. This value is expressed in degree Celsius.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_thermal_correction_temperature_resolution`(*selector_string*, *value*)

Sets the temperature change required before RFmx recalculates the thermal correction settings when entering the running state. This value is expressed in degree Celsius.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

Default value

Name (value)	Description
PXIe-5830/5831/5832/5842/5860	0.2
PXIe-5840/5841	1.0

Supported devices: PXIe-5830/5831/5832/5840/5841/5842/5860

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*float*) – Specifies the temperature change required before RFmx recalculates the thermal correction settings when entering the running state. This value is expressed in degree Celsius.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

`set_trigger_export_output_terminal`(*selector_string*, *value*)

Sets the destination terminal for the exported Reference Trigger. You can also choose not to export any signal.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read that signal instance. Refer to the Selector String topic for information about the string syntax.

The default value is **Do not export signal**.

Supported devices: PXIe-5644/5645/5646 and PXIe-5663/5663E/5665/5668, PXIe-5820/5830/5831/5832/5840/5841/5842/5860

Name (Value)	Description
Do not export signal ()	Does not export the signal.
ClkOut (ClkOut)	Exports the signal to the CLK OUT connector on the PXIe-5622/5624 front panel.
RefOut (RefOut)	Exports the signal to the REF IN/OUT terminal on PXIe-5652, and the REF OUT terminal on PXIe-5644/5645/5646 and PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
RefOut2 (RefOut2)	Exports the signal to the REF OUT2 terminal on the LO. This connector exists on only PXIe-5652.
PFI0 (PFI0)	Exports the signal to the PFI 0 connector. For the PXIe-5841 with PXIe-5655, the signal is exported to the PXIe-5841 PFI 0.
PFI1 (PFI1)	Exports the signal to the PFI 1 connector on PXIe-5142 and PXIe-5622.
PXI_Trig0 (PXI_Trig0)	Exports the signal to the PXI trigger line 0.
PXI_Trig1 (PXI_Trig1)	Exports the signal to the PXI trigger line 1.
PXI_Trig2 (PXI_Trig2)	Exports the signal to the PXI trigger line 2.
PXI_Trig3 (PXI_Trig3)	Exports the signal to the PXI trigger line 3.
PXI_Trig4 (PXI_Trig4)	Exports the signal to the PXI trigger line 4.
PXI_Trig5 (PXI_Trig5)	Exports the signal to the PXI trigger line 5.
PXI_Trig6 (PXI_Trig6)	Exports the signal to the PXI trigger line 6.
PXI_Trig7 (PXI_Trig7)	Exports the signal to the PXI trigger line 7.
PXI_STAR (PXI_STAR)	Exports the signal to the PXI star trigger line. This value is not valid for the PXIe-5644/5645/5646.
PXIe_DStarC (PXIe_DStarC)	Exports the signal to the PXIe DStar C trigger line. This value is valid on only for PXIe-5820/5830/5831/5832/5840/5841/5842/5860.
DIO/PFI0 (DIO/PFI0)	Exports the signal to the PFI 0 on the DIO front panel connector.
DIO/PFI1 (DIO/PFI1)	Exports the signal to the PFI 1 on the DIO front panel connector.
DIO/PFI2 (DIO/PFI2)	Exports the signal to the PFI 2 on the DIO front panel connector.
DIO/PFI3 (DIO/PFI3)	Exports the signal to the PFI 3 on the DIO front panel connector.
DIO/PFI4 (DIO/PFI4)	Exports the signal to the PFI 4 on the DIO front panel connector.
DIO/PFI5 (DIO/PFI5)	Exports the signal to the PFI 5 on the DIO front panel connector.
DIO/PFI6 (DIO/PFI6)	Exports the signal to the PFI 6 on the DIO front panel connector.
DIO/PFI7 (DIO/PFI7)	Exports the signal to the PFI 7 on the DIO front panel connector.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (*string*) – Specifies the destination terminal for the exported Reference Trigger. You can also choose not to export any signal.

Returns
Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type
int

set_tuning_speed(*selector_string*, *value*)

Makes tradeoffs between tuning speed and phase noise.

> **Note**
>
> This attribute is not supported if you are using an external LO.

For PXIe-5830/5831/5832, if you want to configure or read on LO1 or LO2 channel, use "lo1" or "lo2" as part of the selector string. You do not need to use a selector string or use "lo1, lo2" as part of the selector string if you want to configure this attribute for both channels. You can also use *build_lo_string()* utility function to create the LO String. For all other devices, lo channel string is not allowed.

> **Note**
>
> This attribute is not supported on a MIMO session.

You do not need to use a selector string if you want to configure this attribute for all signal instances. Specify the signal name in the selector string if you want to configure or read on that signal instance. Refer to the Selector String topic for information about the string syntax.

> **Note**
>
> The PXIe-5830/5831/5832/5840/5841/5842 supports only **Medium** for this attribute.

Default value: **Normal** for PXIe-5663/5663E/5665/5668, **Medium** for PXIe-5644/5645/5646 and PXIe-5830/5831/5832/5840/5841/5842

Supported devices: PXIe-5644/5645/5646, PXIe-5663/5663E/5665/5668, PXIe-5830/5831/5832/5840/5841/5842

Name (Value)	Description
Normal (0)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an underdamped response. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a narrow loop bandwidth.
Mediun (1)	Specifies that the RF downconverter module uses a medium loop bandwidth. This value is not supported on PXIe-5663/5663E/5665/5668 devices.
Fast (2)	PXIe-5665/5668: Adjusts the YIG main coil on the LO for an overdamped response. Setting this attribute to **Fast** allows the frequency to settle significantly faster for some frequency transitions at the expense of increased phase noise. PXIe-5663/5663E/5644/5645/5646: Specifies that the RF downconverter module uses a wide loop bandwidth.

Parameters

- **selector_string** (*string*) – Pass an empty string.
- **value** (enums.TuningSpeed, *int*) – Makes tradeoffs between tuning speed and phase noise.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

specan_clear_noise_calibration_database(*selector_string*)

Clears the noise calibration database used for noise compensation.

Parameters

selector_string (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

wait_for_acquisition_complete(*timeout*)

Waits and blocks the data flow until the acquisition is complete. This method is typically called after a specific initiate method.

Parameters

timeout (*float*) – This parameter specifies the time to wait for an ongoing acquisition to complete before returning a timeout error. A value of -1 specifies that the method waits indefinitely for acquisition to complete. This value is expressed in seconds. The default value is 10.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

wlan_ofdmmodacc_clear_noise_calibration_database(*selector_string*)

Clears the noise calibration database used for noise compensation.

Parameters

selector_string (*string*) – Pass an empty string. The signal name that is passed when creating the signal configuration is used.

Returns

Returns the status code of this method. The status code either indicates success or describes a warning condition.

Return type

int

CHAPTER

SIX

ADDITIONAL DOCUMENTATION

Refer to the NI-RFmx User Manual for an overview of NI-RFmx, system requirements, troubleshooting, key concepts, etc.

CHAPTER

SEVEN

LICENSE

This project is licensed under the MIT License. While the source code is not publicly released, the license permits binary distribution with attribution.

Note: This Python driver depends on several third-party components that are subject to separate commercial licenses. Users are responsible for ensuring they have the appropriate rights and licenses to use those dependencies in their environments.

CHAPTER

EIGHT

GRPC FEATURES

For driver APIs that support it, passing a GrpcSessionOptions instance as a parameter to `nirfmxinstr.Session.__init__()` is subject to the NI General Purpose EULA.

CHAPTER

NINE

SSL/TLS SUPPORT

The server supports both server-side TLS and mutual TLS. Security configuration is accomplished by setting the *server_cert*, *server_key* and *root_cert* values in the server's configuration file. The server expects the certificate files specified in the configuration file to exist in a *certs* folder that is located in the same directory as the configuration file being used by the server. For more detailed information on SSL/TLS support refer to the [Server Security Support wiki page](https://github.com/ni/grpc-device/wiki/Server-Security-Support).

CHAPTER

TEN

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

n

INDEX

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W