NI WLAN Generation Toolkit

2023-05-16

Contents

NI WLAN Generation Toolkit Help. 3
NI WLAN Generation Toolkit Examples. 5
Working with Application Manifest. 5
Concepts. 5
Setting Up PXI Trigger Lines for Multiple VSTs. 5
Configuring Active Channels (LabVIEW). 7
Configuring Active Channels (LabWindows/CVI). 13
RFSG Database. 21
Windowing. 21
Generating the Trigger Frame. 25
Configuring LO Connection in WLAN Generation Soft Front Panel. 29
Programming. 38
LabVIEW Reference. 38
C Reference. 336
WLAN Calibration Utility. 701
Getting Started. 701
Calibration of NI VST Devices. 701
LO Alignment. 704
Noise Floor Measurement. 704
Validate Hardware. 706
Specify LO Connections. 707
Command Buttons. 709

May 2020, 373725L-01

The NI WLAN Generation Toolkit extends LabVIEW and LabWindows™/CVI™ functionality and provides tools and functions to generate signals to test wireless local area network (WLAN) transceiver signals that conform to the following standards:

- **IEEE Standard 802.11a-1999**
- **IEEE Standard 802.11b-1999**
- **IEEE Standard 802.11g-2003**
- **IEEE Standard 802.11j-2004**
- **IEEE Standard 802.11n-2009**
- **IEEE Standard 802.11p-2010**
- **IEEE Standard 802.11ac-2013**
- **IEEE P802.11ah/D1.3**
- **IEEE Standard 802.11af-2013**
- **IEEE P802.11ax/D8.0**
- **IEEE P802.11be/D0.2**

Refer to the IEEE website to access these standards. This help file assumes you are familiar with the IEEE standards.

The following table lists the NI WLAN Generation Toolkit licenses and the type of signals you can test using the VIs, properties, functions, and attributes available in the toolkit.

License	Signals Conform to these IEEE Standards
NI WLAN Generation Toolkit for IEEE 802.11a/b/g/j/n/p	**IEEE Standard 802.11a-1999** **IEEE Standard 802.11b-1999** **IEEE Standard 802.11g-2003** **IEEE Standard 802.11j-2004** **IEEE Standard 802.11p-2010** **IEEE Standard 802.11n-2009**

NI WLAN Generation Toolkit for IEEE 802.11a/ b/g/j/n/p/ac	**IEEE Standard 802.11a-1999** **IEEE Standard 802.11b-1999** **IEEE Standard 802.11g-2003** **IEEE Standard 802.11j-2004** **IEEE Standard 802.11p-2010** **IEEE Standard 802.11n-2009** **IEEE Standard 802.11ac-2013**
NI WLAN Generation Toolkit for IEEE 802.11ah	**IEEE P802.11ah/D1.**3
NI WLAN Generation Toolkit for IEEE 802.11af	**IEEE Standard 802.11af-2013**
NI WLAN Generation Toolkit for IEEE 802.11a/ b/g/j/n/p/ac/ax	**IEEE Standard 802.11a-1999** **IEEE Standard 802.11b-1999** **IEEE Standard 802.11g-2003** **IEEE Standard 802.11j-2004** **IEEE Standard 802.11p-2010** **IEEE Standard 802.11n-2009** **IEEE Standard 802.11ac-2013** **IEEE P802.11ax/D8.0**
NI WLAN Generation Toolkit for IEEE 802.11a/ b/g/j/n/p/ac/ax/be	**IEEE Standard 802.11a-1999** **IEEE Standard 802.11b-1999** **IEEE Standard 802.11g-2003** **IEEE Standard 802.11j-2004** **IEEE Standard 802.11p-2010** **IEEE Standard 802.11n-2009** **IEEE Standard 802.11ac-2013** **IEEE P802.11ax/D8.0** **IEEE P802.11be/D0.2**

Note Restart LabVIEW after activation.

This help file contains conceptual and programming reference information for the toolkit.

The NI WLAN Generation Toolkit VIs are located on the **Functions»RF Communications»WLAN»Generation** palette.

NI WLAN Generation Toolkit Examples

NI WLAN Generation Toolkit examples can be found in the following location:

- (Windows 7) **Start»All Programs»National Instruments»WLAN Generation Toolkit»WLAN Generation Toolkit Examples**
- (Windows 10/8.1) **Start»National Instruments»WLAN Generation Toolkit Examples**

Working with Application Manifest

NI recommends that you embed an application manifest in the `.exe` file you create with NI WLAN Generation Toolkit to avoid undesirable visual artifacts on very high resolution displays running Windows 8.1 and 10. For WPF-based applications, the `dpiAware` element in the manifest should be set to `true/pm`. For other applications it should be set to `false`. Refer to Application Manifests on the Microsoft TechNet site for more information about application manifests. Refer to your ADE documentation for information about how to embed an application manifest in your `.exe` file.

Setting Up PXI Trigger Lines for Multiple VSTs

If you are using multiple PXIe-5644, PXIe-5645, or PXIe-5646 RF vector signal transceiver devices, you must create static routes for PXI trigger lines from the master device (source) to the slave devices (destination) using NI Measurement & Automation Explorer (NI MAX). For additional information, refer to the **Signal Routing** topic of the **NI RF Signal Generators Help**.

To determine the possible static routes between the PXI trigger buses of your chassis, complete the following steps:

1. Launch MAX, either by navigating to **Start»All Programs»National Instruments»Measurement&Automation** or by double-clicking the NI MAX desktop icon.
2. Expand **Devices and Interfaces**, and then expand the **Chassis** tree.
3. Select your chassis. The attributes of your chassis are displayed on the right of the MAX window.
4. Click the **Trigger** tab below the attributes view. A table in the **Triggers** view shows the PXI trigger bus segments of your chassis.
5. Configure the static route that you want to make.
    a. To find the bus numbers for the respective devices, verify the slot numbers in which the modules are installed in the chassis and the bus number associated with those slots in the **Bus Mappings Legend** of the **Trigger** tab.
    b. The table in the tab gives the routing information about the PXI trigger lines (PXI_Trig0 to PXI_Trig7). If a route is left unconfigured and available for other NI products to perform programmatic or automated routes to connect devices, the route type is Dynamic.
    c. To route a trigger line from the source bus to the destination, select **Away from source bus** as the route type.
6. Save the changes.

For example, assume that there are two VSTs, Dev0 and Dev1, and you want to use Dev0 as master device and configure trigger routing from Dev0 using MAX.

1. Find the bus numbers associated with the devices. Assume for Dev0, it is bus 1; and for Dev1, it is bus 3.
2. Assuming all the lines are free and you want to route PXI_Trig1 from bus 1 to bus3, change the state of **Routing** to Away from bus1.
3. Save the changes.

If you are using the niWLANG RFSG Configure Multiple Device Synchronization VI for multiple NI RF vector signal transceiver devices, you must configure the **PXI trigger lines** parameter with the routed PXI trigger lines.

Note: If you use PXIe-5644 or PXIe-5645 RF vector signal transceiver devices, you must route one PXI trigger line. If you use NI 5646R RF vector signal transceiver devices, you must route two PXI trigger lines.

Configuring Active Channels (LabVIEW)

To set or get certain parameters and properties, you must specify an active channel using the Active Channel property or the active channel parameter. Such parameters and properties are referred to as channel-specific parameters and properties. An active channel is one of many entities that share the same channel-specific parameters and properties. These entities can be segments, transmit channels, MPDUs, and users.

You must set the Active Channel parameter or property for channel-specific parameters and properties only when you set the Standard property to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, **80211AX MIMOOFDM**, or **80211BE MIMOOFDM**. You must specify a valid active channel string. The following table lists the various active channel strings for different standards.

Standard	Valid Active Channel Strings	Comments
80211N MIMOOFDM, **80211 AH MIMOOFDM**	"channelx"	Indicates a channel with index x .
	"channelx - channely"	x < y Indicates all channels with inde x x to y, both inclusive.
	"channelx, channely, channelv - channelw"	x < y Indicates channels specified by each comma-separated string.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.

80211AC MIMOOFDM, 80211 AX MIMOOFDM	"[segmentx]"	Indicates a segment with index x. You may use "" (empty string) if the Number of Segments prope rty is set to 1.
	"segmentx - segmenty"	x < y Indicates all segments with ind ex x to y, both inclusive.
	"segmentx, segmenty"	Indicates two segments with in dices x and y, respectively.
	"[segmentx]/channely"	Indicates a channel, with index y, of the segment with index x. "segment0/" is optional if the N umber of Segments property is set to 1.
	"[segmentx/]channely - channe lz"	y < z Indicates all channels, with ind ex y to z, both inclusive, in the s egment with index x. "segment0/" is optional if the N umber of Segments property is set to 1.
	"segmentx - segmenty/channel v - channelw"	x < y, v < w Indicates all channels, with ind ex v to w, both inclusive, in the segments with index x to y, bot h inclusive.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.
	"userx/mpduy"	Indicates an MPDU with index y of a user with an index x.

	"userx/mpduy - mpduz"	y < z Indicates all MPDUs with index y to z, both inclusive, of a user with an index x.
	"userx - usery/mpduv- mpduw"	x < y, v < w Indicates all MPDUs with index v to w, both inclusive, of users with an index x to y, both inclus ive.
80211BE MIMOOFDM	"channelx"	Indicates a channel with index x .
	"Channelx - channely"	x < y Indicates all channels with inde x x to y, both inclusive.
	"channelx, channely, channelv - channelw"	x < y Indicates two segments with in dices x and y, respectively.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.
	"userx/mpduy"	Indicates an MPDU with index y of a user with an index x.
	"userx/mpduy - mpduz"	y < z Indicates all MPDUs with index y to z, both inclusive, of a user with an index x.
	"userx - usery/mpduv- mpduw"	x < y, v < w Indicates all MPDUs with index v to w, both inclusive, of users with an index x to y, both inclus ive.

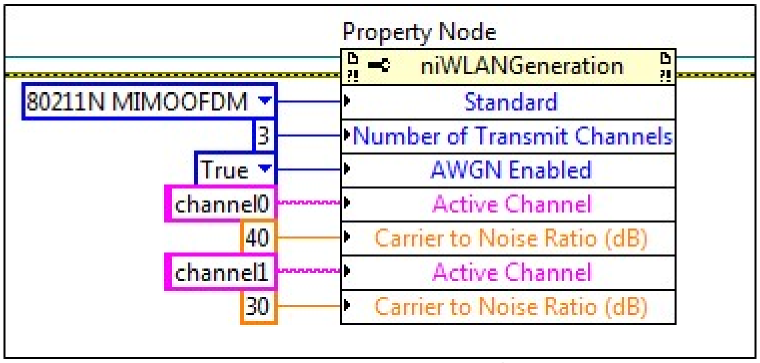
80211AF MIMOOFDM	"[segmentx]"	Indicates a segment with index x. You may use "" (empty string) if the Number of Segments prope rty is set to 1.
	"[segmentx/]channely"	Indicates a channel, with index y, of the segment with index x. "segment0/" is optional if the N umber of Segments property is set to 1.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.
80211A/G OFDM or **80211J OFDM** or **80211P OFDM** or **80211B/G DSSS** or **80211G DSSSOFDM**	"" <empty string>	—

Note Square brackets ([]) indicate optional elements.

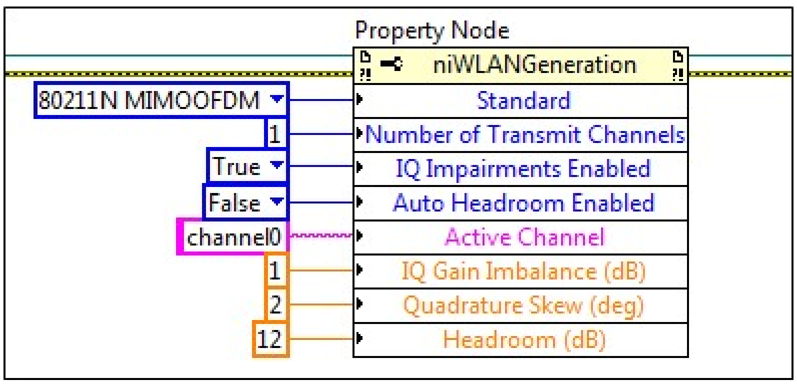
If you specify an active channel, you can read only channel-specific properties for the specified channel. If you attempt to read non-channel-specific properties, the toolkit returns an error.

Active Channel Syntax to Set Properties

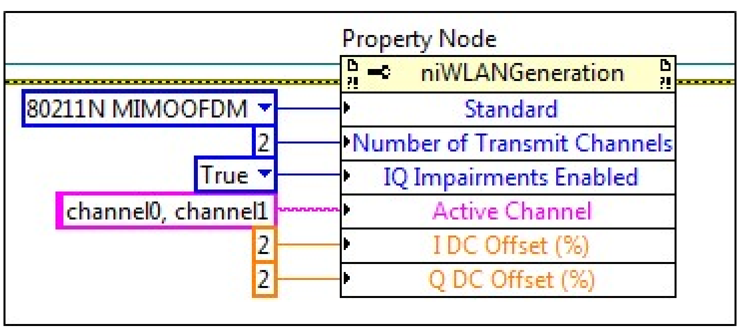
Use the Active Channel property on the niWLANG Property Node to configure the properties of a channel. For example, in the following block diagram, the carrier-to-noise ratio is set to 40 dB for channel0 and 30 dB for channel1. The toolkit sets the carrier-to-noise ratio for channel2 to the default value, which is 50 dB.



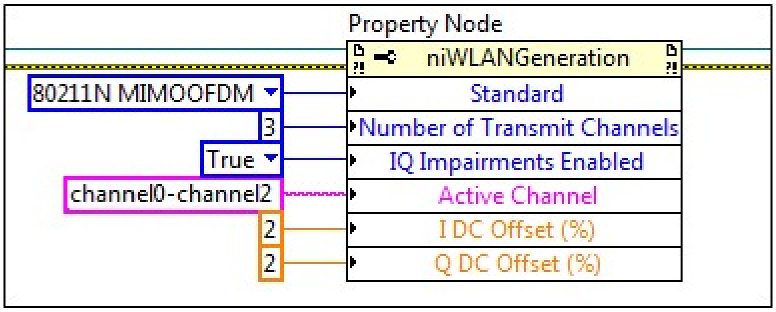
You can set multiple properties for an active channel on one instance of the property node. In the following block diagram, the property node configures channel0 to have an I/Q gain imbalance of 1 dB, quadrature skew of 2 degrees, and headroom of 12 dB.



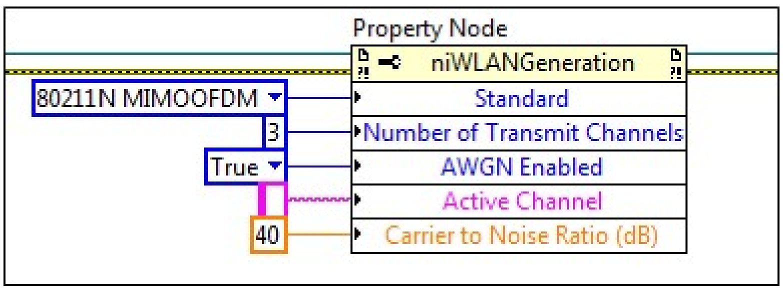
To set the same value on multiple channels, use an active channel to indicate the list of channels that you want to configure. In the following block diagram, channel0 and channel1 are configured to have an I DC offset and Q DC offset of 2% each.



You can also use an active channel to indicate the range of channels that you want to configure. In the following block diagram, channel0, channel1, and channel2 are configured to have an I DC offset and Q DC offset of 2% each.



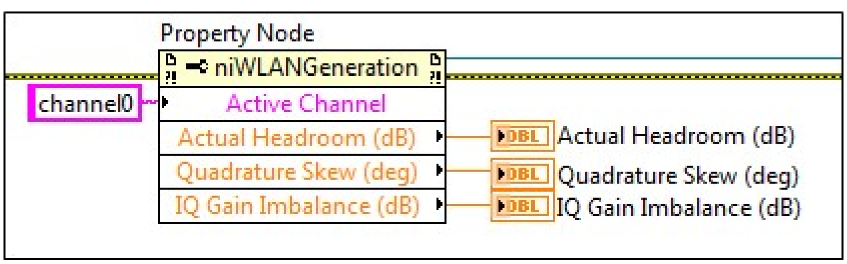
If you set the active channel to "" (empty string) and configure a channel-specific property, the toolkit sets this property to the configured value for all the channels. In the following block diagram, the toolkit sets the carrier-to-noise ratio to 40 dB for all the channels.



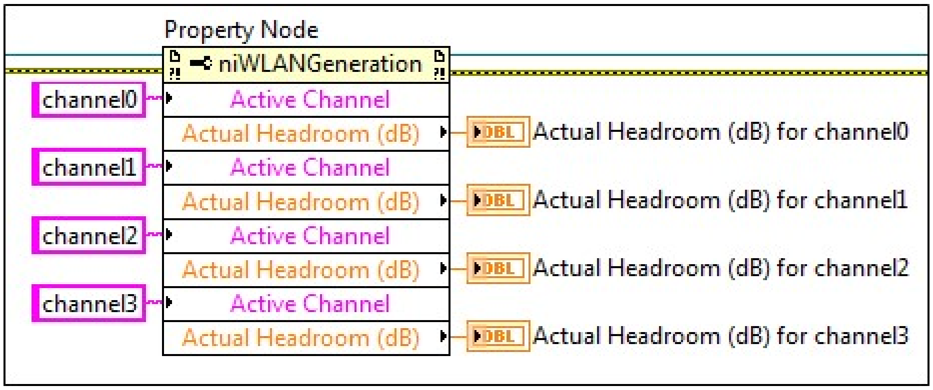
If you specify an active channel, you can configure only channel-specific properties for the specified channel. If you configure non-channel-specific properties, the toolkit returns an error.

Active Channel Syntax to Get Properties

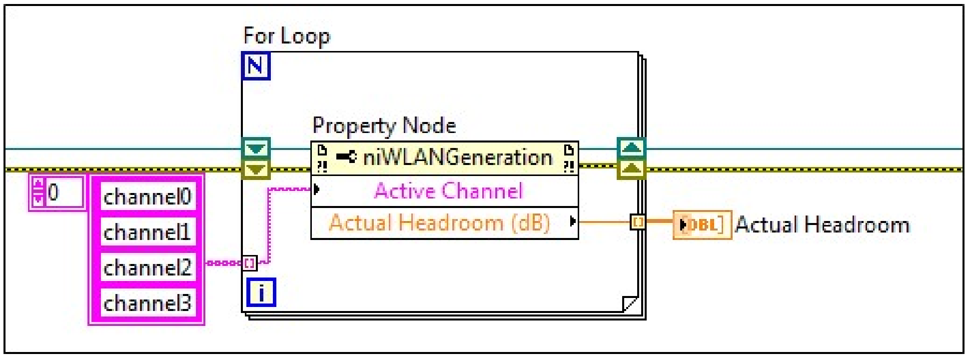
You can read multiple properties for an active channel on one instance of the property node, as shown in the following block diagram.



You cannot read properties for multiple channels using a single active channel. You must specify the active channel for each channel. In the following block diagram, the active channel is specified for each channel to read the actual headroom.



You can also use a For Loop to read properties for multiple channels.



If you specify an active channel, you can read only channel-specific properties for the specified channel. If you attempt to read non-channel specific properties, the toolkit returns an error.

Configuring Active Channels (LabWindows™/CVI™)

To set or get certain parameters and attributes, you must specify an active channel using the **channelString** parameter. Such parameters and attributes are referred to as channel-specific parameters and attributes. An active channel is one of many entities that share the same channel-specific parameters and attributes. These entities can be segments, transmit channels, MPDUs, and users.

You must set the **channelString** parameter for channel-specific parameters and attributes only when you set the NIWLANG_STANDARD attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM`, or `NIWLANG_VAL_STANDARD_80211BE_MIMO_OFDM`. You must specify a valid channel string. The following table lists the various active channel strings for different standards.

NIWLANG_STANDARD	Valid Active Channel Strings	Comments
NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM	"channel**x**"	Indicates a channel with index x .
	"channel**x** - channel**y**"	**x** < **y** Indicates all channels with index **x** to **y**, both inclusive.
	"channel**x**, channel**y**, channel**v** - channel**w**"	**x** < **y** Indicates channels specified by each comma-separated string.
	"mpdu**x**"	Indicates an MPDU with index **x** .
	"mpdu**x** - mpduy"	**x** < **y** Indicates all MPDUs with index **x** to **y**, both inclusive.
NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM	"[segment**x**]"	Indicates a segment with index **x**. You may use "" (empty string) if the Number of Segments property is set to 1.
	"segment**x** - segmenty"	**x** < **y** Indicates all segments with index **x** to **y**, both inclusive.
	"segment**x**, segmenty"	Indicates two segments with indices **x** and **y**, respectively.
	"[segmentx/]channely"	Indicates a channel, with index y, of the segment with index x. "segment0/" is optional if the Number of Segments property is set to 1.

	"[segment**x**/]channel**y** - channel**z**"	**y** < **z** Indicates all channels, with index **y** to **z**, both inclusive, in the segment with index **x**. "segment0/" is optional if the Number of Segments property is set to 1.
	"segmentx - segmenty/channelv - channelw"	x < y, v < w Indicates all channels, with index v to w, both inclusive, in the segments with index x to y, both inclusive.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.
	"userx/mpduy"	Indicates an MPDU with index y of a user with an index x.
	"userx/mpduy - mpduz"	y < z Indicates all MPDUs with index **y** to **z**, both inclusive, of a user with an index x.
	"user**x** - user**y**/mpdu**v**- mpdu**w**"	x < **y**, v < w Indicates all MPDUs with index v to w, both inclusive, of users with an index x to y, both inclusive.
`NIWLANG_VAL_STANDARD_80211BE_MIMO_OFDM`	"channelx"	Indicates a channel with index x.
	"Channelx - channely"	x < y Indicates all channels with index x to y, both inclusive.

	"channelx, channely, channelv - channelw"	x < y Indicates two segments with in dices x and y, respectively.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.
	"userx/mpduy"	Indicates an MPDU with index y of a user with an index x.
	"userx/mpduy - mpduz"	y < z Indicates all MPDUs with index y to z, both inclusive, of a user with an index x.
	"userx - usery/mpduv- mpduw"	x < y, v < w Indicates all MPDUs with index v to w, both inclusive, of users with an index x to y, both inclus ive.
`NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM`	"segmentx"	Indicates a segment with index x. You may use "" (empty string) if the Number of Segments prope rty is set to 1.
	"[segment**x/**]channel**y**"	Indicates a channel, with index y, of the segment with index x. "segment0/" is optional if the N umber of Segments property is set to 1.
	"mpdux"	Indicates an MPDU with index x.
	"mpdux - mpduy"	x < y Indicates all MPDUs with index x to y, both inclusive.

NIWLANG_VAL_STANDARD_80211AG_OFDM or NIWLANG_VAL_STANDARD_80211J_OFDM or NIWLANG_VAL_STANDARD_80211P_OFDM or NIWLANG_VAL_STANDARD_80211BG_DSSS or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM	"" <empty string>	—

Note Square brackets ([]) indicate optional elements.

Active Channel String Syntax to Set Channel-Specific Parameters

You must specify the active channel for configuring channel-specific parameters and attributes using **channelString** parameter to configure the attributes of a channel. In the following example, the carrier-to-noise ratio is set to 40 dB for channel0 and 30 dB for channel1. The toolkit sets the carrier-to-noise ratio for channel2 to its default value, which is 50 dB.

```
niWLANG_SetStandard (generationSession, NULL, NIW
LANG_VAL_STANDARD_80211N_MIMO_OFDM);
niWLANG_SetNumberOfTransmitChannels (generationSe
ssion, NULL, 3);
niWLANG_SetAWGNEnabled (generationSession, NULL,
NIWLANG_VAL_TRUE);
niWLANG_SetCarrierToNoiseRatio (generationSession
, "channel0", 40);
niWLANG_SetCarrierToNoiseRatio (generationSession
, "channel1", 30);
```

You can set multiple attributes for a **channelString** parameter. In the following example, channel0 is configured to have an I/Q gain imbalance of 1 dB, quadrature skew of 2 degrees, and headroom of 12 dB.

```
niWLANG_SetStandard (generationSession, NULL, NIW
LANG_VAL_STANDARD_80211N_MIMO_OFDM);
niWLANG_SetNumberOfTransmitChannels (generationSe
ssion, NULL, 1);
niWLANG_SetAllIQImpairmentsEnabled (generationSes
sion, NULL, NIWLANG_VAL_TRUE);
niWLANG_SetAutoHeadroomEnabled (generationSession
, NULL, NIWLANG_VAL_FALSE);
niWLANG_SetIQGainImbalance (generationSession, "c
hannel0", 1);
niWLANG_SetQuadratureSkew (generationSession, "ch
annel0", 2);
niWLANG_SetHeadroom (generationSession, "channel0
", 12);
```

To set the same value on multiple channels, use the **channelString** parameter to indicate the list of channels that you want to configure. In the following example, channel0 and channel1 are configured to have an I DC offset and Q DC offset of 2% each.

```
niWLANG_SetStandard (generationSession, NULL, NIW
LANG_VAL_STANDARD_80211N_MIMO_OFDM);
niWLANG_SetNumberOfTransmitChannels (generationSe
ssion, NULL, 2);
niWLANG_SetAllIQImpairmentsEnabled (generationSes
sion, NULL, NIWLANG_VAL_TRUE);
niWLANG_SetIDCOffset (generationSession, "channel
0, channel1", 2);
niWLANG_SetQDCOffset (generationSession, "channel
0, channel1", 2);
```

You can also use a **channelString** parameter to indicate the range of channels that you want to configure. In the following example, channel0, channel1, and channel2 are configured to have an I DC offset and Q DC offset of 2% each.

```
niWLANG_SetStandard (generationSession, NULL, NIW
LANG_VAL_STANDARD_80211N_MIMO_OFDM);
niWLANG_SetNumberOfTransmitChannels (generationSe
ssion, NULL, 3);
niWLANG_SetAllIQImpairmentsEnabled (generationSes
sion, NULL, NIWLANG_VAL_TRUE);
niWLANG_SetIDCOffset (generationSession, "channel
0-channel2", 2);
niWLANG_SetQDCOffset (generationSession, "channel
0-channel2", 2);
```

If you set the **channelString** parameter to "" (empty string) or NULL and configure a channel-specific attribute, the toolkit sets this attribute to the configured value for all the channels. In the following example, the toolkit sets the carrier-to-noise ratio to 40 dB for all the channels.

```
niWLANG_SetStandard (generationSession, NULL, NIW
LANG_VAL_STANDARD_80211N_MIMO_OFDM);
niWLANG_SetNumberOfTransmitChannels (generationSe
ssion, NULL, 3);
niWLANG_SetAWGNEnabled (generationSession, NULL,
NIWLANG_VAL_TRUE);
niWLANG_SetCarrierToNoiseRatio (generationSession
, " ", 40);
```

If you specify the **channelString** parameter, you can configure only channel-specific attributes for the specified channel. If you configure non-channel-specific attributes, the toolkit returns error.

Active Channel String Syntax to Get Channel-Specific Parameters

You can read channel-specific parameters for a **channelString** parameter, as shown in the following example:

```
niWLANG_GetActualHeadroom (generationSession, "ch
annel0", &ActualHeadroom);
niWLANG_GetQuadratureSkew (generationSession, "ch
annel0", &QuadratureSkew);
niWLANG_GetIQGainImbalance (generationSession, "c
hannel0", &IQGainImbalance);
```

You cannot read channel-specific parameters for multiple channels using a single **channelString** parameter. You must specify the **channelString** parameter for each channel. In the following example, the active channel is specified for each channel to read the actual headroom.

```
niWLANG_GetActualHeadroom (generationSession, "ch
annel0", &ActualHeadroom0);
niWLANG_GetActualHeadroom (generationSession, "ch
annel1", &ActualHeadroom1);
niWLANG_GetActualHeadroom (generationSession, "ch
annel2", &ActualHeadroom2);
niWLANG_GetActualHeadroom (generationSession, "ch
annel3", &ActualHeadroom3);
```

You can also use a For Loop to read attributes for multiple channels.

```
#define MAX_WLAN_CHANNELS 4
float64 ActualHeadroom[MAX_WLAN_CHANNELS];
static char *MIMOChannelString[] = {"channel0", "
channel1", "channel2", "channel3" };
for(i=0; i< MAX_WLAN_CHANNELS; i++)
{
niWLANG_GetActualHeadroom (generationSession, MIM
OChannelString[i], ActualHeadroom[i]);
}
```

If you specify a **channelString** parameter, you can read only channel-specific attributes for the specified channel. If you attempt to read non-channel-specific properties, the toolkit returns an error.

RFSG Database

The RFSG database is a feature that allows users to store and retrieve various properties associated with a particular waveform.

You can create multiple waveforms for a particular standard and download these waveforms to the arbitrary waveform generator (arb) memory. You can then generate all these waveforms using either a single script or multiple scripts. You need to set the appropriate NI-RFSG properties corresponding to the waveform being generated. Use the RFSG database feature to retrieve the properties for the waveform being generated.

Currently, the toolkit stores the I/Q rate, headroom, burst start locations, burst stop locations, and waveform size properties for each waveform. The toolkit requires that the I/Q rate of all the waveforms be the same so that you can generate them using a single script. The toolkit selects the minimum headroom value and applies it to all waveforms. You can add more properties to the RFSG database to track more RFSG properties per waveform.

If you want to generate a waveform from a specific standard, the toolkit configures the properties of the NI-RFSG to do this task. These properties are global with respect to the driver. Hence, the values of the properties apply to all waveforms. However, this fact may be an issue if you want to generate waveforms that have different values or properties associated with them. For example, if you want to generate waveform 1, waveform 2, and then waveform 1 again, each time a new waveform is generated, you must reconfigure the driver for that waveform, even if you have already configured it.

By using the RFSG database, you can store the required properties for each waveform in the database and then query them later while generating those waveforms.

Windowing

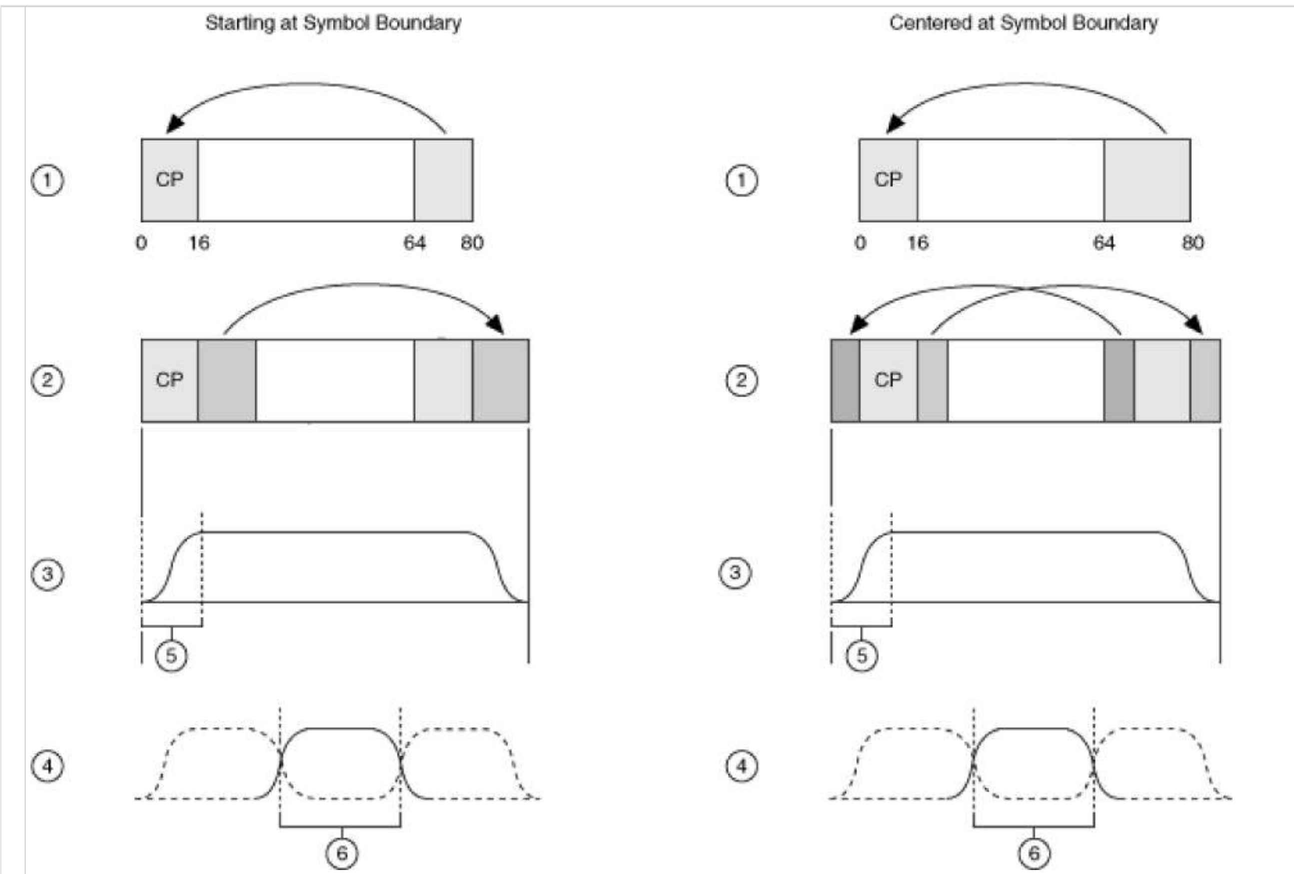
Windowing for OFDM Signals

OFDM symbols are generated one symbol at a time. When the symbols are combined consecutively, spectral regrowth occurs because of the amplitude and phase discontinuity between the consecutive symbols. Time-domain windowing is performed on OFDM symbols to improve the spectral characteristics of the transmitted signal. Thus, spectral regrowth is reduced. Transitions between consecutive symbols are smoothed by the following methods:

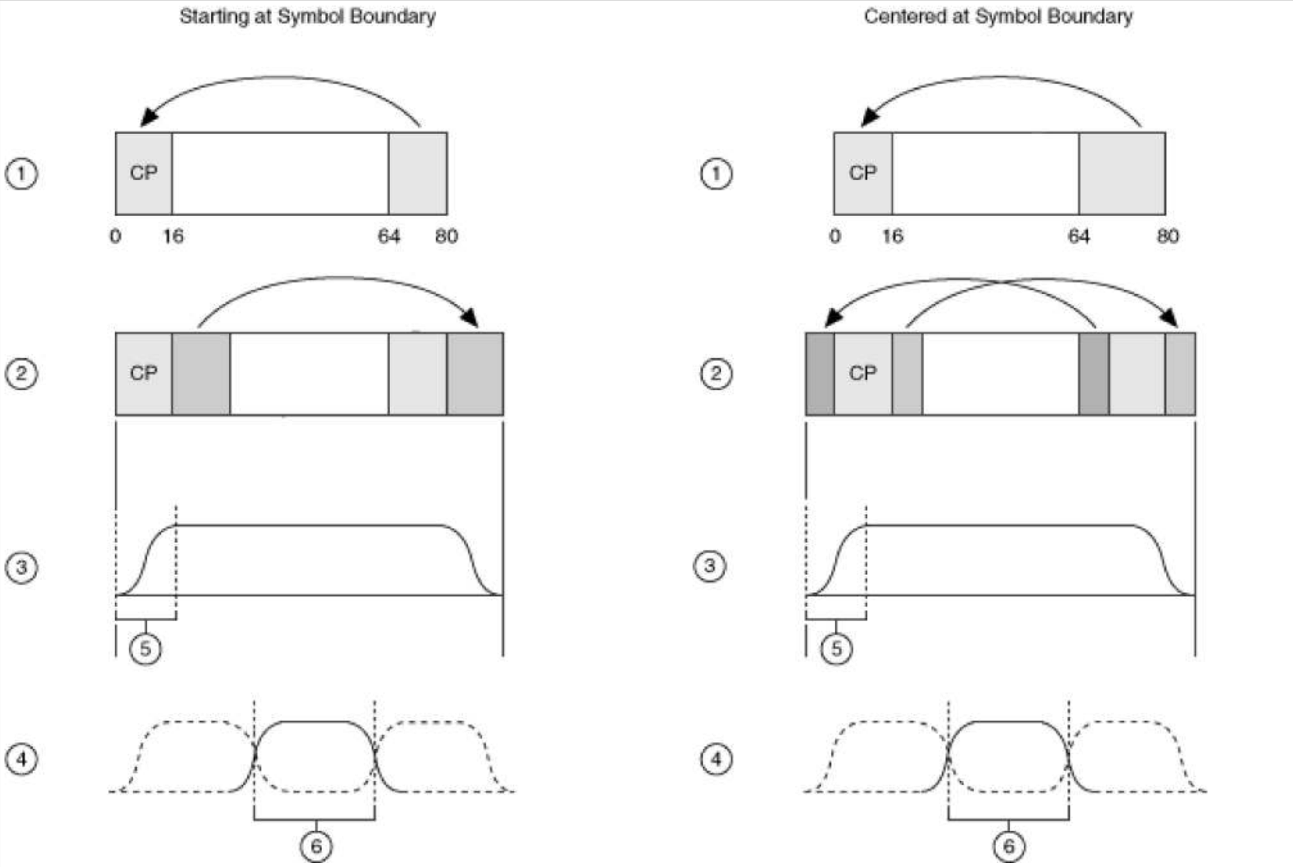
- Extending the OFDM symbols cyclically based on the Windowing Method property.
  - **Starting at Symbol Boundary**:
    - Appending the first window length (W) samples of the symbol at the end.
    - Overlapping the cyclic prefix (CP) with samples of the previous symbol and overlapping the cyclic suffix with the CP samples of the next symbol.
  - **Centered at Symbol Boundary**:
    - Appending the first W/2 samples of the symbols at the end and suffix W/2 samples before CP at the beginning.
    - Overlapping the extended CP with the samples of the previous symbol and overlapping the cyclic suffix with the samples of the CP of the next symbol
- Applying windowing to the beginning and end of the OFDM symbol. Two windows are applied, with one window being the mathematical inverse of the other.
- Applying the first cosine window, which rolls over from 0 to 1, over the window length of samples from the beginning of the symbol after cyclic extension.
- Applying the second cosine window, which rolls over from 1 to 0, over the last window length of samples after cyclic extension.

- After applying windowing to all the OFDM symbols, adding the overlapped segments of the current symbol and the previous symbol, as well as the overlapped segments of the current symbol and the next symbol to maintain the required OFDM symbol length.

The windowing algorithm is controlled by the Window Length and Windowing Method properties. The selected window shape is a raised-cosine window. The shape and length of the window relative to an OFDM symbol length, along with the shaping procedure, are shown in the following image.



1) Normal Symbol with Cyclic Prefix (CP)	5) Window-half-length
2) Symbol Cyclically Extended by Window-half-length Samples from Each Side	6) 2 * Window-half-length
3) Extended Symbol Shaped by Window	7) OFDM Symbol Length
4) Superposition by Summation of Successive Symbols	



1. Normal Symbol with Cyclic Prefix (CP)
2. Symbol Cyclically Extended by Window-half-length Samples from Each Side
3. Extended Symbol Shaped by Window
4. Superposition by Summation of Successive Symbols
5. Window-half-length
6. 2 * Window-half-length
7. OFDM Symbol Length

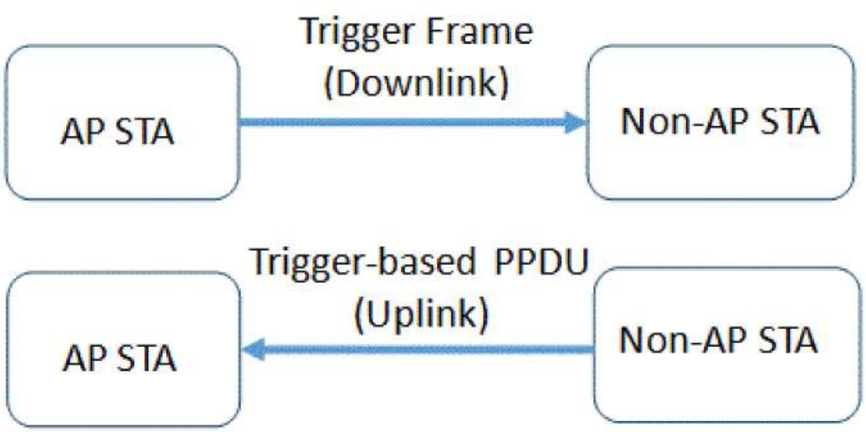
Windowing for DSSS Signals

For direct sequence spread spectrum (DSSS) signals, windowing ensures power ramp up and down for the entire burst. The rising half of the cosine window is applied at the beginning of the burst and the falling half of the window is applied at the end of the burst. The length of the window is specified by the Window Length property.

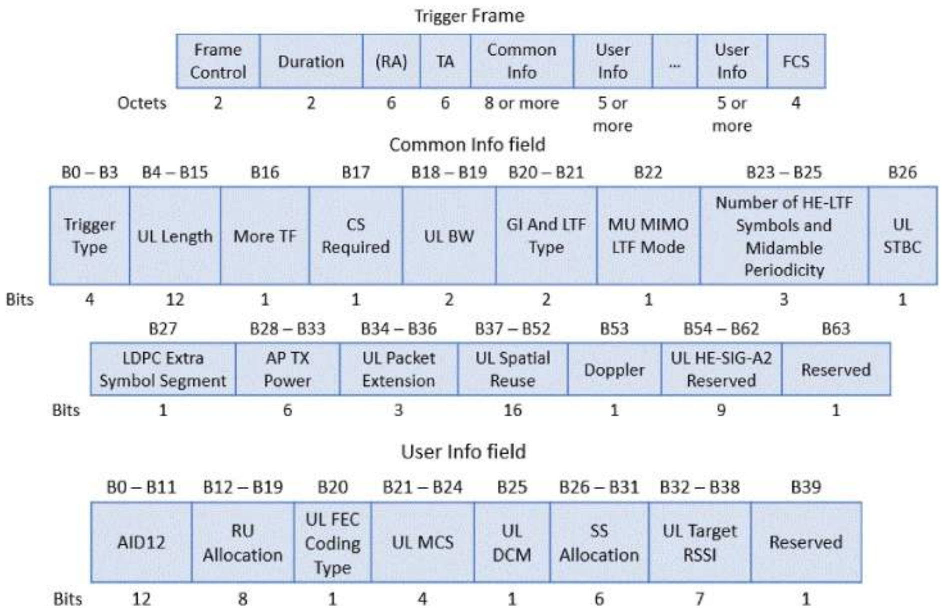
The following figure represents the windowing for DSSS signals.

Generating the Trigger Frame

The multi-user transmission (both OFDMA and MU MIMO) on uplink in 802.11ax (HE) is initiated by a trigger frame from an Access Point Station (AP STA), as shown in the following figure.



As specified in standard IEEE P802.11ax/D8.0; the trigger frame contains all the user related information, and common information across users. The following figure depicts the format of the trigger frame.



To generate a trigger frame using the NI WLAN Generation Toolkit APIs, you must use two generation toolkit sessions. One session to configure the trigger frame content and another one for transmit PHY. The transmit PHY could be one of the OFDM standards 802.11a/g, 802.11n, 802.11ac, or 802.11ax.

You must complete the following steps to create the trigger frame MSDU bits.

1. Configure the first session for 802.11ax Trigger-based PPDU.
2. Enter the common settings such as bandwidth, GI, etc., directly to the session.
3. Configure the payload parameters on the session to derive the other common parameters such as L-SIG Length, Pre-FEC Padding factor, etc.
   Or configure L-SIG Length, Pre-FEC Padding Factor, PE Disambiguity, and LDPC Extra Symbol Segment directly.
4. Configure the per user information such as MCS index, RU allocation etc. with the “userx” active channel.
5. Call the niWLANG Create Trigger Frame MSDU Bits VI or niWLANG_CreateTriggerFrameMSDUBits function to create the bits.

Note: For the list of trigger frame parameters, refer to the following table.

Trigger Frame Subfield	Value

Common Information field	
Trigger Type	Set to 0. Only Basic Trigger is supported.
UL Length	Either derived from specified per user payload s ettings, or specified using L-SIG Length property /NIWLANG_L_SIG_LENGTH attribute
More TF	Set to 0
CS Required	Specified using CS Required property/NIWLANG _CS_REQUIRED attribute
UL BW	Specified using the Channel Bandwidth propert y/NIWLANG_CHANNEL_BANDWIDTH attribute
GI And LTF Type	Specified using the following properties: ▪ Guard Interval Type/NIWLANG_GUARD_ INTERVAL_TYPE ▪ HE-LTF Size/NIWLANG_HE_LTF_SIZE
MU-MIMO LTF Mode	Specified using the MU-MIMO LTF Mode Enabled property/NIWLANG_MU_MIMO_LTF_MODE_ENA BLED attribute
Number of HE-LTF Symbols and Midamble Periodicity	Specified using the Number of HE-LTF Symbols property/NIWLANG_NUMBER_OF_HE_LTF_SYM BOLS attribute and Midamble Periodicity proper ty/NIWLANG_MIDAMBLE_PERIODICITY attribute.
UL STBC	Specified using the STBC All Streams Enabled pr operty/NIWLANG_STBC_ALL_STREAMS_ENABL ED attribute
LDPC Extra Symbol Segment	Either derived from specified per user payload s ettings or specified using LDPC Extra Symbol Se gment property/NIWLANG_LDPC_EXTRA_SYMB OL_SEGMENT attribute
AP Tx Power	Specified using the AP Tx Power property/NIWL ANG_TRIGGER_FRAME_AP_TXP attribute

UL Packet Extension	Either derived from specified per user payload settings or specified using following properties: ▪ Pre-FEC Padding Factor property/NIWLANG_PRE_FEC_PADDING_FACTOR attribute ▪ PE Disambiguity property/NIWLANG_PE_DISAMBIGUITY attribute
UL Spatial Reuse	Set all to 0
Doppler	Set to 0 if Midamble Periodicity property/NIWLANG_MIDAMBLE_PERIODICITY attribute is set to None/NIWLANG_VAL_NONE, set to 1 otherwise.
UL HE-SIG-A2 Reserved	Set all to 1
Reserved	Set to 1
User Info field	
AID12	Specified using the AID12 property/NIWLANG_AID12 attribute
RU Allocation	Specified using the RU Size and RU Offset properties/NIWLANG_RU_SIZE and NIWLANG_RU_OFFSET attributes
UL FEC Coding Type	Specified using the FEC Coding Type property/NIWLANG_FEC_CODING_TYPE attribute
UL MCS	Specified using the MCS Index property/NIWLANG_MCS_INDEX attribute
UL DCM	Specified using the DCM Enabled property/NIWLANG_DCM_ENABLED attribute
SS Allocation	Specified using following properties: ▪ Number of Space Time Streams property/NIWLANG_NUMBER_OF_SPACE_TIME_STREAMS attribute

	▪ Space Time Stream Offset property/NIWLANG_SPACE_TIME_STREAM_OFFSET attribute
UL Target RSSI	Specified using the Target RSSI property/NIWLANG_TRIGGER_FRAME_TARGET_RSSI attribute
Reserved	Set to 1
Trigger Dependent User Info	Set to 32

After completing the aforementioned steps, generate the trigger frame by completing the following steps.

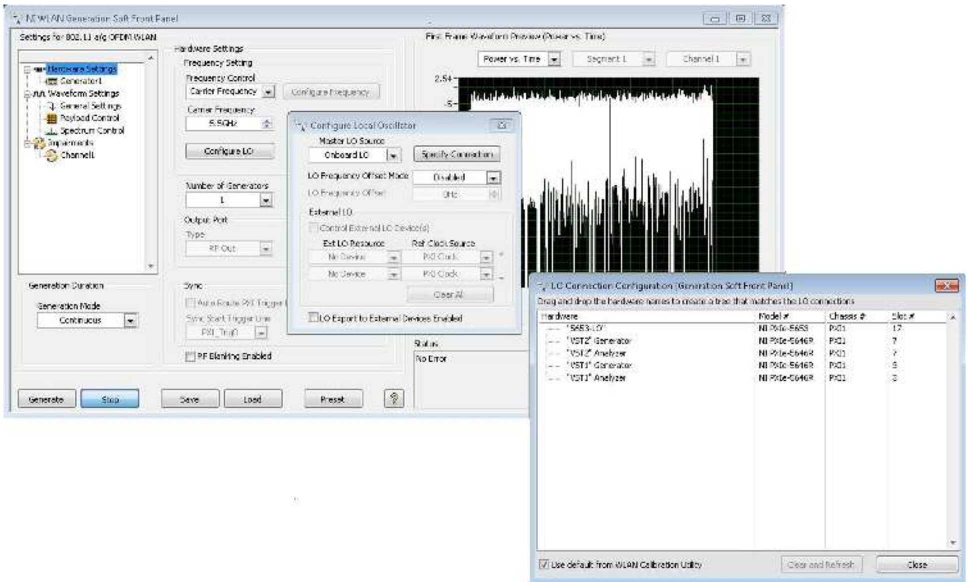
1. Configure the second session to any one of the OFDM standards: 802.11a/g, 802.11n, 802.11ac, or 802.11ax.
2. Configure the session to generate user-defined bit pattern.
3. Input the MSDU bits created by the first session to the current session as a user-defined bit pattern.
4. Configure MAC Frame Type as Trigger Frame.
5. Configure the MAC Header settings and MAC Padding Duration (s).
6. Call the niWLANG RFSG Create and Download Waveforms VI, or niWLANG_RFSGCreateAndDownloadWaveform/ niWLANG_RFSGCreateAndDownloadMIMOWaveforms functions to generate the Trigger Frame waveform.

An example demonstrating the creation and the generation of trigger frame is located at: `LabVIEW\examples\RF Toolkits\WLAN\generation\Basic` or `Documents\National Instruments\WLAN Toolkit\Examples\C\Generation`

Configuring LO Connection in WLAN Generation Soft Front Panel

WLAN Generation Soft Front Panel supports configuring the LO connections.

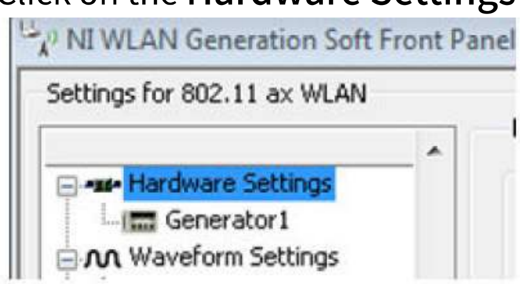
The following image depicts the **LO Connection Configuration** window in the WLAN Generation Soft Front Panel.



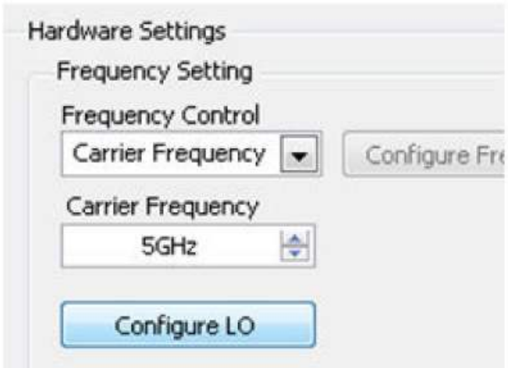
Navigating to LO Configuration Window

Complete the following steps to configure the LO connections:

1. Navigate to **Start»All Programs»National Instruments»WLAN Generation Toolkit»WLAN Generation Soft Front Panel** to launch the WLAN Generation Soft Front Panel.
2. Click on the **Hardware Settings** tab, as illustrated in the following figure.

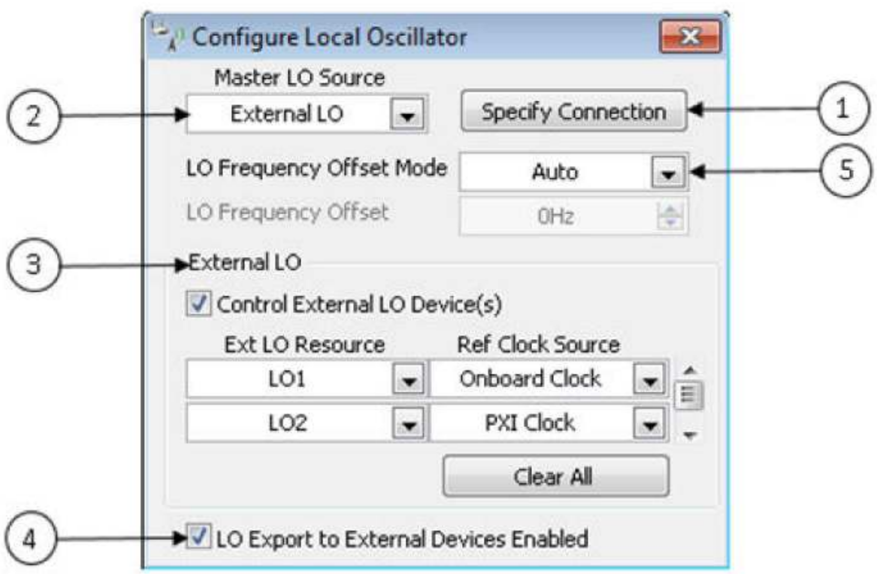


3. Click the **Configure LO** button.



Configuring the Local Oscillator parameters

The following figure illustrates the elements of the **Configure Local Oscillator** window.



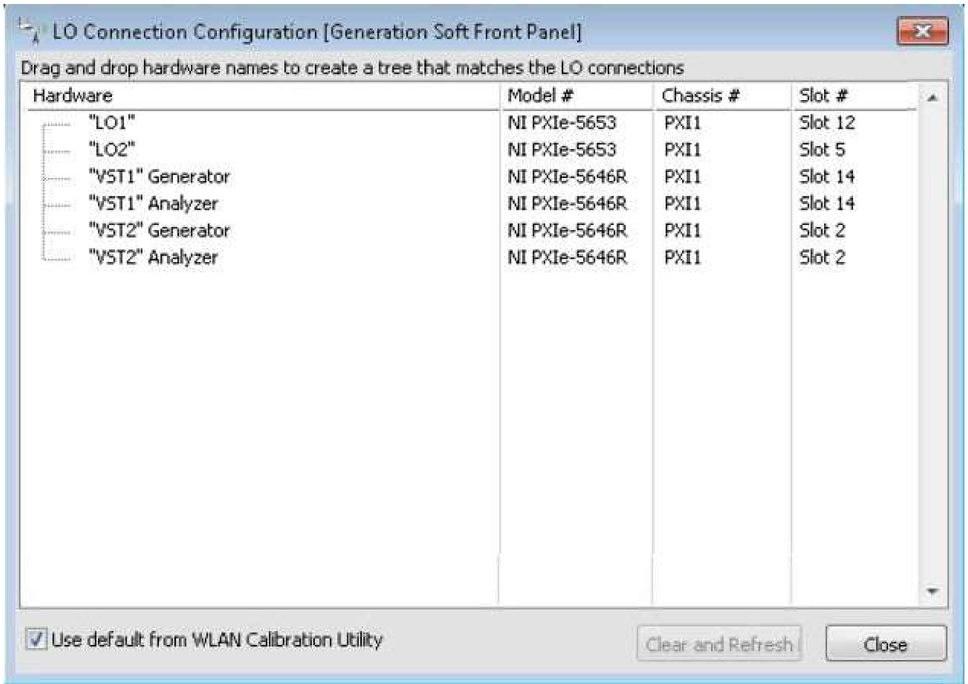
1. **Specify Connection**: Use this button to open the **LO Connection Configuration** window and specify the hardware connections. For more details on how to specify the connections, refer to the Specifying LO Connection in LO Connection Configuration Window section of this topic.
2. **Master LO Source**: In an LO section, 'Master' is an NI RF Vector Signal Generator from which other NI RF Vector Signal Generators are connected in a daisy chain fashion. The value of this control applies to the Master Generator.

In an LO section, when NI RF Vector Signal Generators are not connected in a daisy chain fashion, the value of this control applies to all the NI RF Vector Signal Generators. Select **Onboard LO** to use the Onboard local oscillator for the Master Generator. Select **External LO** to use external local oscillator device/signal as the LO Source for the Master Generator. For more details, refer to the Specifying LO Connection in LO Connection Configuration Window section of this topic.

3. **External LO Configuration:** Use the **Control External LO Device(s)** check box to control the selected external LO devices. Choose the External LO resource names in the **Ext LO Resource** drop down menu. The respective reference clock for the external LO devices are selected in the **Ref Clock Source** drop down menu. Use the **Clear All** button to clear the values of the **Ext LO Resource** and **Ref Clock Source** columns.
4. **LO Export to External Devices Enabled:**Turn on this option if you are exporting the LO signal from an NI RF Vector Signal Generator device to an NI RF Vector Signal Analyzer device.
5. **LO Frequency Offset Mode:** Select **Auto** to automatically configure the LO Frequency Offset. If you want to set the LO Frequency Offset to any other value, select **User defined** from the drop down menu, and specify the desired value for the LO Frequency Offset in the **LO Frequency Offset**. If LO Frequency Offset Mode is **Disabled**, LO Frequency Offset will be set to **0** Hz.

Specifying LO Connection in a LO Connection Configuration Window

Use the **LO Connection Configuration** window to specify the LO connections.



Use Default from WLAN Calibration Utility

Select **Use Default from WLAN Calibration Utility** check box to load the default LO connection configuration saved from the WLAN Calibration Utility. From the default configuration, all the NI RF Vector Signal Generators, NI RF Vector Signal Analyzers, and External LO Devices (NI 5653) along with the LO connection topology, their respective model, chassis, and slot numbers are populated in the configuration tree. WLAN Generation Soft Front Panel launches with the default LO connection configuration, if found.

You cannot edit the configuration tree when this checkbox is selected. If you need to change the LO connections, deselect this checkbox, drag and drop the hardware names to create a tree that matches the LO connections in the system.

Clear and Refresh Hardware

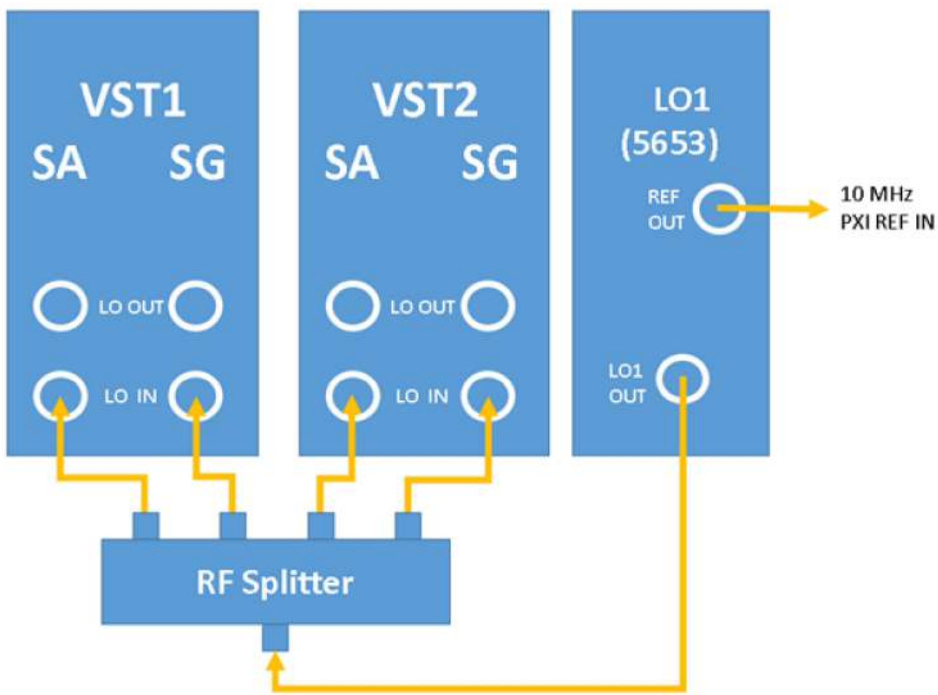
Use **Clear and Refresh Hardware** button to clear the LO Connection configuration tree and populate all the NI RF Vector Signal Generators, NI RF Vector Signal Analyzers, and External LO Devices (NI 5653) present in the system, along with their respective model, chassis, and slot numbers. If **Use Default from WLAN**

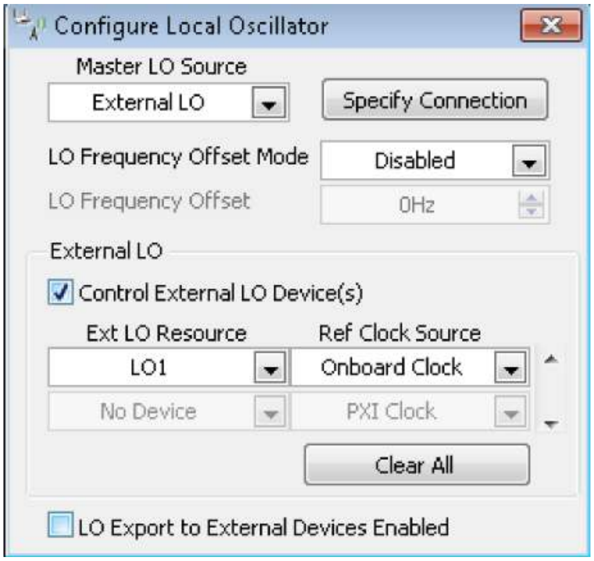
Calibration Utility check box is selected, you cannot clear and refresh the hardware.

Note Analyzer devices displayed in this window are not programmed by the WLAN Generation Soft Front Panel. You must use the WLAN Analysis Soft Front Panel to configure Analyzer devices.

The following figures illustrate different connection configuration topologies and how it is depicted in the **LO Connection Configuration** window.

Illustration 1: 2x2 MIMO System with External LO shared between SAs and SGs





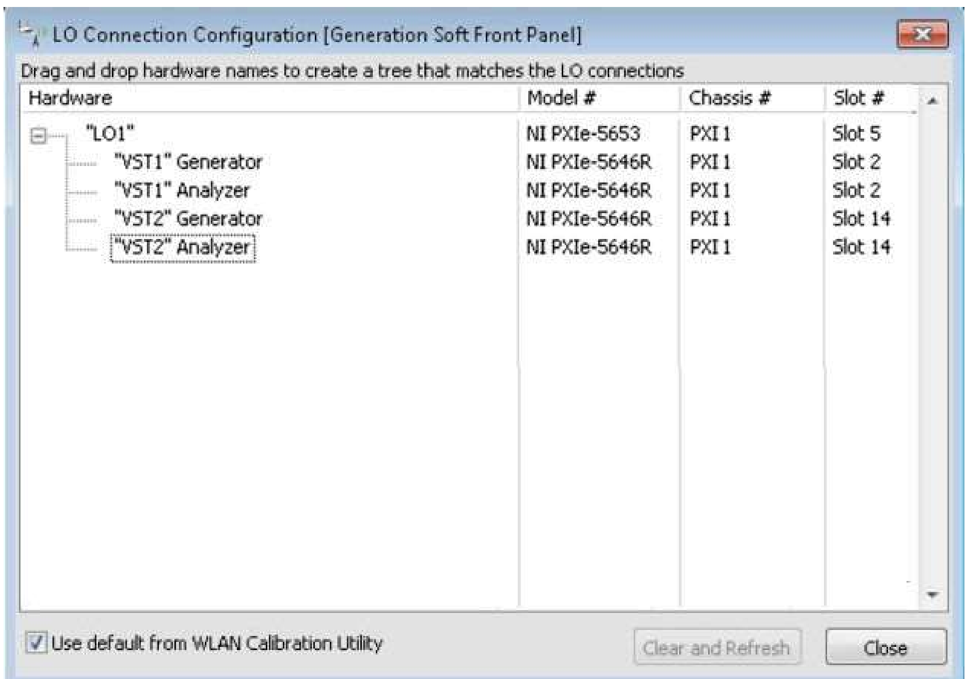
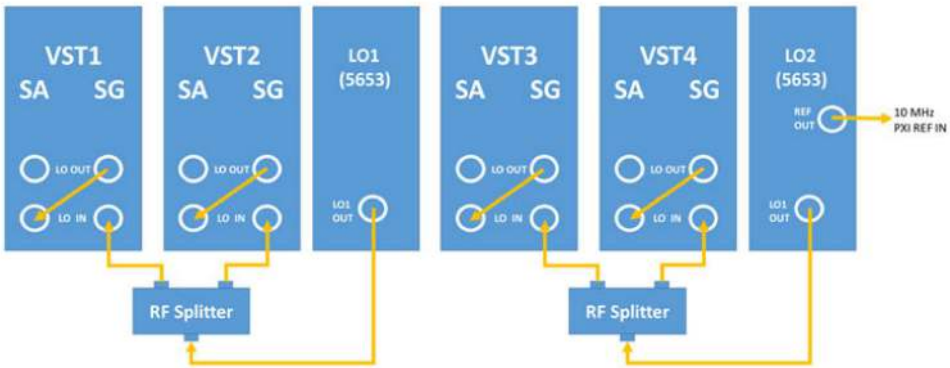
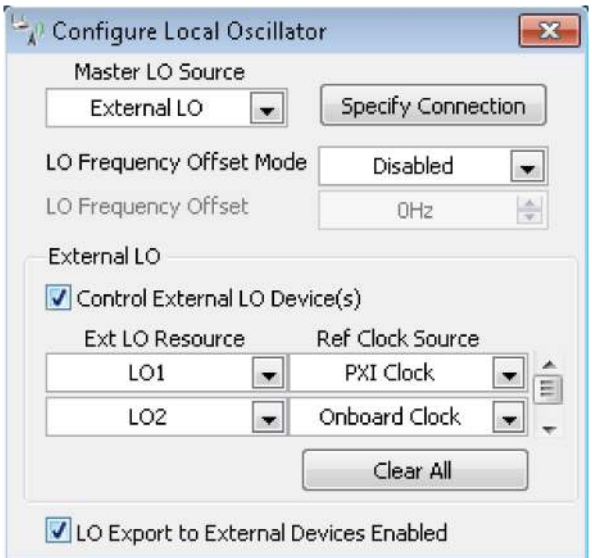


Illustration 2: 4x4 MIMO/Multi Segment System with two External LOs shared between SGs. The SG LOs are exported to SAs


VST1
SA SG
LO OUT
LO IN
VST2
SA SG
LO OUT
LO IN
LO1
(5653)
LO1
OUT
VST3
SA SG
LO OUT
LO IN
VST4
SA SG
LO OUT
LO IN
LO2
(5653)
REF
OUT
10 MHz
PXI REF IN
LO1
OUT
RF Splitter
RF Splitter


Configure Local Oscillator
Master LO Source
External LO
Specify Connection
LO Frequency Offset Mode
Disabled
LO Frequency Offset
0Hz
External LO
Control External LO Device(s)
Ext LO Resource
Ref Clock Source
LO1
PXI Clock
LO2
Onboard Clock
Clear All
LO Export to External Devices Enabled

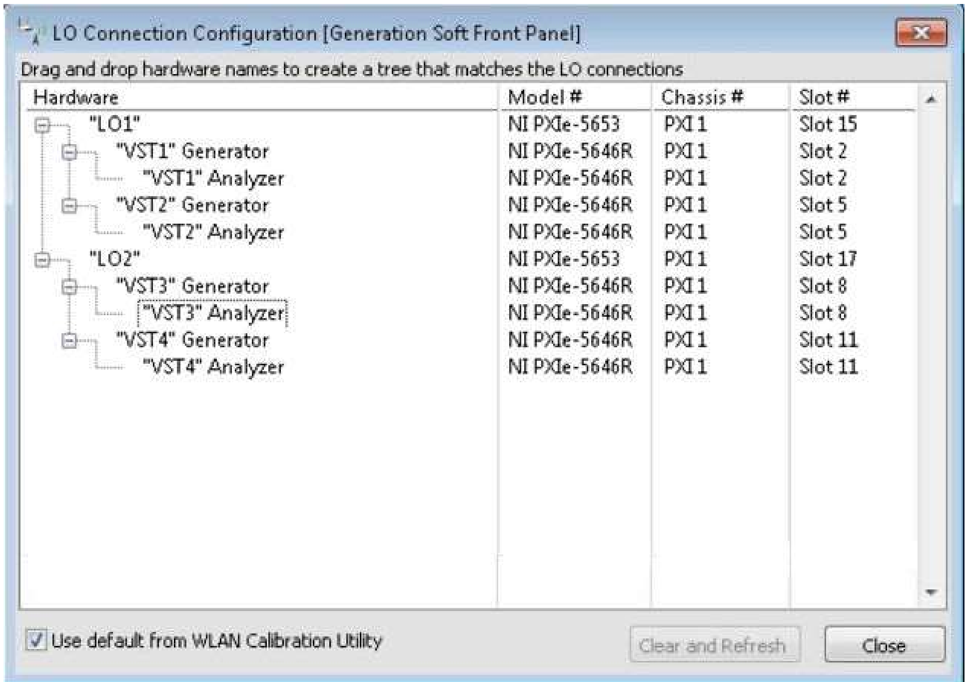
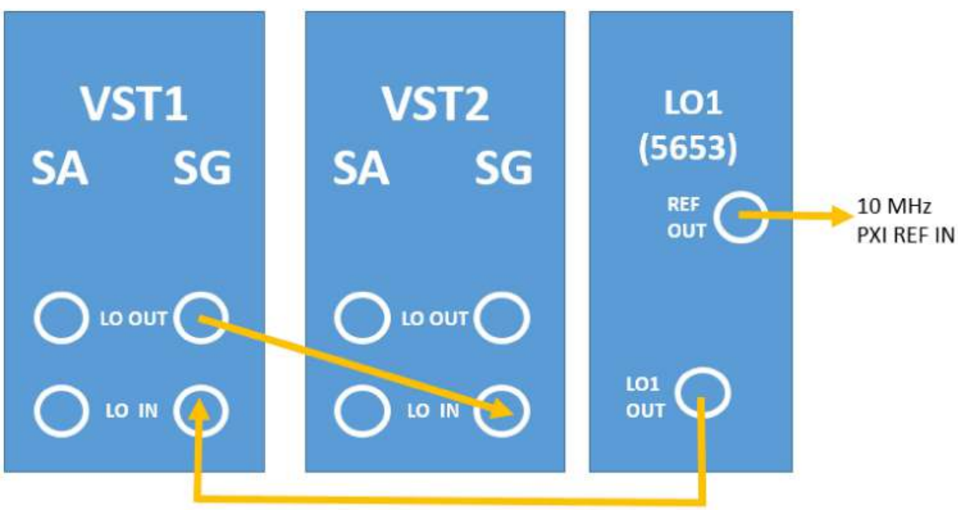
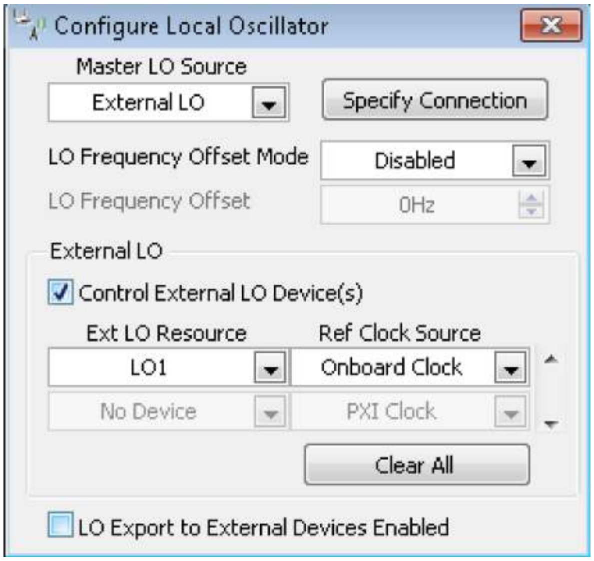
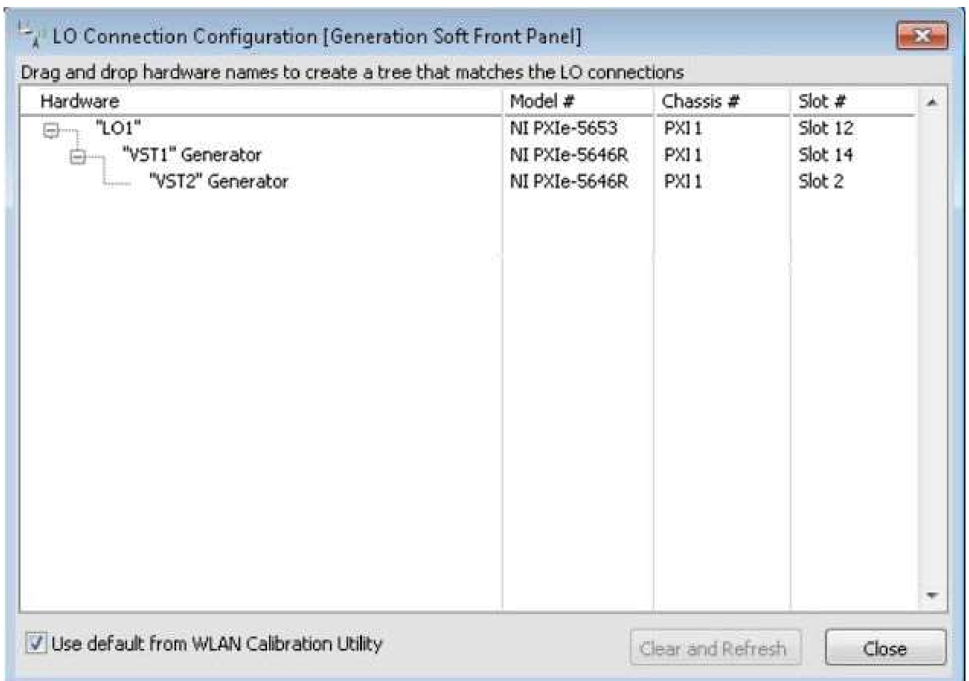


Illustration 3: 2x2 MIMO System with External LO daisy chained across SGs.







VST1 generator is the Master generator in the preceding figure.

NI WLAN Generation Toolkit LabVIEW Reference

Contains information about VIs and properties of the NI WLAN Generation Toolkit.

Using the Standard Functionality for error in Parameters

Many LabVIEW nodes such as VIs contain **error in** parameters you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node. Standard **error in** behavior is as follows

Note Some nodes, such as error handling VIs, contain an **error in** parameter that does not provide standard error in functionality, but that contains an **error in** cluster that is standard.

error in describes error conditions that occur before this node runs. The default is `no error`. If an error occurred before this node runs, the node passes the **error in** value to **error out**. This node runs normally only if no error occurred before this node runs. If an error occurs while this node runs, it runs normally and sets its own error status in **error out**.

Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

The **error in** cluster contains the following cluster elements:

status is TRUE (X) if an error occurred before this node ran or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran. The default is FALSE.

code is the error or warning code. The default is 0. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.

source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning. The default is an empty string.

Using the Standard Functionality for error out Parameters

Many LabVIEW nodes such as VIs contain an **error out** parameter you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node.

Standard **error out** functionality is as follows:

error out contains error information. If **error in** indicates that an error occurred before this VI ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** front panel indicator and select **Explain Error** from the shortcut menu for more information about the error.

error out contains the following cluster elements:

status is TRUE (X) if an error occurred before this node ran or during the running of this node or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran or during the running of this node.

code is the error or warning code. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.

source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning.

Supported Standards for VIs

The following table lists the WLAN Generation VIs applicable for different values of the standard property.

VI	Value of the Standard property										
	80211A/G OFDM	80211J OFDM	80211P OFDM	80211B/G DSSS	80211G DSSSOFDM	80211N MIMO OFDM	80211AC MIMO OFDM	80211AH MIMO OFDM	80211AF MIMO OFDM	80211AX MIMO OFDM	80211BE MIMO OFDM
Main VIs											
niWLANG Open Session	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Configure Multiple Device Synchronization	—	—	—	—	—	√	√	√	√	√	√
niWLANG RFSG Configure Frequency	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Multiple Device Initiate	—	—	—	—	—	√	√	√	√	√	√
niWLANG RF	√	√	√	√	√	√	√	√	√	√	√

SG Configure Frequency (Multiple LO)											
niWLANG Create Single Channel Waveform	√	√	√	√	√	—	—	—	—	—	—
niWLANG Create Multiple Channel Waveform	—	—	—	—	—	√	√	√	√	√	√
niWLANG Reset Session	√	√	√	√	√	√	√	√	√	√	√
niWLANG Close Session	√	√	√	√	√	√	√	√	√	√	√
Configure Generation VIs											
niWLANG Set Standard	√	√	√	√	√	√	√	√	√	√	√
niWLANG Set OFDM Data Rate	√	√	√	—	√	—	—	—	—	—	—
niWLANG Set	—	—	—	√	—	—	—	—	—	—	—

DSSS Data Rate											
niWLANG Set DSSS Preamble Type	—	—	—	√	√	—	—	—	—	—	—
niWLANG Set Channel Bandwidth	√	√	√	—	—	√	√	√	√	√	√
niWLANG Set Number of Transmit Channels	—	—	—	—	—	√	√	√	—	√	√
niWLANG Set MCS Index	—	—	—	—	—	√	√	√	√	√	√
niWLANG Set STBC Index	—	—	—	—	—	√	—	—	—	—	—
niWLANG Set Guard Interval Type	—	—	—	—	—	√	√	√	√	√	√
niWLANG Set Number of Space Time Streams	—	—	—	—	—	—	√	√	—	√	√

niWLANG Set Payload AMPDU Enabled	—	—	—	—	—	√	√	√	√	√	√
niWLANG Set PLCP Frame Format (802_11N)	—	—	—	—	—	√	—	—	—	—	—
niWLANG Set Oversampling Factor	√	√	√	√	√	√	√	√	√	√	√
niWLANG Set Number of Segments	—	—	—	—	—	—	√	—	—	√	—
niWLANG Set Payload Number of MPDUs	—	—	—	—	—	√	√	√	√	√	√
niWLANG Set PPDU Type	—	—	—	—	—	—	√	—	—	√	√
niWLANG Set Number of Users	—	—	—	—	—	—	√	—	—	√	√

niWLANG Set Mapping Matrix Type	—	—	—	—	—	√	√	√	—	√	√
niWLANG Set Preamble Type (802_11AH)	—	—	—	—	—	—	—	√	—	—	—
niWLANG Set Mapping Matrix	—	—	—	—	—	√	√	√	—	√	√
niWLANG Set RU Size	—	—	—	—	—	—	—	—	—	√	√
niWLANG Set RU Offset	—	—	—	—	—	—	—	—	—	√	√
niWLANG Set Idle Interval	√	√	√	√	√	√	√	√	√	√	√
niWLANG Set Number of Frames	√	√	√	√	√	√	√	√	√	√	√
niWLANG Set Auto Headroom Enabled	√	√	√	√	√	√	√	√	√	√	√

niWLANG Set Headroom	√	√	√	√	√	√	√	√	√	√	√
niWLANG Set OFDM Packet Extension Thresholds	—	—	—	—	—	—	—	—	—	√	√
niWLANG Property Node	√	√	√	√	√	√	√	√	√	√	√
Generation Information VIs											
niWLANG Get Mapping Matrix	—	—	—	—	—	√	√	√	—	√	√
niWLANG Get Actual Headroom	√	√	√	√	√	√	√	√	√	√	√
niWLANG Get IQ Waveform Size	√	√	√	√	√	√	√	√	√	√	√
niWLANG Get IQ Rate	√	√	√	√	√	√	√	√	√	√	√
niWLANG Get OFDM Allocat	—	—	—	—	—	—	—	—	—	√	√

ion Map Trace											
niWLANG Get Number of Users from RU Allocation	—	—	—	—	—	—	—	—	—	√	√
Utility VIs											
niWLANG Load Configuration from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Save Configuration to File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Channel Number to Carrier Frequency (802.11abgjpn)	√	√	√	√	√	√	—	—	—	—	
niWLANG Channel Number to Carrier Frequen	—	—	—	—	—	—	√	—	—	—	—

cy (802.11ac)											
niWLANG Channel Number to Carrier Frequency (802.11ah)	—	—	—	—	—	—	—	√	—	—	—
niWLANG Channel Number to Carrier Frequency (802.11af)	—	—	—	—	—	—	—	—	√	—	—
niWLANG Channel Number to Carrier Frequency (802.11ax)	—	—	—	—	—	—	—	—	—	√	—
niWLANG Create Trigger Frame MSDU	—	—	—	—	—	—	—	—	—	√	—
niWLANG Create and Write Wavef	√	√	√	√	√	√	√	√	√	√	√

orms to File											
niWLANG Read Waveform from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Read Waveform Names from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Read RF Blanking Marker Positions from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Read Burst Start Locations from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG Read Burst Stop Locations from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG For	—	—	—	—	—	√	√	√	√	√	√

ce TClk Synchronization											
RFSG Database VIs											
niWLANG RFSG Create and Download Waveform (Single Channel)	√	√	√	√	√	—	—	—	—	—	—
niWLANG RFSG Create and Download Waveform (Multiple Channel)	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Read and Download Waveforms from File	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Configure Script	√	√	√	√	√	√	√	√	√	√	√

niWLANG RFSG Clear Database	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Configure Power Level	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Insert RF Blanking Marker Positions	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Store IQ Rate	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Store PAPR	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Store RF Blanking Marker Positions	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Sto	√	√	√	√	√	√	√	√	√	√	√

re Waveform Size											
niWLANG RFSG Store Burst Start Locations	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Store Burst Stop Locations	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve IQ Rate (Script)	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve Minimum PAPR (Script)	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve IQ Rate (Waveform)	√	√	√	√	√	√	√	√	√	√	√

niWLANG RFSG Retrieve PAPR (Waveform)	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve RF Blanking Marker Positions (Waveform)	√	√	v	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve Burst Start Locations	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve Burst Stop Locations	√	√	√	√	√	√	√	√	√	√	√
niWLANG RFSG Retrieve Waveform Size	√	√	√	√	√	√	√	√	√	√	√

Generation VIs

Use the VIs on this palette to set various parameters used for generating WLAN standard-compliant bursts and configuring the NI RF vector signal generators.

Palette Object	Description
niWLANG Open Session	Looks up an existing WLAN generation session using the **session name** parameter and returns the refnum that you can pass to subsequent niWLAN Generation VIs and the niWLANG Property Node. If the look up fails, the niWLANG Open Session VI creates a new niWLAN generation session and returns a new refnum.
niWLANG RFSG Configure Multiple Device Synchronization	Configures multiple NI RF vector signal generators for sharing the local oscillator (LO), configures the Reference Clock settings, and synchronizes multiple devices.
niWLANG RFSG Configure Frequency	Configures the frequency on NI RF vector signal generators, and NI RF synthesizers used as external LO devices. It also configures the Carrier Frequency property.
niWLANG RFSG Multiple Device Initiate	Commits settings to hardware, waits for hardware settling, and starts multi-device synchronized generation.
niWLANG Create Waveform	Creates WLAN I/Q waveform data according to the properties that you specify. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set the **reset?** parameter to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** parameter is TRUE.
niWLANG Reset Session	Resets all the properties of the session to their default values.

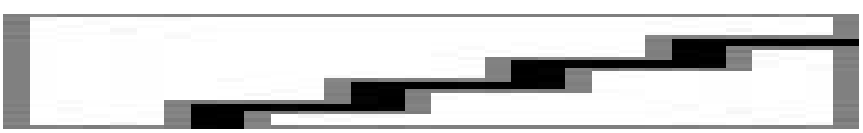
niWLANG Close Session	Closes the niWLAN generation session and releases resources associated with that session. Call this VI once for each unique named session that you have created.

Subpalette	Description
Configure Generation	Use the VIs on this palette to configure various parameters for WLAN signals. You can use the niWLANG Property Node VI to configure additional parameters.
Generation Information	Use the VIs on this palette to get generator parameters used to generate a waveform.
Utility	Use the VIs on this palette to reset all properties, obtain the error code returned by WLAN Generation VIs, save and load configuration files, and calculate carrier frequencies according to the IEEE 802.11 channel numbering scheme.
RFSG Database	Use the VIs on this palette to configure, store, or retrieve information from the RFSG database.

niWLANG Open Session (VI)

Owning Palette: Generation VIs

Looks up an existing WLAN generation session using the **session name** parameter and returns the refnum that you can pass to subsequent niWLAN Generation VIs and the niWLANG Property Node. If the look up fails, the niWLANG Open Session VI creates a new niWLAN generation session and returns a new refnum.



session name specifies the name of the session that you are looking up or creating. If a session with the same name already exists, this VI returns a reference to that session. To get the reference to an already-opened session **x**, specify **x** as the session name.

You can obtain the reference to an existing session multiple times if you have not called the niWLANG Close Session VI in that session. You do not need to close the session multiple times. To create an unnamed session, pass an empty string to the **session name** parameter or leave this parameter unwired.

Tip NI recommends that you call the niWLANG Close Session VI for each unique named instance of the niWLANG Open Session VI or each instance of the niWLANG Open Session VI with an unnamed session.

I32

toolkit compatibility version specifies the version of the toolkit to which the current version of the toolkit is compatible. If the behavior of the toolkit changes in a new version, use this parameter to specify that you want to continue using the behavior of the previous release. The default value is **5.0.0**.

1.0.0 (10000)	Specifies that the toolkit exhibits version 1.0.0 behavior, and all new features in later releases are unavailable. Select this option if you purchased version 1.0.0 and want to maintain functional

	behavior. Refer to the **NI WLAN Generation Toolkit Readme** for a list of changes between versions 1.0.0 and later versions.
2.0.0 (20000)	Specifies that the toolkit exhibits version 2.0.0 behavior, and all new features in later releases are unavailable. Select this option if you purchased version 2.0.0 and want to maintain functional behavior. Refer to the **NI WLAN Generation Toolkit Readme** for a list of changes between versions 2.0.0 and 3.0.0.
3.0.0 (30000)	Specifies that the toolkit exhibits version 3.0.0 behavior. Select this option if you want 3.0.0 behavior and access to new features and bug fixes. Refer to the **NI WLAN Generation Toolkit Readme** for a list of changes between versions 2.0.0 and 3.0.0.
4.0.0 (40000)	Specifies that the toolkit exhibits version 4.0.0 behavior. Select this option if you want 4.0.0 behavior and access to new features

	and bug fixes. Refer to the **NI WLAN Generation Toolkit Readme** for a list of changes between versions 3.0.0 and 4.0.0.
5.0.0 (50000)	Specifies that the toolkit exhibits version 5.0.0 behavior. Select this option if you want 5.0.0 behavior and access to new features and bug fixes. Refer to the **NI WLAN Generation Toolkit Readme** for a list of changes between versions 4.0.0 and 5.0.0.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session returns the niWLAN generation session refnum. Use this parameter to configure the behavior and operation of the appropriate NI WLAN Generation Toolkit VIs that accept the **niwlan generation session** refnum as an input.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid

possible memory leak issues.

new session? indicates whether the VI creates a new session.

TRUE	Indicates that the VI creates a new session.
FALSE	Indicates that the VI returns a reference to an existing session.

error out contains error information. This output provides standard error out functionality.

Configure Generation

Owning Palette: Generation VIs

Use the VIs on this palette to configure various parameters for WLAN signals. You can use the niWLANG Property Node VI to configure additional parameters.

Note The help topics for some VIs do not contain parameter details. Refer to the corresponding property help topics for more information about these VIs.

Palette Object	Description
niWLANG Set Standard	Sets the Standard property.
niWLANG Set OFDM Data Rate	Sets the OFDM Data Rate property.
niWLANG Set DSSS Data Rate	Sets the DSSS Data Rate property.
niWLANG Set DSSS Preamble Type	Sets the DSSS Preamble Type property.
niWLANG Set Channel Bandwidth	Sets the Channel Bandwidth property.
niWLANG Set Number of Transmit Channels	Sets the Number of Transmit Channels property.

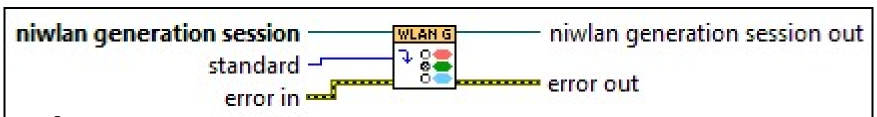
niWLANG Set MCS Index	Sets the MCS Index property.
niWLANG Set STBC Index	Sets the STBC Index property.
niWLANG Set Number of Space Time Streams	Sets the Number of Space Time Streams propert y.
niWLANG Set Payload AMPDU Enabled	Sets the A-MPDU Enabled property.
niWLANG Set PLCP Frame Format (802_11N)	Sets the 80211n PLCP Frame Format property.
niWLANG Set Oversampling Factor	Sets the Oversampling Factor property.
niWLANG Set Guard Interval Type	Sets the Guard Interval Type property.
niWLANG Set Number of Segments	Sets the Number of Segments property.
niWLANG Set Payload Number of MPDUs	Sets the Payload Number of MPDUs property.
niWLANG Set PPDU Type	Sets the PPDU Type property.
niWLANG Set Number of Users	Sets the Number of Users property.
niWLANG Set Preamble Type (802_11AH)	Sets the 80211ah Preamble Type property.
niWLANG Set Mapping Matrix Type	Sets the Mapping Matrix Type property.
niWLANG Set RU Size	Sets the RU Size property.
niWLANG Set RU Offset	Sets the RU Offset property.
niWLANG Set Preamble Type (802_11AH)	Sets the 80211ah Preamble Type property.
niWLANG Set Mapping Matrix	Specifies the matrix for mapping space-time str eams to the transmit channels as defined in sect ion 20.3.11.10.1 of **IEEE Standard 802.11n-200 9**. The toolkit ignores this VI, if the Mapping Matrix Type property is not set to User Defined .
niWLANG Set Idle Interval	Sets the Idle Interval property.
niWLANG Set Number of Frames	Sets the Number of Frames property.
niWLANG Set Auto Headroom Enabled	Sets the Auto Headroom Enabled property.

niWLANG Set Headroom	Sets the Headroom property.
niWLANG Set OFDM Packet Extension Thresholds	Configures the packet extension (PE) thresholds that determine the nominal packet padding of the 802.11ax signal. This VI configures the thresholds for each resource unit (RU) size and each space-time stream of the 802.11ax DUT. You must configure this table based on the nominal packet padding requirements of the DUT.
niWLANG Property Node	Sets (writes) or gets (reads) niWLAN Generation properties.

niWLANG Set Standard (VI)

Owning Palette: Configure Generation

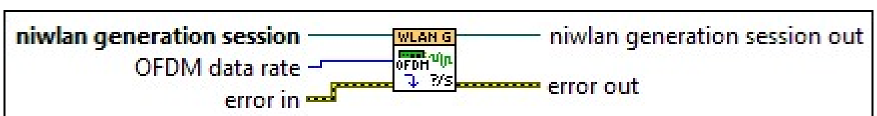
Sets the Standard property.



niWLANG Set OFDM Data Rate (VI)

Owning Palette: Configure Generation

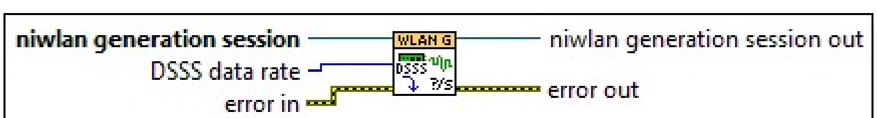
Sets the OFDM Data Rate property.



niWLANG Set DSSS Data Rate (VI)

Owning Palette: Configure Generation

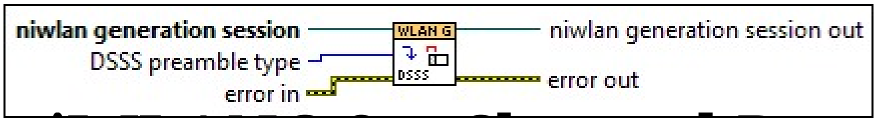
Sets the DSSS Data Rate property.



niWLANG Set DSSS Preamble Type (VI)

Owning Palette: Configure Generation

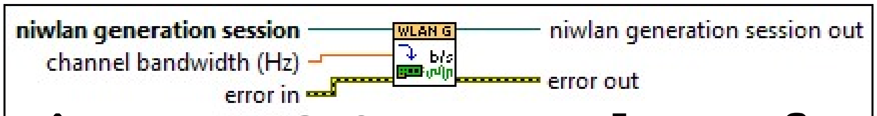
Sets the DSSS Preamble Type property.



niWLANG Set Channel Bandwidth (VI)

Owning Palette: Configure Generation

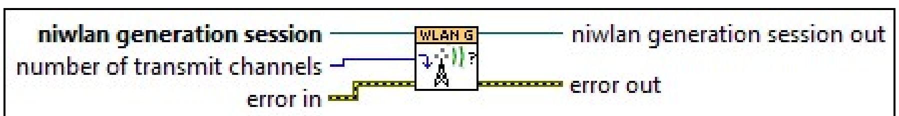
Sets the Channel Bandwidth property.



niWLANG Set Number of Transmit Channels (VI)

Owning Palette: Configure Generation

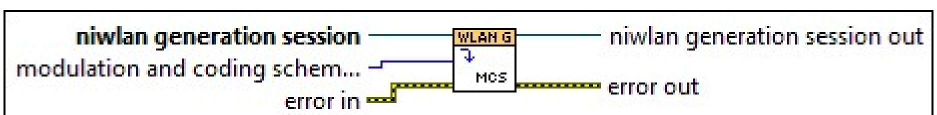
Sets the Number of Transmit Channels property.



niWLANG Set MCS Index (VI)

Owning Palette: Configure Generation

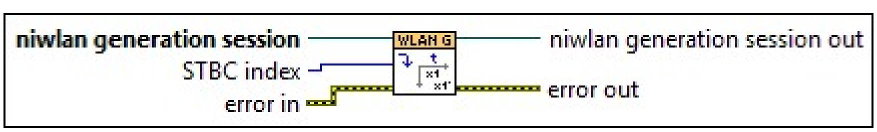
Sets the MCS Index property.



niWLANG Set STBC Index (VI)

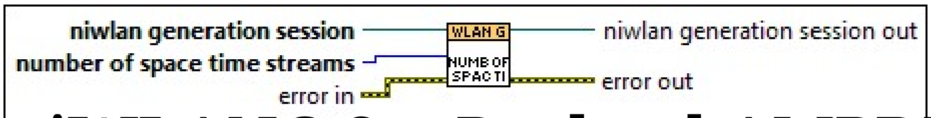
Owning Palette: Configure Generation

Sets the STBC Index property.



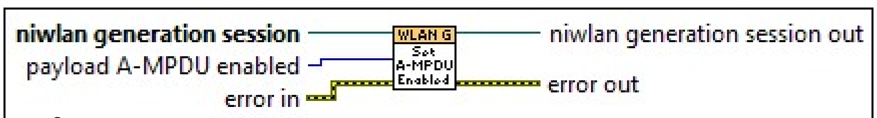
niWLANG Set Number of Space Time Streams (VI)

Owning Palette: Configure Generation
Sets the Number of Space Time Streams property.



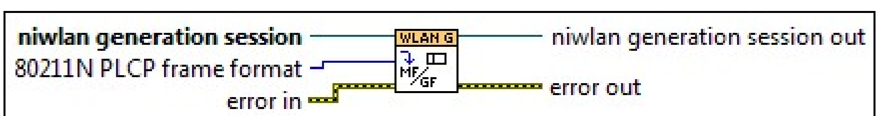
niWLANG Set Payload AMPDU Enabled (VI)

Owning Palette: Configure Generation
Sets the A-MPDU Enabled property.



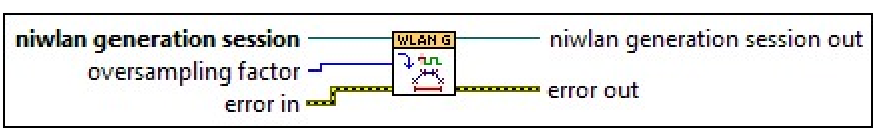
niWLANG Set PLCP Frame Format (802_11N) (VI)

Owning Palette: Configure Generation
Sets the 80211n PLCP Frame Format property.



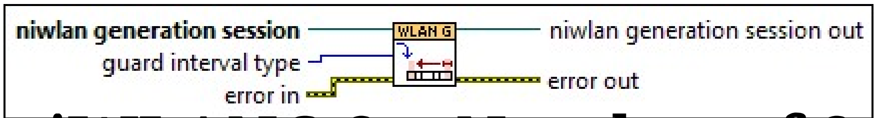
niWLANG Set Oversampling Factor (VI)

Owning Palette: Configure Generation
Sets the Oversampling Factor property.



niWLANG Set Guard Interval Type (VI)

Owning Palette: Configure Generation
Sets the Guard Interval Type property.

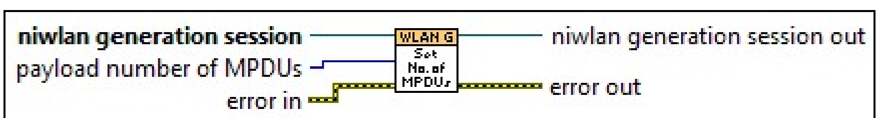


niWLANG Set Number of Segments (VI)

Owning Palette: Configure Generation
Sets the Number of Segments property.

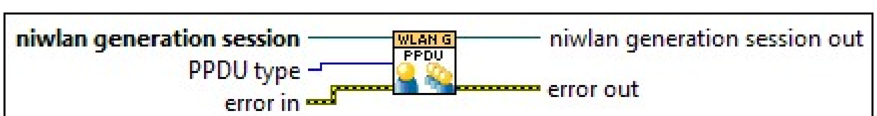
niWLANG Set Payload Number of MPDUs (VI)

Owning Palette: Configure Generation
Sets the Payload Number of MPDUs property.



niWLANG Set PPDU Type (VI)

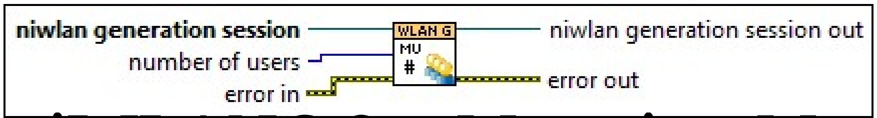
Owning Palette: Configure Generation
Sets the PPDU Type property.



niWLANG Set Number of Users (VI)

Owning Palette: Configure Generation

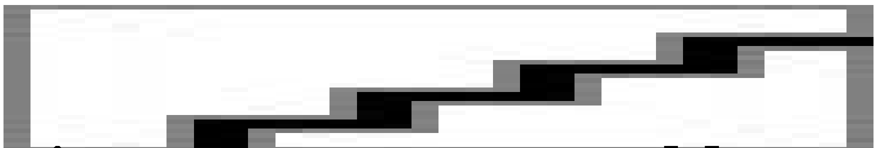
Sets the Number of Users property.



niWLANG Set Mapping Matrix Type (VI)

Owning Palette: Configure Generation

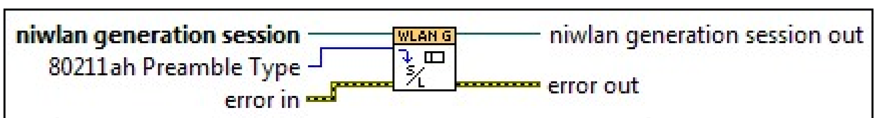
Sets the Mapping Matrix Type property.



niWLANG Set Preamble Type (802_11AH) (VI)

Owning Palette: Configure Generation

Sets the 80211ah Preamble Type property.

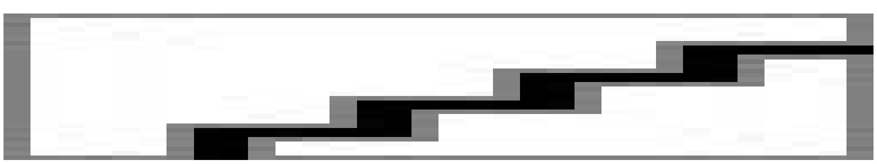


niWLANG Set Mapping Matrix (VI)

Owning Palette: Configure Generation

Specifies the matrix for mapping space-time streams to the transmit channels as defined in section 20.3.11.10.1 of **IEEE Standard 802.11n-2009**. The toolkit ignores this VI, if the Mapping Matrix Type property is not set to **User Defined**.

Details



niwlan generation session specifies the niWLAN generation session refnum.

active channel specifies the active channel string.

You must use the following active channel string formats when you set the PPDU Type property to **Trigger-Based PPDU**.

Spatial Mapping Mode property value	Active channel string format
Common	"" (empty string)
User Specific	"user**x**"

mapping matrix specifies the matrix for mapping spatial streams to the transmit channels.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

Details

The dimensions of the matrix must be **N_Tx** * (**N_STS** + **N_ESS**) if you set the Standard property to 80211N MIMOOFDM, and **N_Tx** * **N_STS** if you set the Standard property to 80211AC MIMOOFDM, 80211AH MIMOOFDM, or 80211AX MIMOOFDM.

where,	**N_Tx** is the number of transmit channels
	N_STS is the number of space-time streams. If you set the Standard property to 80211N MIMOOFDM, **N_STS** is determined by the MCS index and the STBC index
	N_ESS is the number of extension spatial streams

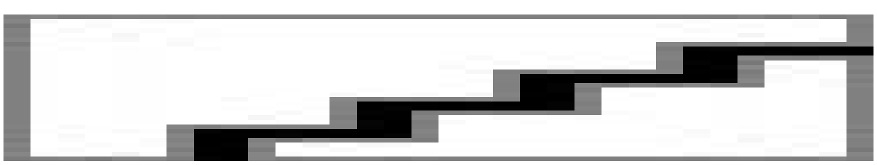
For one-to-one mapping (direct mapping), **N_Tx** = **N_SS** and **N_ESS** = **0**. For one-to-many mapping (spatial expansion), **N_Tx** >= (**N_SS** + **N_ESS**).

If you set the Standard property to 80211AX MIMOOFDM and the PPDU Type property to Trigger-Based PPDU, the **N STS** value is dependent on the Spatial Mapping Mode property.

Spatial Mapping Mode property value	N_STS value
Common	Maximum number of space time streams across Resource Units (RUs)
User Specific	Number of space time streams for the specified user

niWLANG Set RU Size (VI)

Owning Palette: Configure Generation
Sets the RU Size property.

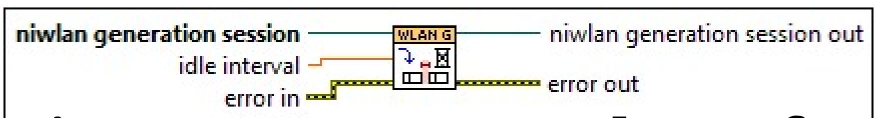


niWLANG Set RU Offset (VI)

Owning Palette: Configure Generation
Sets the RU Offset property.

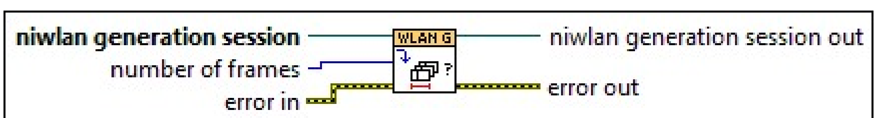
niWLANG Set Idle Interval (VI)

Owning Palette: Configure Generation
Sets the Idle Interval property.



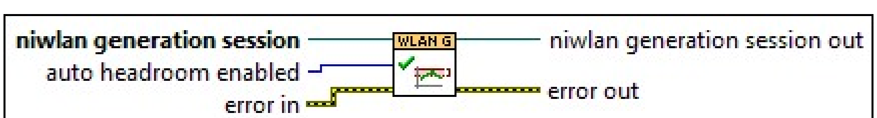
niWLANG Set Number of Frames (VI)

Owning Palette: Configure Generation
Sets the Number of Frames property.



niWLANG Set Auto Headroom Enabled (VI)

Owning Palette: Configure Generation
Sets the Auto Headroom Enabled property.



niWLANG Set Headroom (VI)

Owning Palette: Configure Generation

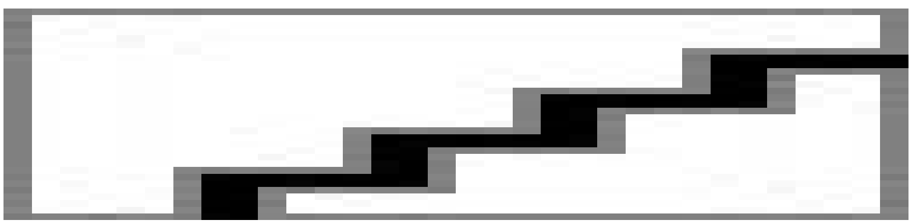
Sets the Headroom property.

niWLANG Set OFDM Packet Extension Thresholds (VI)

Owning Palette: Configure Generation

Configures the packet extension (PE) thresholds that determine the nominal packet padding of the 802.11ax signal. This VI configures the thresholds for each resource unit (RU) size and each space-time stream of the 802.11ax DUT. You must configure this table based on the nominal packet padding requirements of the DUT.

Details



niwlan generation session specifies the niWLAN generation session refnum.

RU size specifies the size of the RU in terms of the number of subcarriers. The default value is "" (empty array).

number of space time streams specifies the number of space-time streams for the corresponding RU size. The default value is "" (empty array).

PPET8 specifies the 8 microsecond mode PE threshold values. The default value is "" (empty array).

PPET16 specifies the 16 microsecond mode PE threshold values. The default value is "" (empty array).

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

> **Note** Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

Details

The PE field of 802.11ax signal provides additional processing time to the receiver to decode the last symbol. The possible durations of the PE field are 0 microseconds, 4 microseconds, 8 microseconds, 12 microseconds, or 16 microseconds. The PE duration is determined by both the pre-FEC padding factor of the data field and the nominal packet padding requested by the recipient; and the modulation scheme of the current PPDU. The nominal packet padding as defined by the HE Capabilities element are 0 microseconds, 8 microseconds and 16 microseconds, which can be specified using this VI.

The nominal packet padding is computed by comparing the constellation index and the threshold values. Packet extension device capability threshold values are defined for all RU sizes greater than or equal to 242 tones. No thresholds are defined for an RU size less than 242 tones. The supported constellations are assigned with a unique index value as shown in following table.

Modulation Scheme	Constellation Index Value
BPSK	0
QPSK	1
16-QAM	2
64-QAM	3
256-QAM	4
1024-QAM	5
4096-QAM	6
None	7

The PE can be defined in two modes.

1. Nominal PE mode, 8 microseconds: PE duration is 0, 0, 4 and 8 microseconds for pre-FEC padding factor of 1, 2, 3 and 4 respectively
2. Nominal PE mode, 16 microseconds: PE duration is 4, 8, 12 and 16 microseconds for pre-FEC padding factor of 1, 2, 3 and 4 respectively

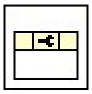
The nominal packet padding is calculated from the threshold values and the constellation index for the specified MCS index as follows.

- If the constellation is greater than or equal to PPET16, you must apply the value of the maximum **PPET16** parameter, or if the constellation is greater than or equal to PPET8, you must apply the value of the maximum **PPET8** parameter; otherwise, there is no packet extension.
- If no PE is required for all constellations, set **PPET8** and **PPET16** to "" (empty array).
- If only the nominal PE 8 microseconds mode is required, set **PPET16** to be "" (empty array), and **PPET8** to be the constellation at which max PE 8 microseconds mode starts.
- If only the nominal PE 16 microseconds mode is required, set **PPET16** to be the constellation at which max PE 16 microseconds mode starts, and **PPET8** to be "" (empty array).

niWLANG Property Node (VI)

Owning Palette: Configure Generation
Sets (writes) or gets (reads) niWLAN Generation properties.

Some niWLAN Generation properties are channel based. When a property is channel based, you must specify an active channel before setting or getting properties.



niwlan generation session specifies the niWLAN generation session refnum.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

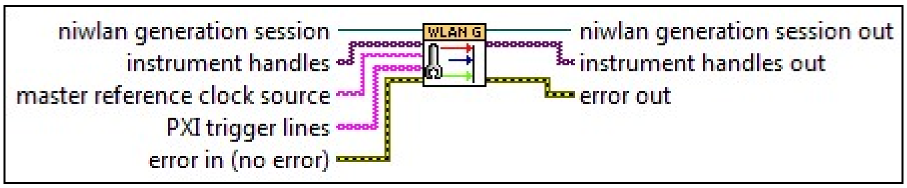
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Configure Multiple Device Synchronization (VI)

Owning Palette: Generation VIs

Configures multiple NI RF vector signal generators for sharing the local oscillator (LO), configures the Reference Clock settings, and synchronizes multiple devices.

Details



niwlan generation session specifies the niWLAN generation session refnum.

instrument handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

master reference clock source specifies the device Reference Clock to configure on the master NI RF vector signal generator. The default value is **PXI_CLK**.

OnboardClock (OnboardClock)	Uses the onboard Reference Clock as the clock source.
RefIn (RefIn)	Uses the clock signal present at the front panel REF IN connector as the clock source.
PXI_CLK (PXI_CLK)	Uses the PXI_CLK signal, which is present on the PXI

	backplane, as the clock source.
ClkIn (ClkIn)	Uses the clock signal present at the front panel CLK IN connector as the clock source.

PXI trigger lines specifies the array of trigger lines used for distribution of synchronized trigger signals. The default value is "" (empty string).

The valid values are PXI_Trig0, PXI_Trig1, PXI_Trig2, PXI_Trig3, PXI_Trig4, PXI_Trig5, PXI_Trig6, PXI_Trig7, and PFI0.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

instrument handles out passes an array of RFSG instrument sessions to the next VI.

error out contains error information. This output provides standard error out functionality.

Details

For PXIe-5673/5673E/5840/5841/5830/5831 RF vector signal generators, this VI configures LO sharing and clock settings in daisy-chained manner.

For PXIe-5644, PXIe-5645, and PXIe-5646 RF vector signal transceiver (VST) devices, this VI configures daisy-chained LO sharing and low-level properties for multidevice synchronization. These VSTs do not support daisy-chained Reference Clocks for synchronization. You must set the master Reference Clock source to PXI_CLK for these devices.

For PXIe-5840/5841/5830/5831 RF VST devices, this VI configures daisy-chained LO sharing and Reference Clocks for synchronization, and for PXIe-5820 baseband I/Q VST devices, this VI configures daisy-chained Reference Clocks for synchronization. For additional information, refer to the **Synchronization Using NI-RFSA and NI-RFSG** topic in the **NI RF Signal Generators Help**.

This VI assumes that the devices are interconnected in the same order as the elements in the instrument handles array. In time synchronization, assume the first device is the master device, and the remaining devices the slave devices. The VI divides the devices into two sets when the number of segments is two. In LO sharing, the first device in each set is assumed to be the master device, and the remaining devices in the set are assumed to be slave devices. To interconnect multiple devices, refer to the following topics in the **NI RF Signal Generators Help**:

- **Interconnecting Multiple PXIe-5673E Modules**
- **Interconnecting Multiple PXIe-5673 Modules**
- **Interconnecting Multiple PXIe-5644 RF Channels (Homogeneous Channel Types)**
- **Interconnecting Multiple PXIe-5645 RF Channels (Homogeneous Channel Types)**
- **Interconnecting Multiple PXIe-5646 RF Channels (Homogeneous Channel Types)**
- **Interconnecting Multiple PXIe-5840 RF Channels (Homogeneous Channel Types)**
- **Interconnecting Multiple PXIe-5841 RF Channels (Homogeneous Channel Types)**

If you use PXIe-5644, PXIe-5645, or PXIe-5646 RF VSTs, the VI completes the following actions:

- The VI sets the niRFSG Reference Clock Source to the value you specify in the **master reference clock source** parameter.
- If you set the LO Sharing Enabled property to **True**, the VI completes the following actions.
  - For the master device, the VI exports the LO by setting the niRFSG LO Out Enabled property to **True**. The VI also reads the niRFSG Upconverter Center Frequency property so that the value can be set on the slave devices.
  - For the slave devices, the VI sets the niRFSG LO Source property to **LO IN** and the niRFSG Upconverter Center Frequency property to the value read from the master device. With the exception of the last slave device, the VI also exports the LO to the next device in the daisy chain by setting the niRFSG LO Out Enabled property to **True**.
- The VI configures the trigger synchronization for the master and slave devices.
  - For the master device, the VI completes the following actions:
    - Sets the niRFSG Sync Start Trigger Master property to **True**. For PXIe-5646R, the VI sets the niRFSG Sync Sample Clock Master property to **True**.
    - Sets the niRFSG Sync Start Dist Line property to the first element of the PXI trigger lines. For PXIe-5646, the VI sets the niRFSG Sync Sample Clock Dist Line property to the second element of the **PXI trigger lines** array.
    - Calls the niRFSG Commit VI to commit the synchronization settings to the device.
  - For the slave devices, the VI completes the following actions:
    - Sets the niRFSG Sync Start Trigger Master property to **False**. For PXIe-5646, the VI sets the niRFSG Sync Sample Clock Master Property to **False**.

        - Sets the niRFSG Sync Start Dist Line property to the first element of the PXI trigger lines. For PXIe-5646, the VI sets the niRFSG Sync Sample Clock Dist Line property to the second element of the **PXI trigger lines** array.
        - Sets the niRFSG Start Trigger Type property to **Digital Edge** and the niRFSG Start Trigger Digital Edge Source property to **sync_start**.

If you use PXIe-5673/5673E RF vector signal generator, the VI completes the following actions:

- The VI configures the devices for daisy-chained Reference Clock synchronization.
    - For the master device, the VI sets the niRFSG Reference Clock Source property to the value you specify in **master reference clock source** and sets the niRFSG Reference Clock Export Output Terminal property to **ClkOut**.
    - For the slave devices, the VI sets the niRFSG Reference Clock Source property to **ClkIn** and sets the niRFSG Reference Clock Export Output Terminal property to **ClkOut**.
    - If you set the LO Sharing Enabled Property to **True**, the VI does the following actions:
        - For the master device, the VI exports the LO by setting the niRFSG LO Export Enabled property to True. The VI also reads the niRFSG LO Out Power(dBm) property so that it can be set on the next device in the daisy chain.
        - For the slave devices, the VI sets the niRFSG LO In Power (dBm) property to the value read from the master and reads the niRFSG LO Out Power (dBm) property. The VI also exports the LO to the next device in the daisy chain by setting the niRFSG LO Out Enabled property to **True**.

If you use PXIe-5840, PXIe-5841, PXIe-5830, or PXIe-5831 RF vector signal generators, the VI completes the following actions:

- For reference clock synchronization:
  - The VI configures all the devices to use PXI_CLK as reference clock if you set the **master reference clock source** parameter to **PXI_CLK**.
  - The VI configures the devices for daisy-chained reference clock synchronization, if you set the **master reference clock source** parameter to a value other than **PXI_CLK**.
    - For the master device, the VI sets the niRFSG Reference Clock Source property to the value you specify in the **master reference clock source** parameter and sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.
    - For the slave devices, the VI sets the niRFSG Reference Clock Source property to **RefIn**. Except for the last slave device, this VI also sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.
- If you set the LO Sharing Enabled property to **True**, the VI completes the following actions:
  - For the master device, the VI exports the LO by setting the niRFSG LO Out property to **True**. The VI also reads the niRFSG Upconverter Center Frequency property so that the value can be set on slave devices.
  - For the slave devices, the VI sets the niRFSG LO Source property to **LO_In** and the niRFSG Upconverter Center Frequency property to the value read from the master device. Except for the last slave device, this VI also exports the LO to the next device in the daisy chain by setting the niRFSG LO Out property to **True**.

If you use PXIe-5820 baseband I/Q vector signal generators, the VI completes the following actions:

- For reference clock synchronization:
  - The VI configures all the devices to use PXI_CLK as reference clock if you set the **master reference clock source** parameter to **PXI_CLK**.

- The VI configures the devices for daisy-chained reference clock synchronization, if you set the **master reference clock source** parameter to a value other than **PXI_CLK**.
    - For the master device, the VI sets the niRFSG Reference Clock Source property to the value you specify in the **master reference clock source** parameter and sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.
    - For the slave devices, the VI sets the niRFSG Reference Clock Source property to **RefIn**. Except for the last slave device, this VI also sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.

If you use PXIe-5830 or PXIe-5831 RF vector signal generators, the VI completes the following actions:

- For reference clock synchronization:
    - The VI configures all the devices to use PXI_CLK as reference clock if you set the **master reference clock source** parameter to **PXI_CLK**.
    - The VI configures the devices for daisy-chained reference clock synchronization, if you set the **master reference clock source** parameter to a value other than **PXI_CLK**.
        - For the master device, the VI sets the niRFSG Reference Clock Source property to the value you specify in the **master reference clock source** parameter and sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.
        - For the slave devices, the VI sets the niRFSG Reference Clock Source property to **RefIn**. Except for the last slave device, this VI also sets the niRFSG Reference Clock Exported Terminal property to **RefOut**.
- If you set the LO Sharing Enabled property to **True**, the VI completes the following actions:
    - For the master device, the VI exports the LO by setting the niRFSG LO Out property to **True**. The VI also reads the niRFSG Upconverter Center Frequency property so that the value can be set on slave devices.

- For the slave devices, the VI sets the niRFSG LO Source property to **LO_In** and the niRFSG Upconverter Center Frequency property to the value read from the master device. Except for the last slave device, this VI also exports the LO to the next device in the daisy chain by setting the niRFSG LO Out property to **True**.

niWLANG RFSG Configure Frequency (VI)

Owning Palette: Generation VIs

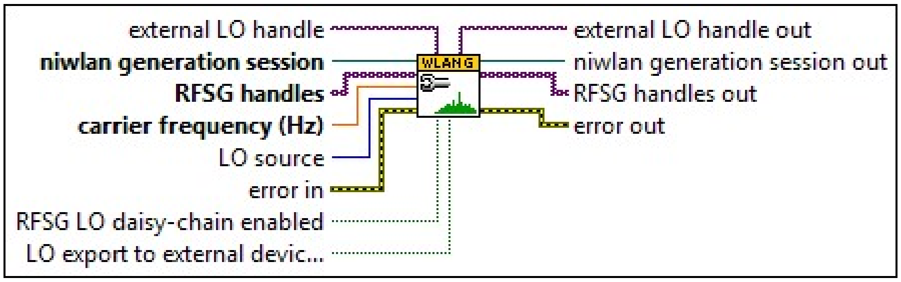
Configures the frequency on NI RF vector signal generators, and NI RF synthesizers used as external LO devices. It also configures the Carrier Frequency property.

You must ensure that after a frequency configuration change, the output of the external LO device settles before generating the WLAN signal. You must use one of the following programming flows:

- Before calling the niWLANG RFSG Configure Frequency VI, if the external LO device is in generation state, ensure that it is brought in configuration state, by stopping signal generation. Initiate signal generation on the external LO device after calling the niWLANG RFSG Configure Frequency VI.
- You may choose to do on-the-fly frequency change on the external LO device while the device is in generation state. You must allow its output to settle by calling the niRFSG Wait Until Settled VI after the niWLANG RFSG Configure Frequency VI.

niWLANG RFSG Configure Frequency (Single LO)

Configures the frequency on NI RF vector signal generators and NI RF synthesizers. NI RF synthesizers are used as external local oscillator (LO) devices. This VI also configures LO frequency offset based on the LO Frequency Offset Mode and LO Frequency Offset properties. It also configures the Signal Bandwidth and IQ Rate properties on NI RF vector signal generators and the Carrier Frequency property.



niwlan generation session specifies the niWLAN generation session refnum.

RFSG handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

carrier frequency specifies the carrier frequency used to generate signals. This value is expressed in Hz.

LO source specifies whether to use the internal or the external LO source. This value is applicable only for the first RFSG device within a set if the **RFSG LO daisy-chain enabled** parameter is set to **True**. The default value is **Onboard**.

Onboard (0)	Uses an internal LO as the LO source.
External (1)	Uses an external LO as the LO source.
SG SA Shared (2)	Shares the internal RFSG LO between RFSG and RFSA.

external LO handle identifies the instrument session of the external LO device. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

RFSG LO daisy-chain enabled specifies whether to export the LO signal from one RFSG device to the next. The default value is FALSE.

LO export to external devices enabled specifies whether to export the LO signal from each RFSG device on its LO OUT terminal, which you can use to share the LO signal with an external device. An example of an external device would be an RFSA device. The default value is FALSE.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

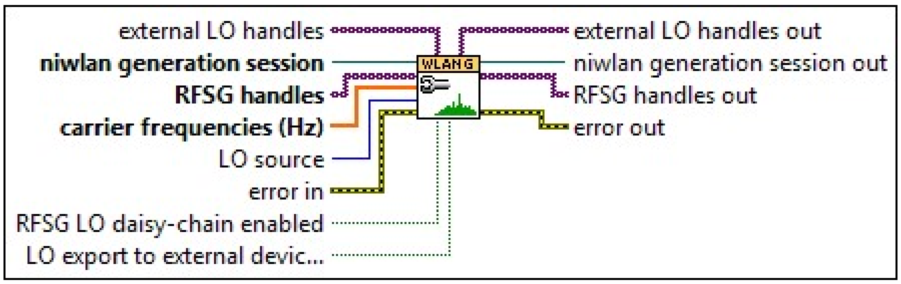
RFSG handles out passes an array of RFSG instrument sessions to the next VI.

external LO handle out passes a reference from the external LO session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Configure Frequency (Multiple LO)

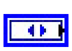
Configures the frequency on NI RF vector signal generators and NI RF synthesizers. NI RF synthesizers are used as external LO devices. This VI also configures LO frequency offset based on the LO Frequency Offset Mode and LO Frequency Offset properties. This VI equally divides the NI RF vector signal generators and NI RF synthesizers into sets such that each set corresponds to one carrier frequency. The number of sets is equal to the size of the carrier frequencies array. In each set, the first NI RF vector signal generator is used as the master device for LO daisy chaining. It also configures the Signal Bandwidth and IQ Rate properties on NI RF vector signal generators and the Carrier Frequency property.



niwlan generation session specifies the niWLAN generation session refnum.

RFSG handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

carrier frequencies specifies an array of the carrier frequencies used to generate signals. This value is expressed in Hz.



LO source specifies whether to use the internal or the external LO source. This value is applicable only for the first RFSG device within a set if the **RFSG LO daisy-chain enabled** parameter is set to **True**. The default value is **Onboard**.

Onboard (0)	Uses an internal LO as the LO source.
External (1)	Uses an external LO as the LO source.
SG SA Shared (2)	Shares the internal RFSG LO between RFSG and RFSA.

[I/O] **external LO handles** identifies the instrument sessions of external LO devices. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

TF **RFSG LO daisy-chain enabled** specifies whether to export the LO signal from one RFSG device to the next. The default value is FALSE.

TF **LO export to external devices enabled** specifies whether to export the LO signal from each RFSG device on its LO OUT terminal, which you can use to share the LO signal with an external device. An example of an external device would be an RFSA device. The default value is FALSE.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid

possible memory leak issues.

RFSG handles out passes an array of RFSG instrument sessions to the next VI.

external LO handles out passes an array of references from the external LO sessions to the next VI.

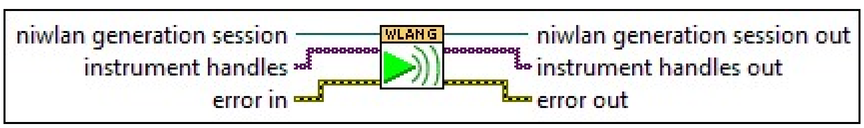
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Multiple Device Initiate (VI)

Owning Palette: Generation VIs

Commits settings to hardware, waits for hardware settling, and starts multi-device synchronized generation.

Details



niwlan generation session specifies the niWLAN generation session refnum.

instrument handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

instrument handles out passes an array of RFSG instrument sessions to the next VI.

error out contains error information. This output provides standard error out functionality.

Details

If you use PXIe-5644, PXIe-5645, or PXIe-5646, this VI performs the following steps:

- Commits the device settings to the master device by calling the niRFSG Commit VI.
- Initializes generation first for slave devices and then for the master device by calling the niRFSG Initiate VI in a loop.

If you use PXIe-5673/5673E, PXIe-5840, PXIe-5841, PXIe-5820, PXIe-5830, or PXIe-5831, this VI performs the following steps:

- If the number of devices required is more than 1, this VI uses the niWLANG RFSG Configure TClk For Homogeneous Triggers VI, niWLANG RFSG Synchronize TClk VI, and the niTClk Initiate VI for synchronized generation. Refer to the NI-TClk Synchronization section of **NI RF Vector Signal Generators** help for more information about the niTClk VIs.

- Otherwise, this VI calls the niRFSG Initiate VI for the first device.

niWLANG Create Waveform (VI)

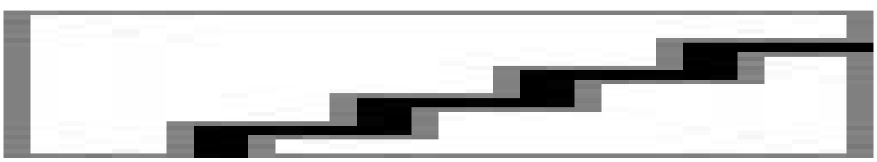
Owning Palette: Generation VIs

Creates WLAN I/Q waveform data according to the properties that you specify. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set the **reset?** parameter to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** parameter is TRUE.

niWLANG Create Single Channel Waveform

Creates WLAN I/Q data and returns the data as a complex waveform. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set **reset?** to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** output is TRUE.

Note Use this instance if the Standard property is set to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS** or **80211G DSSSOFDM**.



niwlan generation session specifies the niWLAN generation session refnum.

reset? specifies whether to reset data. The default value is FALSE.

TRUE	Resets the data. Set thi s parameter to TRUE af ter you have generated the required number o f frames or if you want

	to reset the pseudonoise (PN) seed.
FALSE	Does not reset the data.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

output waveform returns the WLAN I/Q data.

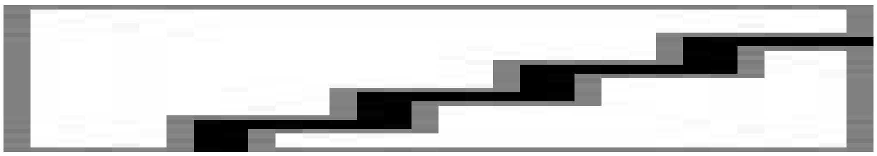
generation done? indicates whether the VI has generated all the data.

error out contains error information. This output provides standard error out functionality.

niWLANG Create Multiple Channel Waveform

Creates WLAN I/Q data for multiple channels and returns the data as an array of complex waveforms. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set the **reset?** parameter to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** parameter is TRUE.

Note Use this instance if the Standard property is set to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**.



niwlan generation session specifies the niWLAN generation session refnum.

reset? specifies whether to reset data. The default value is FALSE.

TRUE	Resets the data. Set this parameter to TRUE after you have generated the required number of frames or if you want to reset the pseudonoise (PN) seed.
FALSE	Does not reset the data.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close

Session VI before the completion of execution to avoid possible memory leak issues.

output waveform returns an array of WLAN I/Q data.

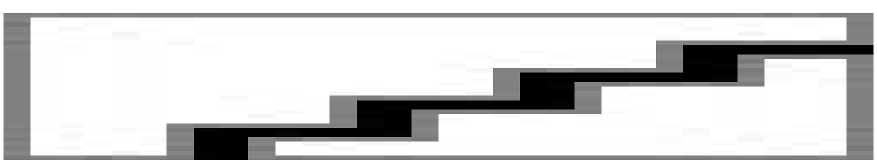
generation done? indicates whether the VI has generated all the data.

error out contains error information. This output provides standard error out functionality.

niWLANG Create Single Channel Waveform (IQ Complex Cluster)

Creates WLAN I/Q data and returns the data as a complex cluster. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set the **reset?** parameter to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** parameter is TRUE.

Note Use this instance if the Standard property is set to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS** or **80211G DSSSOFDM**.



niwlan generation session specifies the niWLAN generation session refnum.

reset? specifies whether to reset data. The default value is FALSE.

TRUE	Resets the data. Set thi s parameter to TRUE af

	ter you have generated the required number o f frames or if you want to reset the pseudonoi se (PN) seed.
FALSE	Does not reset the dat a.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

output waveform returns the WLAN I/Q data.

t0 returns the start time, in seconds.

dt returns the time interval, in seconds, between baseband I/Q samples.

Y returns the waveform data.

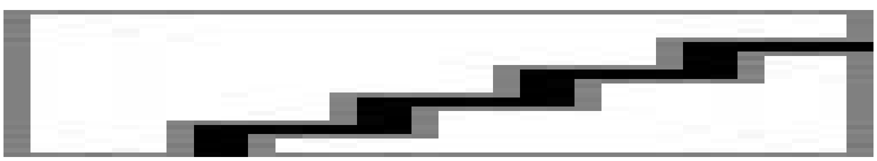
generation done? indicates whether the VI has generated all the data.

error out contains error information. This output provides standard error out functionality.

niWLANG Create Multiple Channel Waveform (IQ Complex Cluster)

Creates WLAN I/Q data for multiple channels and returns the data as an array of complex clusters. This VI returns one frame, including the idle interval, at a time. For multi-frame generation, set the **reset?** parameter to FALSE and run the VI in a loop for a specified number of times or until the **generation done?** parameter is TRUE.

Note Use this instance if the Standard property is set to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**.



niwlan generation session specifies the niWLAN generation session refnum.

reset? specifies whether to reset data. The default value is FALSE.

TRUE	Resets the data. Set this parameter to TRUE after you have generated the required number of frames or if you want

	to reset the pseudonoise (PN) seed.
FALSE	Does not reset the data.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

output waveform returns an array of WLAN I/Q data.

t0 returns the start time, in seconds.

dt returns the time interval, in seconds, between baseband I/Q samples.

Y returns the waveform data.

generation done? indicates whether the VI has generated all the data.

error out contains error information. This output provides standard error out functionality.

Generation Information

Owning Palette: Generation VIs

Use the VIs on this palette to get generator parameters used to generate a waveform.

Note The help topics for some VIs do not contain parameter details. Refer to the corresponding property help topics for more information about these VIs.

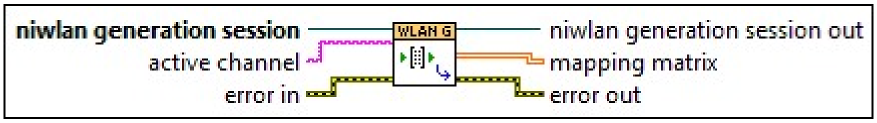
Palette Object	Description
niWLANG Get Mapping Matrix	Returns the matrix used for mapping space-time streams to the transmit channels as defined in section 20.3.11.10.1 of **IEEE Standard 802.11n-2009**.
niWLANG Get Actual Headroom	Returns the value of the Actual Headroom property.
niWLANG Get IQ Waveform Size	Returns the value of the IQ Waveform Size property.
niWLANG Get IQ Rate	Returns the value of the IQ Rate property.
niWLANG Get Number of Users from RU Allocation	Returns the number of users derived from the configured RU allocation in a 802.11ax MU PPDU signal. To use this VI, you must set the RU Allocation Mode property to **Group** and configure the RU Allocation property.
niWLANG Get OFDM Allocation Map Trace	Returns a map of allocation of users in space frequency grid, for the 802.11ax MU PPDU and Trigger-Based PPDU signals.

niWLANG Get Mapping Matrix (VI)

Owning Palette: Generation Information

Returns the matrix used for mapping space-time streams to the transmit channels as defined in section 20.3.11.10.1 of **IEEE Standard 802.11n-2009**.

Details



niwlan generation session specifies the niWLAN generation session refnum.

active channel specifies the active channel string.

You must use the following active channel string formats when you set the PPDU Type property to **Trigger-Based PPDU**.

Spatial Mapping Mode property value	Active channel string format
Common	"" (empty string)
User Specific	"user**x**"

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

mapping matrix returns the matrix for mapping spatial streams to transmit channels.

error out contains error information. This output provides standard error out functionality.

Details

The dimensions of the matrix must be **N_Tx** * (**N_STS** + **N_ESS**) if you set the Standard property to 80211N MIMOOFDM and **N_Tx** * **N_STS** if you set the Standard property to 80211AC MIMOOFDM, 80211AH MIMOOFDM, or 80211AX MIMOOFDM.

where	**N_Tx** is the number of transmit channels
	N_STS is the number of space-time streams. If you set the Standard property to 80211N MIMOOFDM, **N_STS** is determined by the MCS index and the STBC index
	N_ESS is the number of extension spatial streams

For one-to-one mapping (direct mapping), **N_Tx** = **N_SS** and **N_ESS** = **0**. For one-to-many mapping (spatial expansion), **N_Tx** >= (**N_SS** + **N_ESS**).

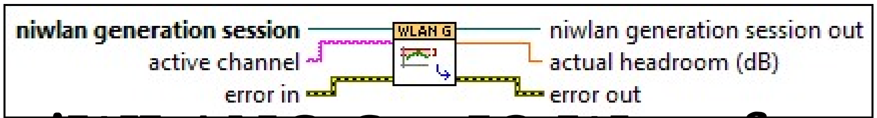
If you set the Standard property to 80211AX MIMOOFDM and the PPDU Type property to Trigger-Based PPDU, the **N STS** value is dependent on the Spatial Mapping Mode property.

Spatial Mapping Mode property value	N_STS value
Common	Maximum number of space time streams across Resource Units (RUs)
User Specific	Number of space time streams for the specified user

niWLANG Get Actual Headroom (VI)

Owning Palette: Generation Information

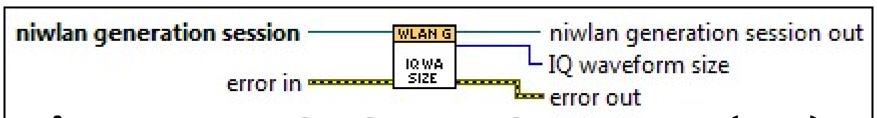
Returns the value of the Actual Headroom property.



niWLANG Get IQ Waveform Size (VI)

Owning Palette: Generation Information

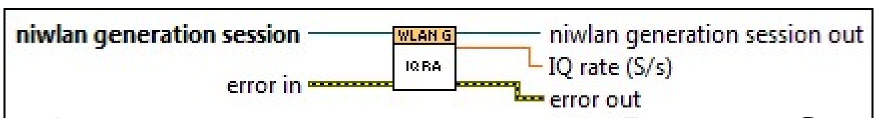
Returns the value of the IQ Waveform Size property.



niWLANG Get IQ Rate (VI)

Owning Palette: Generation Information

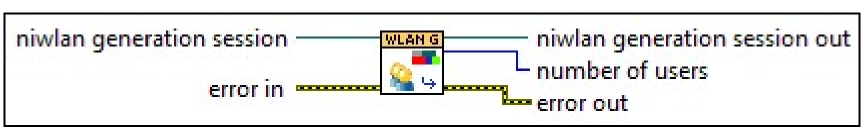
Returns the value of the IQ Rate property.



niWLANG Get Number of Users from RU Allocation (VI)

Owning Palette: Generation Information

Returns the number of users derived from the configured RU allocation in a 802.11ax MU PPDU signal. To use this VI, you must set the RU Allocation Mode property to **Group** and configure the RU Allocation property.



niwlan generation session specifies the niWLAN generation session refnum.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

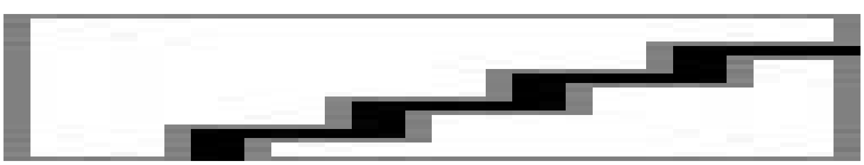
number of users returns the number of users in 802.11ax MU PPDU signal.

error out contains error information. This output provides standard error out functionality.

niWLANG Get OFDM Allocation Map Trace (VI)

Owning Palette: Generation Information

Returns a map of allocation of users in space frequency grid, for the 802.11ax MU PPDU and Trigger-Based PPDU signals.



niwlan generation session specifies the niWLAN generation session refnum.

intensity graph refnum specifies the reference to the intensity graph on which space frequency map will be displayed.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

space frequency map returns an array of the space frequency maps of the set configuration.

error out contains error information. This output provides standard error out functionality.

Utility

Owning Palette: Generation VIs

Use the VIs on this palette to reset all properties, obtain the error code returned by WLAN Generation VIs, save and load configuration files, and calculate carrier frequencies according to the IEEE 802.11 channel numbering scheme.

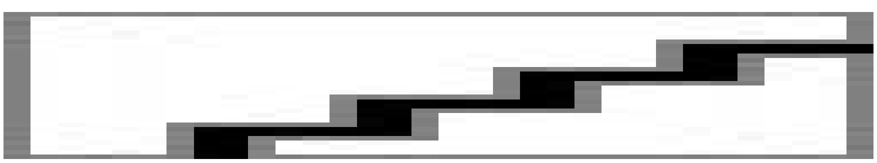
Palette Object	Description
niWLANG Load Configuration from File	Loads the properties of a session saved in a file.
niWLANG Save Configuration to File	Saves all properties of the session to a file located at a specified path. Use this file to save the current state of the toolkit.
niWLANG Channel Number to Carrier Frequency	Calculates the carrier frequency according to the numbering scheme by converting a set of input parameters, including the channel number, for **IEEE 802.11a/b/g/j/n/p**, **IEEE 802.11ac**, **IEEE 802.11ah**, **IEEE 802.11af** and **IEEE P802.11ax/D6.0** standards.
niWLANG Create Trigger Frame MSDU	Creates 802.11ax trigger frame MSDU bits according to the configuration you specify. You must set the Standard property to 80211AX MIMOOFDM and the PPDU Type property to Trigger-Based PPDU to configure this VI. For more information about Trigger frame generation, refer to the Generating the Trigger Frame topic.
niWLANG Create and Write Waveforms to File	Creates waveforms according to the properties configured in an niWLAN generation session and saves the waveforms to a file.
niWLANG Read Waveform from File	Reads a waveform from a TDMS file. You can save this file using the **WLAN Generation Soft Front Panel**. This VI returns headroom and I/Q rate waveform data that you can subsequently download to an NI RF vector signal generator.
niWLANG Read Waveform Names from File	Reads names of all waveforms present in a TDMS file. You can save this file using the **WLAN Generation Soft Front Panel** or the niWLANG Create and Write Waveforms to File VI in a programming environment.
niWLANG Read RF Blanking Marker Positions from File	Reads RF blanking marker positions from a TDMS file. You can save this file using the WLAN Generation Soft Front Panel or the niWLANG Create

	and Write Waveforms to File VI in a programmin g environment.
niWLANG Read Burst Start Locations from File	Reads the burst start locations from a TDMS file. You can save this file using the WLAN Generatio n Soft Front Panel or the niWLANG Create and Write Waveforms to File VI in a programming en vironment.
niWLANG Read Burst Stop Locations from File	Reads the burst stop locations from a TDMS file. You can save this file using the WLAN Generatio n Soft Front Panel or the niWLANG Create and Write Waveforms to File VI in a programming en vironment.
niWLANG RFSG Force TClk Synchronization	Specifies that the TClk synchronization has to b e performed as the previous TClk synchronizatio n is invalid due to niRFSG Reset VI. This setting will be used by niWLANG RFSG Synchronize TClk VI for TClk synchronization.

niWLANG Load Configuration from File (VI)

Owning Palette: Utility

Loads the properties of a session saved in a file.



	**niwlan generation session** specifies the niWLAN generation session refnum.
	**file path** specifies the complete path to the file from which the toolkit loads the configuration.
	**reset session?** specifies whether to reset all the properties of the session before loading the settings from a file. The default value is TRUE.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

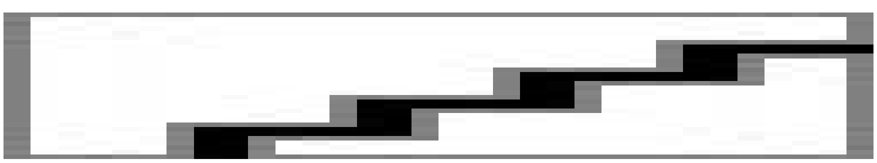
Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

niWLANG Save Configuration to File (VI)

Owning Palette: Utility

Saves all properties of the session to a file located at a specified path. Use this file to save the current state of the toolkit.



niwlan generation session specifies the niWLAN generation session refnum.

file path specifies the complete path to the TDMS file to which the toolkit saves the configuration.

operation specifies the operation to perform on the file. The default value is **create or replace**.

open (0)	Opens an existing file to write the niWLANG settings.
open or create (1)	Opens an existing file or creates a new file if the file does not exist.
create or replace (2)	Creates a new file or replaces an existing file.
create (3)	Creates a new file.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

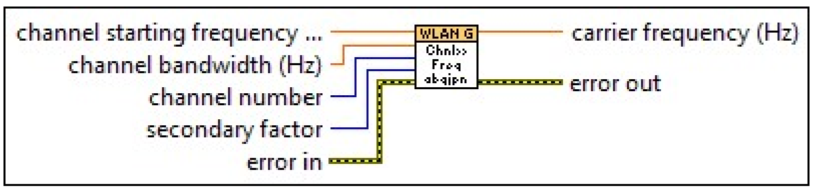
niWLANG Channel Number to Carrier Frequency (VI)

Owning Palette: Utility

Calculates the carrier frequency according to the numbering scheme by converting a set of input parameters, including the channel number, for **IEEE 802.11a/b/g/j/n/p**, **IEEE 802.11ac**, **IEEE 802.11ah**, **IEEE 802.11af** and **IEEE P802.11ax/D6.0** standards.

niWLANG Channel Number to Carrier Frequency (802.11abgjpn)

Calculates carrier frequency of 802.11a/b/g/j/p/n channels according to sections 16.4.6, 17.4.6, 18.3.8.4, and 20.3.15 of **IEEE Standard 802.11-2012**.



channel starting frequency specifies the start frequency, in Hz, of the frequency band.

channel bandwidth specifies the channel bandwidth, in Hz, used for transmitting the signal. You can choose a 5 MHz, 10 MHz, 20 MHz, or 40 MHz channel.

channel number specifies the offset of the center frequency, in increments of 5 MHz, above the start frequency of the channel.

secondary factor specifies whether the secondary channel is above or below the primary channel when you set the **channel bandwidth** parameter to 40 MHz.

A 40 MHz channel is created by combining the primary channel and the secondary channel, where each of the channels have a bandwidth of 20 MHz. The value of -1 indicates that the secondary channel is below the primary channel whereas the value of +1 indicates that the secondary channel is above the primary channel.

Valid values are -1 and +1.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

carrier frequency returns the carrier frequency. This value is expressed in Hz.

The VI calculates the carrier frequency using the following equation: **Carrier Frequency** (Hz) = **Channel Starting Frequency** (Hz) + (**Channel Number** * 5 MHz).

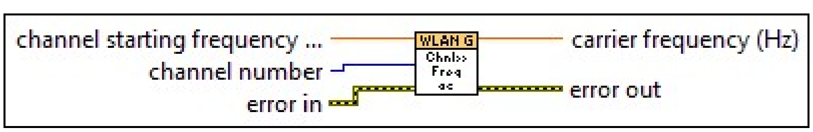
When you set the **channel bandwidth** parameter to 40 MHz, the **channel number** parameter is the primary channel number. The VI calculates the carrier frequency using the following formula:

Carrier frequency (Hz) = **Channel Starting Frequency** (Hz) + (**Channel Number** * 5 MHz) + (**Secondary Factor** * 20 MHz)

error out contains error information. This output provides standard error out functionality.

niWLANG Channel Number to Carrier Frequency (802.11ac)

Calculates carrier frequency of 802.11ac channels according to section 22.3.14 of **IEEE Standard 802.11ac-2013**.



channel starting frequency specifies the start frequency, in Hz, of the frequency band.

channel number specifies the offset of the center frequency, in increments of 5 MHz, above the start frequency of the channel. The default value is 1.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

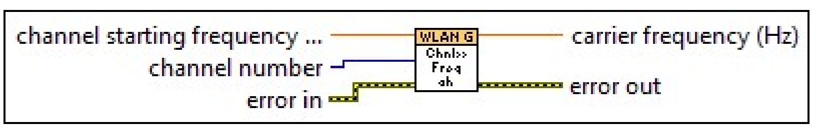
carrier frequency returns the carrier frequency. This value is expressed in Hz.

The VI calculates the carrier frequency using the following equation: **Carrier Frequency** (Hz) = **Channel Starting Frequency** (Hz) + (**Channel Number** * 5 MHz).

error out contains error information. This output provides standard error out functionality.

niWLANG Channel Number to Carrier Frequency (802.11ah)

Calculates the carrier frequency of 802.11ah according to section 22.3.13 of **IEEE P802.11ah/D1.3**.



channel starting frequency specifies the start frequency, in Hz, of the frequency band.

channel number specifies the offset of the center frequency, in increments of 0.5 MHz, above the start frequency of the channel. The default value is 1.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

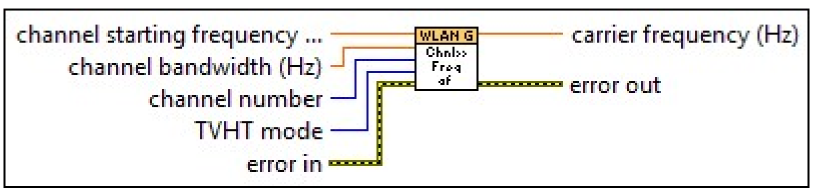
carrier frequency returns the carrier frequency. This value is expressed in Hz.

The VI calculates the carrier frequency using the following equation: **Carrier Frequency** (Hz) = **Channel Starting Frequency** (Hz) + (**Channel Number** * 0.5 MHz).

error out contains error information. This output provides standard error out functionality.

niWLANG Channel Number to Carrier Frequency (802.11af)

Calculates the carrier frequency of 802.11af channels, as defined in section 23.3.14 of **IEEE Standard 802.11af-2013**.



channel starting frequency specifies the start frequency, in Hz, of the frequency band. The channel start frequency is given by the following formula:

Channel Starting Frequency (Hz) = **Channel Starting Factor** * 500 kHz

channel bandwidth specifies the channel bandwidth, in Hz, used for transmitting the signal. The default value is 6 MHz.

channel number specifies the offset of the center frequency, in increments of the channel bandwidth, used for transmitting the signal.

This value is expressed in Hz. The default value is 1 MHz.

TVHT mode specifies the mode of the 802.11af signal transmission.

The default value is TVHT_MODE_4C.

TVHT_MODE_1 (0)	Specifies the TVHT mode representing a single basic channel unit (BCU).
TVHT_MODE_2C (1)	Specifies the TVHT mode representing two contiguous BCUs.
TVHT_MODE_2N (2)	Specifies the TVHT mode representing two noncontiguous BCUs.
TVHT_MODE_4C (3)	Specifies the TVHT mode representing four contiguous BCUs.
TVHT_MODE_4N (4)	Specifies the TVHT mode representing two noncontiguous frequency segments, each of which is composed of two BCUs.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

carrier frequency returns the carrier frequency. This value is expressed in Hz.

The VI calculates the carrier frequency using the following equation:

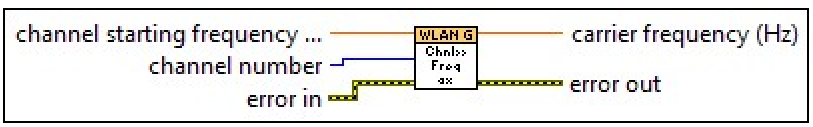
carrier frequency (Hz) = **channel starting frequency (Hz)** + (**TVHT_W** * **channel number**) + **channel center frequency correction (Hz)**

where **channel center frequency correction** is used to adjust the carrier frequency in the different TVHT modes. It is 0 for TVHT_MODE_1 and TVHT_MODE_2N, 0.5 * **TVHT_W** for TVHT_MODE_2C and TVHT_MODE_4N, and 1.5 * **TVHT_W** for TVHT_MODE_4C.

error out contains error information. This output provides standard error out functionality.

niWLANG Channel Number to Carrier Frequency (802.11ax)

Calculates the carrier frequency of the 802.11ax channels.



channel starting frequency specifies the start frequency of the frequency band. This value is expressed in Hz.

channel number specifies the offset of the center frequency, in increments of 5 MHz, above the start frequency of the channel. The default value is 1.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

carrier frequency returns the carrier frequency. This value is expressed in Hz.

The VI calculates the carrier frequency using the following equation: **Carrier Frequency** (Hz) =

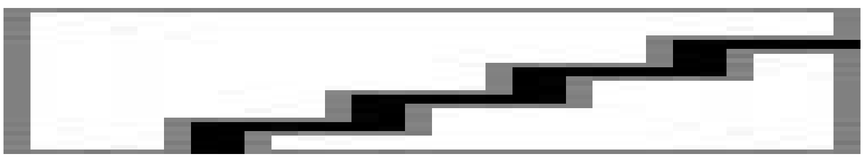
Channel Starting Frequency (Hz) + (**Channel Number** * 5 MHz).

error out contains error information. This output provides standard error out functionality.

niWLANG Create Trigger Frame MSDU (VI)

Owning Palette: Utility

Creates 802.11ax trigger frame MSDU bits according to the configuration you specify. You must set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **Trigger-Based PPDU** to configure this VI. For more information about Trigger frame generation, refer to the Generating the Trigger Frame topic.



niwlan generation session specifies the niWLAN generation session refnum.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid

possible memory leak issues.

trigger frame MSDU bits returns an array of the encoded bits trace.

generation done? indicates whether the VI has generated all the data.

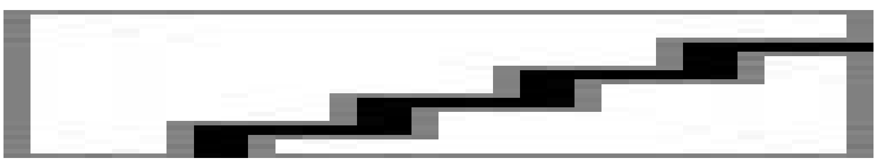
error out contains error information. This output provides standard error out functionality.

niWLANG Create and Write Waveforms to File (VI)

Owning Palette: Utility

Creates waveforms according to the properties configured in an niWLAN generation session and saves the waveforms to a file.

Details



niwlan generation session specifies the niWLAN generation session refnum.

file path specifies the absolute path to the TDMS file to which the toolkit writes the waveforms.

operation specifies the operation to perform on the file. The default value is **create or replace**.

open (0)	Opens an existing file to write the niWLANG settings.

open or create (1)	Opens an existing file or creates a new file if the file does not exist.
create or replace (2)	Creates a new file or replaces an existing file.
create (3)	Creates a new file.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

Details

In addition to creating the waveform, this VI also saves the Headroom and IQ Rate properties for each waveform. The VI reads the RF Blanking Marker Positions property and saves the value to the file. The RF Blanking Marker Positions property stored in the file is applicable to all waveforms stored in the file. This VI also saves Signal Bandwidth property for each waveform, which is computed as follows:

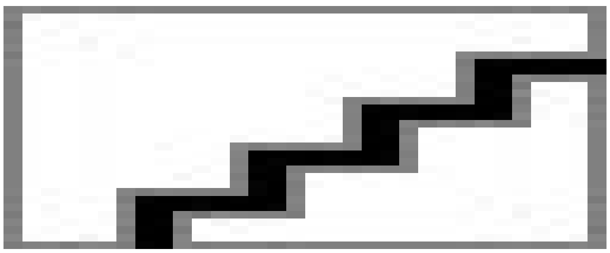
Signal Bandwidth = 2 * {(Channel Bandwidth/2) + |Max Carrier Frequency Offset|}

niWLANG Read Waveform from File (VI)

Owning Palette: Utility

Reads a waveform from a TDMS file. You can save this file using the **WLAN Generation Soft Front Panel**. This VI returns headroom and I/Q rate waveform data that you can subsequently download to an NI RF vector signal generator.

Details



file path specifies the absolute path to the TDMS file from which the toolkit reads the waveform.

waveform name specifies the name of the waveform to read from the file. For example, use the **channel0** string as a waveform name to read the waveform for channel 0.

count specifies the maximum number of samples of the I/Q complex waveform to read from the file.

The default value is -1, which returns all samples. If you set **count** to 1,000 and **offset** to 2, the VI returns 1,000 samples, starting from index 2 and ending at index 1,002.

offset specifies the number of waveform samples at which the VI begins reading the I/Q data.

The default value is 0. If you set **count** to 1,000 and **offset** to 2, the VI returns 1,000 samples, starting from index 2 and ending at index 1,002.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

waveform returns the waveform saved in a TDMS file.

t0 returns the start time, in seconds, of the **Y** array.

dt returns the time interval, in seconds, between the samples in the **Y** array. The reciprocal of **dt** indicates the I/Q rate of the signal.

Y returns an array of complex-valued time domain data. The real and imaginary parts of this complex data array correspond to the in-phase (I) and quadrature-phase (Q) data, respectively.

IQ rate returns the I/Q rate, in samples per second (S/s), of the waveform.

headroom returns the headroom, in dB, of the waveform.

eof indicates whether the end of file has been reached with this read.

error out contains error information. This output provides standard error out functionality.

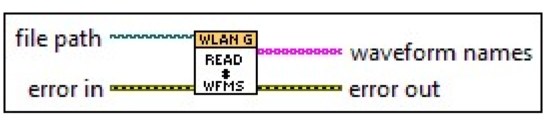
Details

Standard Property Value	Active Channel String Syntax	Comments
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, or **80211G DSSSOFDM**	""	"" for all waveforms
80211N MIMOOFDM, **80211AH MIMOOFDM**	"channel**x**"	"channel1" for a waveform of channel with index 1
80211AC MIMOOFDM, **80211AF MIMOOFDM**, **80211AX MIMOOFDM**	"[segment**z**/]channel**x**"	"segment1/channel0" for a waveform of channel with index 0 of segment with index 1. "segment0/" is optional if the segment index is 0.

niWLANG Read Waveform Names from File (VI)

Owning Palette: Utility

Reads names of all waveforms present in a TDMS file. You can save this file using the **WLAN Generation Soft Front Panel** or the niWLANG Create and Write Waveforms to File VI in a programming environment.



file path specifies the absolute path to the TDMS file from which the toolkit reads the waveform.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

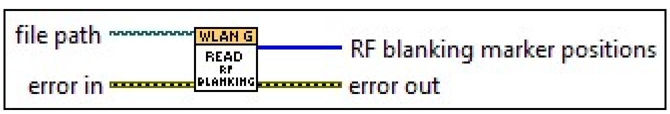
waveform names returns an array of names of all the waveforms present in the TDMS file.

error out contains error information. This output provides standard error out functionality.

niWLANG Read RF Blanking Marker Positions from File (VI)

Owning Palette: Utility

Reads RF blanking marker positions from a TDMS file. You can save this file using the WLAN Generation Soft Front Panel or the niWLANG Create and Write Waveforms to File VI in a programming environment.



file path specifies the absolute path to the TDMS file from which the toolkit reads the waveform.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

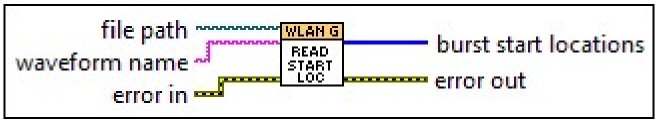
RF blanking marker positions returns the RF blanking marker positions saved in the TDMS file. It is an array of sample positions, within waveform, of marker events that can be used to toggle the state of RF blanking.

error out contains error information. This output provides standard error out functionality.

niWLANG Read Burst Start Locations from File (VI)

Owning Palette: Utility

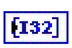
Reads the burst start locations from a TDMS file. You can save this file using the WLAN Generation Soft Front Panel or the niWLANG Create and Write Waveforms to File VI in a programming environment.



file path specifies the absolute path to the TDMS file from which the toolkit reads the waveform.

waveform name specifies the name of the waveform to read from the file. For example, use the **channel0** string as a waveform name to read the waveform for channel 0.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.



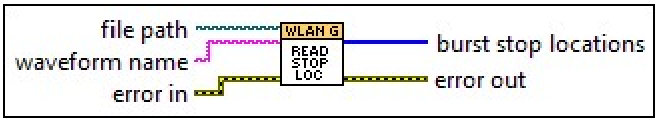
burst start locations returns the burst start locations saved in the TDMS file. It is an array of sample positions of the start of the burst, within the waveform.

error out contains error information. This output provides standard error out functionality.

niWLANG Read Burst Stop Locations from File (VI)

Owning Palette: Utility

Reads the burst stop locations from a TDMS file. You can save this file using the WLAN Generation Soft Front Panel or the niWLANG Create and Write Waveforms to File VI in a programming environment.



file path specifies the absolute path to the TDMS file from which the toolkit reads the waveform.

waveform name specifies the name of the waveform to read from the file. For example, use the **channel0** string as a waveform name to read the waveform for channel 0.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

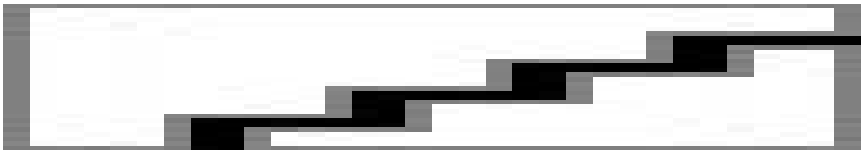
burst stop locations returns the burst stop locations saved in the TDMS file. It is an array of sample positions of the end of the burst, within the waveform.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Force TClk Synchronization (VI)

Owning Palette: Utility

Specifies that the TClk synchronization has to be performed as the previous TClk synchronization is invalid due to niRFSG Reset VI. This setting will be used by niWLANG RFSG Synchronize TClk VI for TClk synchronization.



niwlan generation session specifies the niWLAN generation session refnum.

instrument handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

force sync? specifies that a fresh TClk synchronization has to be performed for TClk synchronized generation.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

instrument handles out passes an array of RFSG instrument sessions to the next VI.

error out contains error information. This output provides standard error out functionality.

RFSG Database

Owning Palette: Generation VIs

Use the VIs on this palette to configure, store, or retrieve information from the RFSG database.

Palette Object	Description
niWLANG RFSG Create and Download Waveform	Creates the waveform according to the paramet ers that you specify, writes it into the NI RF vect or signal generator memory, and stores the I/Q r ate and actual headroom of the waveform in the RFSG database.
niWLANG RFSG Read and Download Waveforms from File	Reads the waveforms stored in a TDMS file, writ es them to the memory of the respective RF vect or signal generato, and stores the I/Q rate, actua l headroom, burst start locations, burst stop loc ations, waveform size, and RF blanking marker positions of the waveforms in the RFSG Database.
niWLANG RFSG Configure Script	Configures the I/Q rate and power level of the w aveforms that you specify in the **script** paramet er. If the I/Q rates are the same for all the wavefo rms, this VI sets the niRFSG IQ Rate property to t he I/Q rate in the RFSG database. The VI sets the niRFSG Power Level property to the sum of the power level that you specify in the **power level** parameter and the minimum headroom value of all the waveforms.
niWLANG RFSG Clear Database	Clears the properties stored in the RFSG database and clears the waveforms from the RF vector signal generator memory.
niWLANG RFSG Configure Power Level	Looks up the waveforms in the script, retrieves t he minimum actual headrooms of the wavefor ms in the script, adds this value to the **power le vel** parameter, and sets the result to the niRFSG Power Level property. Set the niRFSG Power Lev

	el Type property to **Peak Power** before calling this VI.
niWLANG RFSG Insert RF Blanking Marker Positions	Configures the script you specify in the **script** parameter for RF blanking marker events.

Subpalette	Description
Store	Use the VIs on this palette to store parameters for the NI RF vector signal generator.
Retrieve	Use the VIs on this palette to retrieve parameters for the NI RF vector signal generator.

niWLANG RFSG Create and Download Waveform (VI)

Owning Palette: RFSG Database

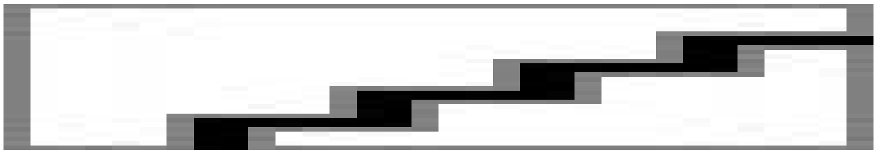
Creates the waveform according to the parameters that you specify, writes it into the NI RF vector signal generator memory, and stores the I/Q rate and actual headroom of the waveform in the RFSG database.

niWLANG RFSG Create and Download Waveform (Single Channel)

Creates a single channel waveform, writes it into the RFSG memory, and stores the I/Q rate, burst start locations, burst stop locations, waveform size, and the actual headroom of the waveform in the RFSG database.

Note Use this instance if the Standard property is set to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, or **80211G DSSSOFDM**.

Instance Details



niwlan generation session specifies the niWLAN generation session refnum.

instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name used to write the waveform to NI-RFSG device memory and store its properties to RFSG database. This string is case-insensitive, alphanumeric, and does not use reserved words.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

> **Note** Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Create and Download Waveform (Single Channel) Details

Inside the niWLANG Create and Download Waveform (Single Channel) VI, all the following operations that involve NI-RFSG are performed on the device. The niWLANG RFSG Create and Download Waveforms (Single Channel) VI completes the following operations:

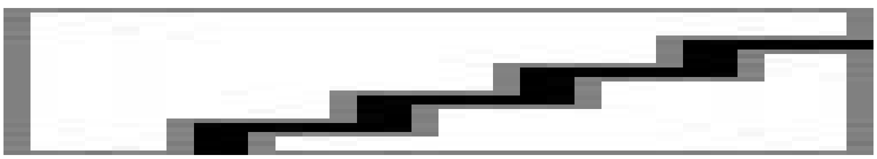
1. Sets the Maximum Hardware IQ Rate property according to the device model. You must set the value of the Maximum Hardware IQ Rate property to 1250 MS/s if you are using the PXIe-5840/5841/5841 with 5655/5820/5830/5831, 250 MS/s if you are using the PXIe-5646R, 120 Ms/s if you are using the PXIe-5644R/5645R, and 200MS/s if you are using the PXIe-5673/5673E.
2. Reads the IQ Rate property and sets the niRFSG IQ Rate property to the value specified in the IQ Rate property. The value is stored in the RFSG database for the waveform and device.
3. If the RF Blanking Enabled property is set to **True** and the device model PXIe-5644R, PXIe-5645R, PXIe-5646R, PXIe-5840, PXIe-5841, or PXIe-5841 with PXIe-5655 is used, sets the niRFSG RF Blanking Source property to "marker0", if it is not set already.
4. Reads and stores the Signal Bandwidth property in the RFSG database for PXIe-5820/5830/5831/5841/5841 with 5655.
5. Reads and stores the Burst Start Locations property, the Burst Stop Locations property, and the waveform size in the RFSG database.
6. Creates the WLAN waveform and downloads it to device.
7. Stores the Actual Headroom property to the RFSG database.

niWLANG RFSG Create and Download Waveforms (Multiple Channel)

The VI creates multiple channel waveforms, writes each waveform into the respective RFSG memory, and stores the I/Q rate, burst start locations, burst stop locations, waveform size, and the actual headroom for each channel of the waveform in the respective RFSG database, if you set the Standard property to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**.

The VI creates a single channel waveform and writes the same waveform to RFSG memory of all devices, and stores the I/Q rate, burst start locations, burst stop locations, waveform size and actual headroom of the waveform in all RFSG databases, if you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, or **80211G DSSSOFDM**.

Instance Details



niwlan generation session specifies the niWLAN generation session refnum.

instrument handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name used to write the waveform to NI-RFSG device memory and store its properties to RFSG database. This string is case-insensitive, alphanumeric, and does not use reserved words.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

Note Close the niWLAN generation

session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

instrument handles out passes an array of RFSG instrument sessions to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Create and Download Waveforms (Multiple Channel) Details

Inside the niWLANG Create and Download Waveform (Multiple Channel) VI, all the operations listed below involving NI-RFSG are performed on the device corresponding to each channel. The niWLANG RFSG Create and Download Waveforms (Multiple Channel) VI completes the following operations.

1. Sets the Maximum Hardware IQ Rate property according to the device model. You must set the value of the Maximum Hardware IQ Rate property to 1250 MS/s if you are using the PXIe-5840/5841/5820/5830/5831, 250 MS/s if you are using the PXIe-5646, 120 Ms/s if you are using the PXIe-5644/5645, and 200MS/s if you are using the PXIe-5673/5673E.
2. Reads the IQ Rate property and sets the niRFSG IQ Rate property to the value specified in the IQ Rate property. The value is stored in the RFSG database for the waveform and device.
3. Sets the niRFSG RF Blanking Source property to "marker0" (if it is not set already), if the RF Blanking Enabled property is set to **True** and the device model PXIe-5644, PXIe-5645, PXIe-5646, PXIe-5840, or PXIe-5841 is used.
4. Reads and stores the Signal Bandwidth property in the RFSG database for PXIe-5820/5830/5831/5841.

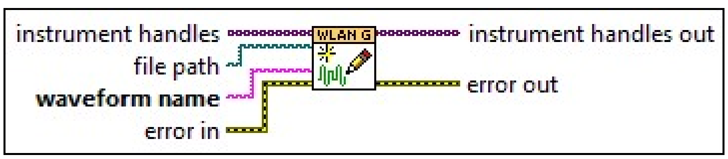
5. Reads and stores the Burst Start Locations property, the Burst Stop Locations property, and the waveform size in the RFSG database.
6. Creates the WLAN waveform and downloads it to device.
7. Stores the Actual Headroom property to the RFSG database.

niWLANG RFSG Read and Download Waveforms from File (VI)

Owning Palette: RFSG Database

Reads the waveforms stored in a TDMS file, writes them to the memory of the respective RF vector signal generato, and stores the I/Q rate, actual headroom, burst start locations, burst stop locations, waveform size, and RF blanking marker positions of the waveforms in the RFSG Database.

Details



instrument handles identifies instruments for multiple NI-RFSG sessions. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

file path specifies the absolute path to the TDMS file from which the toolkit reads the waveforms.

waveform name specifies the names of the waveforms to clear. If you set this parameter as empty, the VI clears all the waveforms and their properties.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handles out passes an array of RFSG instrument sessions to the next VI.

error out contains error information. This output provides standard error out functionality.

Details

This VI completes the following tasks:

- Calls the niWLANG Read Waveform Names from File VI to read the names of all waveforms present in the TDMS file.
- Calls the niWLANG Read RF Blanking Marker Positions from File VI to read RF blanking marker positions.
- Completes the following tasks for each waveform in the file and corresponding NI-RFSG device (instrument handle).
  - Calls the niWLANG Read Waveform from File VI to read the waveform, along with its I/Q rate and headroom. The niRFSG I/Q Rate (S/s) property is set to the value of I/Q rate read from the file. The I/Q rate and headroom are stored in the RFSG database for the waveform specified in the waveform name parameter and NI-RFSG device.
  - For PXIe-5820/5830/5831/5841/5841 with 5655 devices, reads the signal bandwidth from the file and stores in the RFSG database for the waveform specified in the **waveform name** parameter and NI-RFSG device. If the value is not found in the file, then stores the value equal to 0.8*I/Q rate.
  - If the RF blanking marker positions, or burst start locations and burst stop locations read from the file are not empty arrays and are any of PXIe-5644, PXIe-5645, PXIe-5646, PXIe-5840, PXIe-5841, or PXIe-5841 with PXIe-5655 devices, the niRFSG RF Blanking Source property is set to "marker0", if it is not set. The RF blanking marker positions, burst start locations, burst stop locations, and the waveform size are stored in the RFSG database for the waveform specified in the waveform name parameter and the NI-RFSG device.
  - The VI downloads the waveform read from the file to the NI-RFSG device.

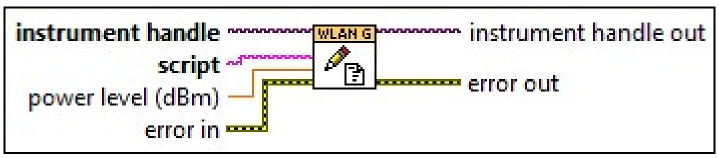
Note If there is a single waveform in the file, these operations are repeated for each NI-RFSG device with the same input waveform.

niWLANG RFSG Configure Script (VI)

Owning Palette: RFSG Database

Configures the I/Q rate and power level of the waveforms that you specify in the **script** parameter. If the I/Q rates are the same for all the waveforms, this VI sets the niRFSG IQ Rate property to the I/Q rate in the RFSG database. The VI sets the niRFSG Power Level property to the sum of the power level that you specify in the **power level** parameter and the minimum headroom value of all the waveforms.

Details



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

script specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name using the NI-RFSG Selected Script property.

power level specifies the average power level, in dBm. The default value is -10 dBm.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

Details

Note Call the niWLANG RFSG Create and Download Waveform VI before calling the niWLANG RFSG Configure Script VI.

This VI completes the following tasks:

1. For devices other than PXIe 5820, calls the niWLANG RFSG Retrieve Minimum PAPR (Script) VI to retrieve the minimum peak-to-average power ratio (PAPR) across all waveforms present in the **script** parameter. The niRFSG Power Level property is set to the sum of the **power level** parameter and the minimum PAPR.
2. For PXIe 5820/5830/5831/5841/5841 with 5655, calls niWLANG RFSG Retrieve Signal Bandwidth (Script) VI to retrieve the Signal Bandwidth and set niRFSG Signal Bandwidth to that value.
3. Calls the niWLANG RFSG Retrieve IQ Rate (Script) VI to retrieve the I/Q rate and sets niRFSG IQ Rate to the value.
4. Sets the niRFSG Power Level Type property to **Peak Power**.
5. Calls the niWLANG RFSG Insert RF Blanking Marker Positions VI to configure the script with RF blanking marker events.
6. Writes the resultant script to the NI-RFSG device using the niRFSG Write Script VI.
7. Parses the script name from the script parameter and sets the niRFSG Selected Script property to the value.

Store

Owning Palette: RFSG Database

Use the VIs on this palette to store parameters for the NI RF vector signal generator.

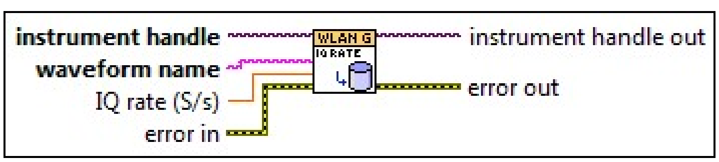
Palette Object	Description

niWLANG RFSG Store IQ Rate	Stores the I/Q rate that you specify in the **IQ rate** parameter, in the RFSG Database.
niWLANG RFSG Store PAPR	Stores the headroom, or peak-to-average power ratio (PAPR), which you specify in the **PAPR** parameter, in the RFSG database.
niWLANG RFSG Store RF Blanking Marker Positions	Stores the RF blanking marker positions that you specify in the **RF Blanking Marker Positions** parameter in the RFSG database.
niWLANG RFSG Store Waveform Size	Stores the waveform size that you specify in the **waveform size** parameter in the RFSG Database.
niWLANG RFSG Store Burst Start Locations	Stores the burst start locations that you specify in the **burst start locations** parameter in the RFSG Database.
niWLANG RFSG Store Burst Stop Locations	Stores the burst stop locations that you specify in the **burst stop locations** parameter in the RFSG Database.

niWLANG RFSG Store IQ Rate (VI)

Owning Palette: Store

Stores the I/Q rate that you specify in the **IQ rate** parameter, in the RFSG Database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to store the I/Q rate.

IQ rate specifies the I/Q rate, in samples per second (S/s), to store in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Store PAPR (VI)

Owning Palette: Store

Stores the headroom, or peak-to-average power ratio (PAPR), which you specify in the **PAPR** parameter, in the RFSG database.

instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to store the maximum expected peak-to-average power ratio (PAPR).

PAPR specifies the headroom (or PAPR), in dB, to store in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

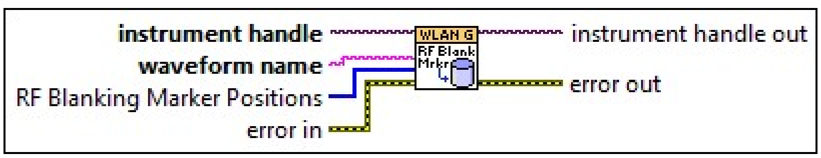
instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Store RF Blanking Marker Positions (VI)

Owning Palette: Store

Stores the RF blanking marker positions that you specify in the **RF Blanking Marker Positions** parameter in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the waveform for which you want to store the RF blanking marker positions.

RF Blanking Marker Positions specifies the RF blanking marker positions to store. It is an array of sample positions, within the waveform, of marker events that can be used to toggle the state of RF blanking.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

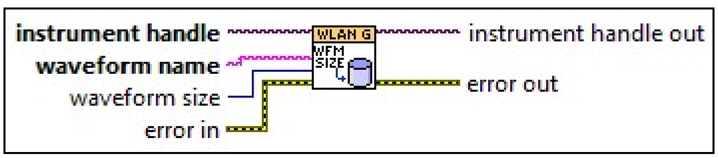
instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Store Waveform Size (VI)

Owning Palette: Store

Stores the waveform size that you specify in the **waveform size** parameter in the RFSG Database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to store the waveform size.

waveform size specifies the waveform size to store. This value is expressed in samples. The default value is 1.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

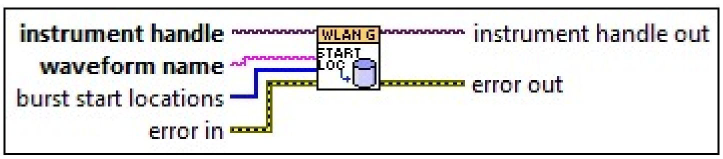
instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Store Burst Start Locations (VI)

Owning Palette: Store

Stores the burst start locations that you specify in the **burst start locations** parameter in the RFSG Database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to store the burst start locations.

burst start locations specifies the burst start locations to store. It is an array of sample positions of the start of the burst, within the waveform. The default value is "" (empty array).

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

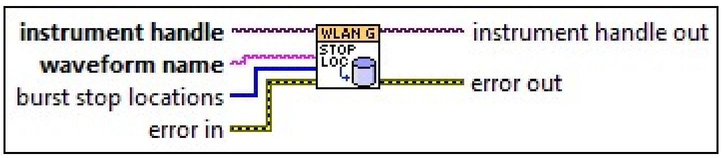
instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Store Burst Stop Locations (VI)

Owning Palette: Store

Stores the burst stop locations that you specify in the **burst stop locations** parameter in the RFSG Database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to store the burst stop locations.

burst stop locations specifies the burst stop locations to store. It is an array of sample positions of the end of the burst, within the waveform. The default value is "" (empty array).

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

Retrieve

Owning Palette: RFSG Database

Use the VIs on this palette to retrieve parameters for the NI RF vector signal generator.

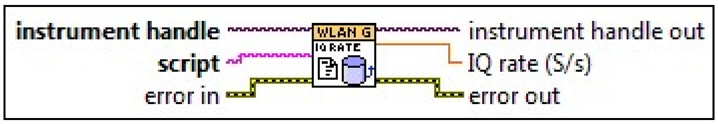
Palette Object	Description
niWLANG RFSG Retrieve IQ Rate (Script)	Checks the I/Q rate of all the waveforms in the script that you specify in the **script** parameter. This VI returns the I/Q rate if the I/Q rates are the same for all the waveforms. If the I/Q rates are different, the VI returns an error.
niWLANG RFSG Retrieve Minimum PAPR (Script)	Returns the minimum value of the headroom, or PAPR, of all the waveforms in the script that you specify in the **script** parameter.
niWLANG RFSG Retrieve IQ Rate (Waveform)	Returns the I/Q rate stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform properties.
niWLANG RFSG Retrieve PAPR (Waveform)	Returns the headroom, or PAPR, stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform PAPR.
niWLANG RFSG Retrieve RF Blanking Marker Positions (Waveform)	Returns the RF blanking marker positions stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform properties.
niWLANG RFSG Retrieve Burst Start Locations	Returns the burst start locations stored in the RFSG database.
niWLANG RFSG Retrieve Burst Stop Locations	Returns the burst stop locations stored in the RFSG database.

niWLANG RFSG Retrieve Waveform Size	Returns the waveform size stored in the RFSG database.

niWLANG RFSG Retrieve IQ Rate (Script) (VI)

Owning Palette: Retrieve

Checks the I/Q rate of all the waveforms in the script that you specify in the **script** parameter. This VI returns the I/Q rate if the I/Q rates are the same for all the waveforms. If the I/Q rates are different, the VI returns an error.



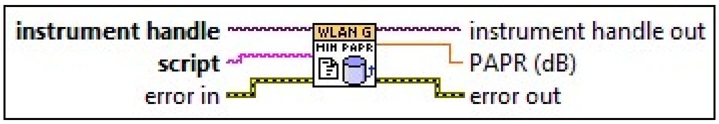
	**instrument handle** identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.
	**script** specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name using the NI-RFSG Selected Script property.
	**error in** describes error conditions that occur before this node runs. This input provides standard error in functionality.
	**instrument handle out** passes a reference from the instrument session to the next VI.
	**IQ rate** returns the I/Q rate, in samples per second (S/s), if the I/Q rates are the same for all the waveforms that you specify in the **script** parameter. If the I/Q rates are different, the VI returns an error.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve Minimum PAPR (Script) (VI)

Owning Palette: Retrieve

Returns the minimum value of the headroom, or PAPR, of all the waveforms in the script that you specify in the **script** parameter.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

script specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name using the NI-RFSG Selected Script property.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

PAPR returns the minimum of all the maximum expected peak-to-average power ratio (PAPR) values, in dB, stored in the RFSG database.

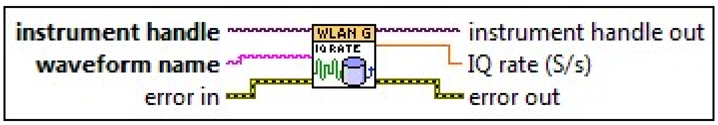
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve IQ Rate (Waveform) (VI)

Owning Palette: Retrieve

Returns the I/Q rate stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform properties.

Note Use the niWLANG RFSG Store IQ Rate VI to store the I/Q rate in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to retrieve the I/Q rate. The toolkit uses the **waveform name** parameter as the key to retrieve the waveform properties in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

IQ rate returns the I/Q rate, in samples per second (S/s), stored in the RFSG database for the waveform you specified in the **waveform name** parameter.

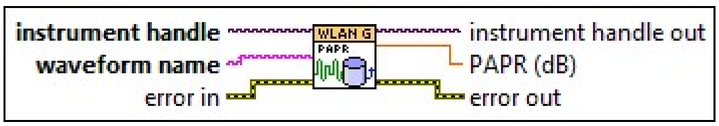
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve PAPR (Waveform) (VI)

Owning Palette: Retrieve

Returns the headroom, or PAPR, stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform PAPR.

Note Use the niWLANG RFSG Store PAPR VI to store the PAPR in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to retrieve the PAPR. The toolkit uses the **waveform name** parameter as the key to retrieve the waveform properties in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

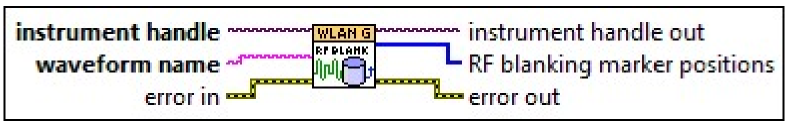
PAPR returns the peak-to-average power ratio (PAPR), in dB, stored in the RFSG database, for the waveform you specified in the **waveform name** parameter.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve RF Blanking Marker Positions (Waveform) (VI)

Owning Palette: Retrieve

Returns the RF blanking marker positions stored in the RFSG database. The VI uses the waveform name as the key to retrieve the waveform properties.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the names of the waveforms to clear. If you set this parameter as empty, the VI clears all the waveforms and their properties.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

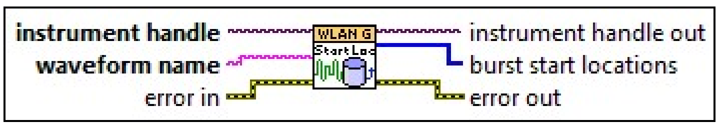
RF blanking marker positions returns the RF blanking marker positions saved in the TDMS file. It is an array of sample positions, within waveform, of marker events that can be used to toggle the state of RF blanking.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve Burst Start Locations (VI)

Owning Palette: Retrieve

Returns the burst start locations stored in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to retrieve the burst start locations. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

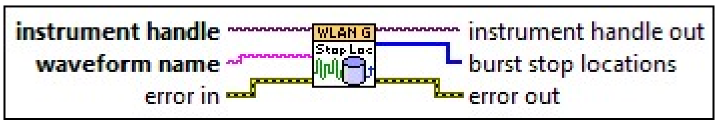
burst start locations returns the burst start locations stored, for the waveform you specified in the **waveform name** parameter. It is an array of sample positions of the start of the burst, within the waveform.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve Burst Stop Locations (VI)

Owning Palette: Retrieve

Returns the burst stop locations stored in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to retrieve the burst stop locations. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

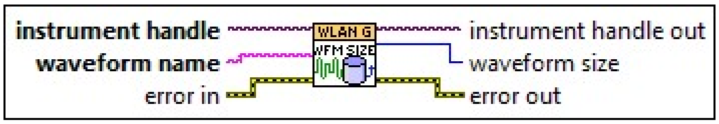
burst stop locations returns the burst stop locations stored, for the waveform you specified in the **waveform name** parameter. It is an array of sample positions of the end of the burst, within the waveform.

error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Retrieve Waveform Size (VI)

Owning Palette: Retrieve

Returns the waveform size stored in the RFSG database.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the name of the waveform for which you want to retrieve the waveform size. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

waveform size returns the waveform size stored, for the waveform you specified in the **waveform name** parameter.

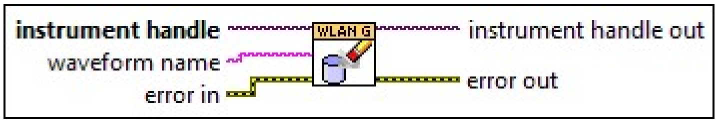
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Clear Database (VI)

Owning Palette: RFSG Database

Clears the properties stored in the RFSG database and clears the waveforms from the RF vector signal generator memory.

Details



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

waveform name specifies the names of the waveforms to clear. If you set this parameter as empty, the VI clears all the waveforms and their properties.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

error out contains error information. This output provides standard error out functionality.

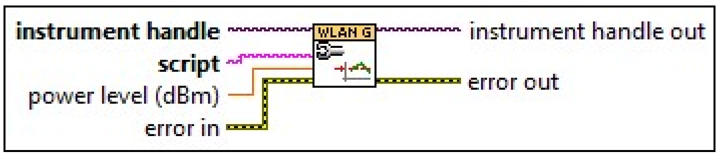
Details

This VI clears the waveforms and the properties of the waveforms you specified in the **waveform name** parameter. If you do not set the **waveform name** parameter or set it to "" (empty string), this VI clears all the waveforms and their properties.

niWLANG RFSG Configure Power Level (VI)

Owning Palette: RFSG Database

Looks up the waveforms in the script, retrieves the minimum actual headrooms of the waveforms in the script, adds this value to the **power level** parameter, and sets the result to the niRFSG Power Level property. Set the niRFSG Power Level Type property to **Peak Power** before calling this VI.



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

script specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name using the NI-RFSG Selected Script property.

power level specifies the average power level, in dBm.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

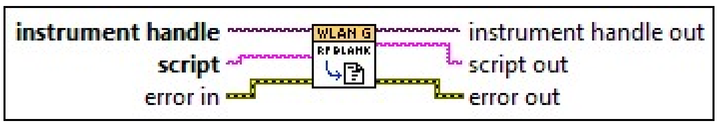
error out contains error information. This output provides standard error out functionality.

niWLANG RFSG Insert RF Blanking Marker Positions (VI)

Owning Palette: RFSG Database

Configures the script you specify in the **script** parameter for RF blanking marker events.

Details



instrument handle identifies the instrument session. The toolkit obtains this parameter from the niRFSG Initialize VI or the niRFSG Initialize With Options VI.

script specifies the script that controls waveform generation.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

instrument handle out passes a reference from the instrument session to the next VI.

script out returns the modified script that has RF blanking marker events configured.

error out contains error information. This output provides standard error out functionality.

Details

The following example will illustrate how the VI configures the script you specify:

Assume that the niRFSG RF Blanking Source property is set to “marker0”, the value of the RF Blanking Marker Positions property is {10000, 12000}, and you configure the **script** parameter as shown in the following example.

```
script scriptName
generate waveformName
end script
```

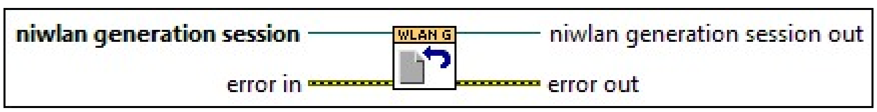
The **script out** parameter value is as shown in the following example.

```
script scriptName
generate waveformName marker0 (10000, 12000)
end script
```

niWLANG Reset Session (VI)

Owning Palette: Generation VIs

Resets all the properties of the session to their default values.



niwlan generation session specifies the niWLAN generation session refnum.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

niwlan generation session out passes a reference of the WLAN generation session to the next VI.

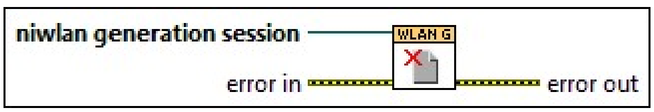
Note Close the niWLAN generation session reference using the niWLANG Close Session VI before the completion of execution to avoid possible memory leak issues.

error out contains error information. This output provides standard error out functionality.

niWLANG Close Session (VI)

Owning Palette: Generation VIs

Closes the niWLAN generation session and releases resources associated with that session. Call this VI once for each unique named session that you have created.



niwlan generation session specifies the niWLAN generation session refnum.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

error out contains error information. This output provides standard error out functionality.

Supported Standards for Properties

The following table lists the WLAN Generation properties and the standards applicable.

Property	Value of the Standard property										
	80211 A/G OFDM	80211J OFDM	80211 P OFDM	80211 B/G DSSS	80211 G DSSSOFDM	80211 N MIMO OFDM	80211 AC MIMO OFDM	80211 AH MIMO OFDM	80211 AF MIMO OFDM	80211 AX MIMO OFDM	80211 BE MIMO OFDM

Active Channel	√	√	√	√	√	√	√	√	√	√	√
Signal Settings											
Standard	√	√	√	√	√	√	√	√	√	√	√
Channel Bandwidth	√	√	√	—	—	√	√	√	√	√	√
Carrier Frequency	—	—	—	—	—	—	√	—	—	—	—
Oversampling Factor	√	√	√	√	√	√	√	√	√	√	√
Idle Interval	√	√	√	√	√	√	√	√	√	√	√
Idle Interval Mode	√	√	√	√	√	√	√	√	√	√	√
Number of Frames	√	√	√	√	√	√	√	√	√	√	√
Frame Duration (s)	√	√	√	—	—	√	√	—	—	√	√
Unframed Data Modulation Enabled	√	√	√	√	—	√	√	—	—	√	—
Waveform File Version	√	√	√	√	√	√	√	√	√	√	√
Hardware Settings (Inputs)											

Auto Headroom Enabled	√	√	√	√	√	√	√	√	√	√	√
Headroom	√	√	√	√	√	√	√	√	√	√	√
Fullscale Backoff	√	√	√	√	√	√	√	√	√	√	√
Run Time Scaling	√	√	√	√	√	√	√	√	√	√	√
Average Power Reference	—	—	—	—	—	—	—	—	—	√	√
Maximum Hardware IQ Rate	√	√	√	√	√	√	√	√	√	√	√
RF Blanking Enabled	√	√	√	√	√	√	√	√	√	√	√
LO Sharing Enabled	—	—	—	—	—	√	√	√	√	√	√
LO Frequency Offset Mode	√	√	√	√	√	√	√	√	√	√	√
LO Frequency Offset	√	√	√	√	√	√	√	√	√	√	√
Recommended Hardware Settings (Outputs)											
IQ Rate	√	√	√	√	√	√	√	√	√	√	√

Actual Headroom	√	√	√	√	√	√	√	√	√	√	√
RF Blanking Marker Positions	√	√	√	√	√	√	√	√	√	√	√
Burst Start Locations	√	√	√	√	√	√	√	√	√	√	√
Burst Stop Locations	√	√	√	√	√	√	√	√	√	√	√
Signal Bandwidth	√	√	√	√	√	√	√	√	√	√	√
Payload Settings											
A-MPDU Enabled	—	—	—	—	—	√	√	√	√	√	√
Auto Payload Number of MPDUs	—	—	—	—	—	—	—	—	—	√	√
MAC Frame Type	√	—	—	—	—	√	√	—	—	√	—
Payload Number of MPDUs	—	—	—	—	—	√	√	√	√	√	√
Auto Payload Data L	—	—	—	—	—	—	—	—	—	√	√

ength Mode											
Payload Data Length	√	√	√	√	√	√	√	√	√	√	√
Payload Data Type	√	√	√	√	√	√	√	√	√	√	√
Payload PN Order	√	√	√	√	√	√	√	√	√	√	√
Payload PN Seed	√	√	√	√	√	√	√	√	√	√	√
Payload User Defined Bits	√	√	√	√	√	√	√	√	√	√	√
MAC Header Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Frame Format	—	—	—	—	—	—	—	√	—	—	—
MAC Header Frame Control	√	√	√	√	√	√	√	√	√	√	√
MAC Header Duration/ID	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address1 Enabled	√	√	√	√	√	√	√	√	√	√	√

MAC Header Address1 Length	—	—	—	—	—	—	—	√	—	—	—
MAC Header Address1	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address2 Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address2 Length	—	—	—	—	—	—	—	√	—	—	—
MAC Header Address2	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address3 Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address3	√	√	√	√	√	√	√	√	√	√	√
MAC Header Sequence Control Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header	√	√	√	√	√	√	√	√	√	√	√

Sequence Control											
MAC Header Address4 Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header Address4	√	√	√	√	√	√	√	√	√	√	√
MAC Header QoS Control Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header QoS Control	√	√	√	√	√	√	√	√	√	√	√
MAC Header HT Control Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header HT Control	√	√	√	√	√	√	√	√	√	√	√
MAC Header Frag Num Increment Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC Header	√	√	√	√	√	√	√	√	√	√	√

Seq Num Increment Interval											
MAC Header Seq Num Increment Enabled	√	√	√	√	√	√	√	√	√	√	√
MAC FCS Enabled	√	√	√	√	√	√	√	√	√	√	√
AID12	—	—	—	—	—	—	—	—	—	√	—
AP Tx Power	—	—	—	—	—	—	—	—	—	√	—
CS Required	—	—	—	—	—	—	—	—	—	√	—
Target RSSI	—	—	—	—	—	—	—	—	—	√	—
MAC Padding Duration	√	—	—	—	—	√	√	—	—	√	—
Payload Settings (Outputs)											
MPDU Length	√	√	√	√	√	√	√	√	√	√	√
Number of Data Symbols	√	√	√	√	√	√	√	√	√	√	√
Coding Settings											
Subcarrier Mask	√	√	√	—	√	—	—	—	—	—	—

Locked Clocks Bit Enabled	—			√	√	—	—	—	—	—	—
Header Encoder Enabled	√	√	√	—	√	—	—	—	—	—	
Header Interleaver Enabled	√	√	√	—	√	—	—	—	—	—	—
Payload Scrambler Enabled	√	√	√	√	√	—	—	—	—	—	—
Payload Scrambler Seed	√	√	√	√	√	√	√	√	√	√	√
Payload Encoder Enabled	√	√	√	—	√	—	—	—	—	—	—
FEC Coding Type	—	—	—	—	—	√	√	—	√	√	√
Payload Interleaver Enabled	√	√	√	—	√	—	—	—	—	—	—
Data Rate and Frame Format Settings											
PPDU Type	—	—	—	—	—	—	√	—	√	√	√
RU Allocation Mode	—	—	—	—	—	—	—	—	—	√	—

RU Size	—	—	—	—	—	—	—	—	—	√	√
RU Offset	—	—	—	—	—	—	—	—	—	√	√
RU Allocation	—	—	—	—	—	—	—	—	—	√	—
User Enabled	—	—	—	—	—	—	—	—	—	√	√
Power Boost Factor	—	—	—	—	—	—	—	—	—	√	√
Relative Power	—	—	—	—	—	—	—	—	—	√	√
Non-HT Modulation Mode	—	—	—	—	—	√	√	—	√	—	—
OFDM Data Rate	√	√	√	—	√	—	—	—	—	—	—
Actual OFDM Data Rate	√	√	√	—	√	—	—	—	—	—	—
DSSS Data Rate	—	—	—	√	—	—	—	—	—	—	—
DSSS Preamble Type	—	—	—	√	√	—	—	—	—	—	—
STA-ID	—	—	—	—	—	—	—	—	—	√	√
MCS Index	—	—	—	—	—	√	√	√	√	√	√
DCM Enabled	—	—	—	—	—	—	—	—	—	√	—

HE-SIG-B MCS Index	—	—	—	—	—	—	—	—	—	√	—
HE-SIG-B DCM Enabled	—	—	—	—	—	—	—	—	—	√	—
EHT-SIG Compression Enabled	—	—	—	—	—	—	—	—	—	—	√
EHT-SIG MCS Index	—	—	—	—	—	—	—	—	—	—	√
80211n PLCP Frame Format	—	—	—	—	—	√	—	—	—	—	—
80211ah Preamble Type	—	—	—	—	—	—	—	√	—	—	—
Transmission Mode	—	—	—	—	—	—	—	√	—	√	√
Preamble Puncturing Enabled	—	—	—	—	—	—	—	—	—	√	√
Primary 20 MHz Channel Index	—	—	—	—	—	—	—	—	—	√	√
Preamble Pu	—	—	—	—	—	—	—	—	—	√	√

ncturing Mask											
BSS Color	—	—	—	—	—	—	—	—	—	√	√
Not Sounding Bit	—	—	—	—	—	√	—	—	—	—	—
Guard Interval Type	—	—	—	—	—	√	√	√	√	√	√
LTF Size	—	—	—	—	—	—	—	—	—	√	√
Midamble Periodicity	—	—	—	—	—	—	—	—	—	√	—
L-SIG Length	—	—	—	—	—	—	—	—	—	√	√
Pre-FEC Padding Factor	—	—	—	—	—	—	—	—	—	√	√
PE Disambiguity	—	—	—	—	—	—	—	—	—	√	√
LDPC Extra Symbol Segment	—	—	—	—	—	—	—	—	—	√	√
Nominal Packet Padding	—	—	—	—	—	—	—	—	—	√	√
Packet Extension Duration	—	—	—	—	—	—	—	—	—	√	√
MIMO Settings											

Number of Segments	—	—	—	—	—	—	√	—	—	√	—
Number of Transmit Channels	—	—	—	—	—	√	√	√	—	√	√
Number of Space Time Streams	—	—	—	—	—	√	√	√	—	√	√
Space Time Stream Offset	—	—	—	—	—	—	—	—	—	√	√
Number of HE-LTF Symbols	—	—	—	—	—	—	—	—	—	√	√
MU-MIMO LTF Mode Enabled	—	—	—	—	—	—	—	—	—	√	—
Number of Users	—	—	—	—	—	—	√	—	—	√	√
Multi-segment Generation Mode	—	—	—	—	—	—	√	—	—	—	—
Spatial Mapping Mode	—	—	—	—	—	—	—	—	—	√	√

Mapping Matrix Type	—	—	—	—	—	√	√	√	—	√	√
STBC Index	—	—	—	—	—	√	—	—	—	—	—
STBC All Streams Enabled	—	—	—	—	—	—	√	—	—	√	—
Number of Extension Spatial Streams	—	—	—	—	—	√	—	—	—	—	—
Impairments Settings											
Carrier Frequency Offset	√	√	√	√	√	√	√	√	√	√	√
Sample Clock Offset	√	√	√	√	√	√	√	√	√	√	√
IQ Impairments Enabled	√	√	√	√	√	√	√	√	√	√	√
IQ Gain Imbalance	√	√	√	√	√	√	√	√	√	√	√
I DC Offset	√	√	√	√	√	√	√	√	√	√	√
Q DC Offset	√	√	√	√	√	√	√	√	√	√	√
Quadrature Skew	√	√	√	√	√	√	√	√	√	√	√

Timing Skew	√	√	√	√	√	√	√	√	√	√	√
Time Delay	—	—	—	—	—	—	—	—	—	√	√
AWGN Enabled	√	√	√	√	√	√	√	√	√	√	√
Carrier to Noise Ratio	√	√	√	√	√	√	√	√	√	√	√
Spectrum Control Settings											
Pulse Shaping Filter Enabled	√	√	√	√	√	√	√	√	√	√	√
Pulse Shaping Filter Type	√	√	√	√	√	√	√	√	√	√	√
Pulse Shaping Filter Parameter	√	√	√	√	√	√	√	√	√	√	√
Pulse Shaping Filter Length	√	√	√	√	√	√	√	√	√	√	√
Windowing Method	√	√	√	—	√	√	√	√	√	√	√
DSSS Window Length	—	√	√	√	√	—	—	—	—	—	—
OFDM Window Length	√	√	√	—	√	√	√	√	√	√	√

Advanced Settings											
Swap I and Q Enabled	√	√	√	√	√	√	√	√	√	√	√
Sample Clock Rate Factor	√	√	√	√	√	√	√	√	√	√	√
Advanced Settings (Outputs)											
Toolkit Compatibility Version	√	√	√	√	√	√	√	√	√	√	√
IQ Waveform Size	√	√	√	√	√	√	√	√	√	√	√

niWLANGeneration Properties

Use the niWLANGeneration properties to access configuration options for WLAN applications using an NI RF vector signal generator.

Note When you query a property, the toolkit verifies the WLAN session to ensure that all relevant properties are set and are valid. If the verification fails, the toolkit returns an error.

Property	Description
Active Channel	Specifies the active channel string used to access all subsequent properties in a property node until a new active channel is specified. Details
Standard	Specifies the IEEE 802.11 standard, which indicates the type of physical layer, for signal generation. Details

Channel Bandwidth (Hz)	Specifies the channel width used for transmittin g the signal. This value is expressed in Hz. Details
Carrier Frequency (Hz)	Specifies the carrier frequency for signal genera tion. If you set the Multi-segment Generation Mode property to **Single Generator**, you must set this property for each segment using the acti ve channel string "segment**y**". This value is expr essed in Hz. Details
Oversampling Factor	Specifies the number of times the toolkit increa ses the Nyquist sample rate to obtain the final s ample rate of the signal. Details
Idle Interval (s)	Specifies the interframe spacing for signal gener ation. If you set the Idle Interval Mode property t o **Split**, the toolkit places half of the interframe spacing on either side of the burst in the genera ted waveform. If you set the Idle Interval Mode p roperty to **Post Burst**, the toolkit places the int erframe spacing at the end of the waveform. Th e waveform contains zeros for the duration of th e interframe spacing. This value is expressed in seconds. Details
Idle Interval Mode	Specifies how the idle interval is placed in the g enerated waveform. You cannot set this propert y to **Split** if you set the RF Blanking Enabled pro perty to **True**. Details
Number of Frames	Specifies the number of frames to generate. Eac h iteration of the niWLANG Create Waveform VI g enerates only one frame along with the idle inte rval that you specify using the Idle Interval prop erty. Details
Frame Duration (s)	Specifies the duration of a generated frame excl uding idle interval. This property is applicable o nly if you set Auto Data Length Mode property to **Frame Duration**. Details

Unframed Data Modulation Enabled	Specifies whether to enable Unframed Data Mo dulation. If you set this property to **True**, only the data po rtion is present in the frame, and the idle interva l is coerced to 0. Details
Hardware Settings:Auto Headroom Enabled	Specifies whether the toolkit calculates the hea droom or uses the value that you specify. For m ultiframe generation, the toolkit uses the headr oom calculated on the first frame to scale the w aveform for every frame. NI recommends that y ou do not set this property to **True** for multifra me generation because variation of the peak-to-average power ratio (PAPR) across frames may l ead to excessive clipping. To avoid excessive cli pping, set this property to **False** and use the def ault values for the Headroom property. Details
Hardware Settings:Headroom (dB)	Specifies the headroom per transmit channel. T his value represents the maximum peak-to-aver age power ratio (PAPR) allowed in the generated signal. This value is expressed in dB. Details
Hardware Settings:Fullscale Backoff (dB)	Specifies the additional scaling factor applied to the waveform when you set the Auto Headroom Enabled property to **True**. This value is express ed in dB. Details
Hardware Settings:Average Power Reference	Specifies the portions of the packet used to com pute the average power value to apply wavefor m power scaling and additive white Gaussian no ise (AWGN). This property is valid when you set t he Standard property to **80211AX MIMOOFD M** and the PPDU Type property to **MU PPDU** or **Extended Range SU PPDU**. Details
Hardware Settings:Maximum Hardware IQ Rate (S/s)	Specifies the maximum I/Q rate that the NI vect or signal generator supports. This property will be set according to the device model in the niWLANG RFSG Create and Download Waveform VI. Details

Hardware Settings:RF Blanking Enabled	Specifies whether to enable RF blanking. If you want to attenuate the RF OUT signal durin g the idle interval, set this property to **True**. Thi s behavior prevents any DC leakage from the loc al oscillator of the signal generator from appeari ng at the RF OUT signal. RF blanking attenuates the RF OUT signal of signal generators quickly. Details
Hardware Settings:LO Sharing Enabled	Specifies whether to enable configuration for sh aring of local oscillator (LO) signal for multiple N I RF vector signal analyzers or vector signal tran sceivers. This property is read when you call the niWLANG RFSG Configure Multiple Device Synchronization VI. Details
Hardware Settings:LO Frequency Offset Mode	Specifies how the LO frequency offset is derived to configure frequency on the NI RF vector signa l generators and the NI synthesizers in the niWLANG RFSG Configure Frequency VI. The tool kit ignores this property if you do not use PXIe-5 840, PXIe-5841, PXIe-5841 with PXIe-5655, PXIe-5646, PXIe-5830, or PXIe-5831 devices. Details
Hardware Settings:LO Frequency Offset (Hz)	Specifies the LO frequency offset to be used wh en you set the LO Frequency Offset Mode proper ty to **User Defined**. Details
Hardware Settings:Recommended Settings:IQ R ate (S/s)	Returns the recommended sample rate for the c urrent signal configuration. The **dt** parameter of the created waveform is the inverse of the reco mmended sample rate. If you set the **use wave form dt for IQ rate?** parameter of the niRFSG Write Arb Waveform VI to **False**, wire the IQ Rate property to the niRFSG IQ Rate property. This va lue is expressed in samples per second. Details
Hardware Settings:Recommended Settings:Actu al Headroom (dB)	Returns the actual headroom that the toolkit ap plies to the waveform. Use an active channel stri

	ng to query this property for a transmit channel. This value is expressed in dB. Details
Hardware Settings:Recommended Settings:RF Blanking Marker Positions []	Returns the array of sample positions of marker events, which are used to toggle the state of RF blanking, within the waveform. Details
Hardware Settings:Recommended Settings:Burst Start Locations []	Returns the array of sample positions of start of the burst, within the waveform. Details
Hardware Settings:Recommended Settings:Burst Stop Locations []	Returns the array of sample positions of end of the burst, within the waveform. Details
Hardware Settings:Recommended Settings:Signal Bandwidth (Hz)	Returns the Signal Bandwidth value that needs to be configured on the NI RFSG session for the PXIe-5820, PXIe-5830, PXIe-5831, PXIe-5841, or PXIe-5841 with PXIe-5655 devices. Details
Payload:A-MPDU Enabled	Specifies whether all medium access control (MAC) protocol data units (MPDUs) are transmitted as aggregate-MPDU (A-MPDU). Details
Payload:Auto Number of MPDUs	Specifies whether to compute the number of MPDUs in an AMPDU of 802.11ax Trigger-Based PPDU using the following trigger frame parameters; L-SIG Length Property, Pre-FEC Padding Factor Property, PE Disambiguity Property, and LDPC Extra Symbol Segment Property. Details
Payload:MAC Frame Type	Specifies the type of frame for the MPDU. This property can be set to Trigger Frame only if you set the Standard property to **80211A/G OFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, or **80211AX MIMOOFDM**. Details
Payload:Number of MPDUs	Specifies the number of medium access control (MAC) protocol data units (MPDUs) to combine into one aggregate-MPDU (A-MPDU). Details
Payload:Auto Data Length Mode	Specifies whether to use the value specified in Data Length (bytes) property or automatically compute the data length of the MPDUs in an AMPDU. Details

Payload:Data Length (bytes)	Specifies the length of the payload, excluding the lengths of the medium access control (MAC) header and frame check sequence (FCS). The payload length encoded into the physical layer (PHY) header is the sum of the payload data length and the lengths of the MAC header and FCS. This value is expressed in bytes. Details
Payload:Data Type	Specifies the type of payload for waveform generation. Details
Payload:PN Order	Specifies the order (length of memory) of the pseudorandom bit sequence (PRBS) generator. Details
Payload:PN Seed	Specifies the initialization seed used for the pseudorandom bit sequence (PRBS) generator. If you set the Payload Data Type property to **User Defined Pattern**, the toolkit ignores the Payload PN Seed property. Details
Payload:User Defined Bits []	Specifies a user-defined bit pattern as an array of zeros and ones. If the array length is greater than the configured payload data length, the toolkit uses a subset of the required length from the beginning of the array for waveform generation. If the array length is less than the configured payload length, the toolkit repeats the user-defined bit pattern until the required length is achieved. If you set the Payload Data Type property to **PN Sequence**, the toolkit ignores the Payload User Defined Bits property. Details
Payload:MAC Frame Format	Specifies whether the medium access control (MAC) frame is long or short. Details
Payload:MAC Header:Enabled	Specifies whether to enable the medium access control (MAC) header, as defined in section 7.1.2 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3** and section 9.2.4 of **IEEE P802.11ax/D6.0**. Details

Payload:MAC Header:Frame Control	Specifies the two-byte frame control field of the medium access control (MAC) header as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) at the rightmost position. Details
Payload:MAC Header:Duration/ID	Specifies the two-byte duration ID field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) in the rightmost position. Details
Payload:MAC Header:Address1 Enabled	Specifies whether to enable the Address1 field of the medium access control (MAC) header. Details
Payload:MAC Header:Address1 Length (bytes)	Specifies the length of Address1 field when you set the MAC Frame Format property to Short. Details
Payload:MAC Header:Address1	Specifies the Address1 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. The length of this field is defined by the Address1 Length property if you set the MAC Frame Format property to Short. In all other instances, the length of this field is six bytes. If this field is a MAC address, it is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position. Details

Payload:MAC Header:Address2 Enabled	Specifies whether to enable the Address2 field o f the medium access control (MAC) header. Details
Payload:MAC Header:Address2 Length (bytes)	Specifies the length of Address2 field when the MAC Frame Format is Short. Details
Payload:MAC Header:Address2	Specifies the Address2 field as defined in sectio n 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE S tandard 802.11ac-2013** and **IEEE P802.11ah/D 1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. The length of this field is defined by the Address2 Length property if you set the MAC Frame Format property to Short. In all other ins tances, the length of the field is six bytes. If this f ield is a MAC address, it is represented by the lea st significant byte in the leftmost position, and e ach byte is represented by the least significant b it in the rightmost position. Details
Payload:MAC Header:Address3 Enabled	Specifies whether to enable the Address3 field o f the medium access control (MAC) header. Details
Payload:MAC Header:Address3	Specifies the six-byte Address3 field as defined i n section 7.1.3 of **IEEE Standard 802.11-2007** a nd **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P80 2.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11 ax/D6.0**. This field is represented with the least significant byte in the leftmost position, and eac h byte is represented with the least significant b it in the rightmost position. Details
Payload:MAC Header:Sequence Control Enabled	Specifies whether to enable the Sequence Contr ol field of the medium access control (MAC) hea der. Details
Payload:MAC Header:Sequence Control	Specifies the two-byte Sequence Control field a s defined in section 7.1.3 of **IEEE Standard 802. 11-2007**, **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE**

	P802.11ah/D1.3, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position. Details
Payload:MAC Header:Address4 Enabled	Specifies whether to enable the Address4 field of the medium access control (MAC) header. Details
Payload:MAC Header:Address4	Specifies the six-byte Address4 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position. Details
Payload:MAC Header:QoS Control Enabled	Specifies whether to enable the QoS Control field of the medium access control (MAC) header. Details
Payload:MAC Header:QoS Control	Specifies the two-byte QoS Control field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position. Details
Payload:MAC Header:HT Control Enabled	Specifies whether to enable the HT Control field of the medium access control (MAC) header. Details
Payload:MAC Header:HT Control	Specifies the four-byte HT Control field as defined in section 7.1.3 of **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position. Details

Payload:MAC Header:Sequence Control:Sequence Number Increment Enabled	Specifies whether to increment the sequence number in a sequence of frames. Details
Payload:MAC Header:Sequence Control:Sequence Number Increment Interval (frames)	Specifies the number of frames after which the sequence number is incremented by 1. The starting number is the value represented by the sequence number sub-field of the Sequence Control field. The sequence number is wrapped to 0 after reaching the value 4,095 or (2 -1). Details
Payload:MAC Header:Sequence Control:Fragment Number Increment Enabled	Specifies whether to increment the fragment number in a sequence of frames. The starting number is the value represented by the fragment number sub-field of the MAC Header Sequence Control property. If you set the MAC Header Frag Num Increment Enabled property to **True**, the toolkit increments the fragment number by 1 for every successive frame having the same sequence number. The fragment number wraps to the starting number when the sequence number increments. The fragment number wraps to 0 after reaching the value 15. Details
Payload:MPDU Length (bytes)	Returns the length of the medium access control (MAC) protocol data unit (MPDU). An MPDU comprises of a MAC header, a frame body, and a frame check sequence (FCS). The MPDU Length property is the sum of the length of MAC header, the value of the Payload Data Length property, and the length of FCS, which is equal to four bytes. If you disable the MAC Header Enabled and MAC FCS Enabled properties, the lengths of MAC header and FCS are zero. This value is expressed in bytes. Details
Payload:MAC FCS Enabled	Specifies whether to enable the medium access control (MAC) frame check sequence (FCS), as defined in section 7.1.2 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013**, **IEEE P802.11ah/D1.3** and **IEEE Standard P802.11af-20**

	13, and section 9.2.4 of **IEEE P802.11ax/D6.0**. Details
Payload:Trigger Frame:AID12	Specifies the value of the AID12 field in the trigg er frame. Details
Payload:Trigger Frame:AP Tx Power	Specifies the value of the AP Tx Power field of th e trigger frame. The power values -20 dBm to 40 dBm are mapped to the field values 0 to 60, resp ectively. Details
Payload:Trigger Frame:CS Required	Specifies the CS required sub-field in the 802.11 ax Trigger Frame. Details
Payload:Trigger Frame:Target RSSI	Specifies the value of the UL-Target RSSI field of the trigger frame. The power values -110 dBm to -20 dBm are mapped to the field values 0 to 90, r espectively. To specify the maximum transmit p ower for the assigned MCS you must set the valu e of 127. Details
Payload:Trigger Frame:MAC Padding Duration (s)	Specifies the padding duration when the MAC Frame Type property is set to **Trigger Frame**. T his property is valid if you set the Standard prop erty to **80211A/G OFDM**, **80211N MIMOOFD M**, **80211AC MIMOOFDM**, or **80211AX MIMO OFDM**. Details
Payload:Number of Data Symbols	Returns the number of symbols in the data porti on of the WLAN frame. The symbol refers to the chip if the Standard property is set to **80211B/ G DSSS**, and the symbol refers to the OFDM sy mbol for other values of the Standard property. Details
Coding:Subcarrier Mask	Specifies the sequence of attenuation values on each subcarrier in the signal and payload symb ols if you set the Standard property to **80211A/ G OFDM**, **80211J OFDM**, **80211P OFDM** or **8 0211G DSSSOFDM**. You must specify a 64-ele ment array. The first element of the array corres ponds to subcarrier index-32, and the 64th elem ent corresponds to subcarrier index 31, as defin

	ed in section 17.3.2.5 of **IEEE Standard 802.11a-1999**. Details
Coding:Locked Clocks Bit Enabled	Specifies whether to enable the Locked Clocks Bit flag for the direct sequence spread spectrum (DSSS) header, as defined in sections 18.2.3.4 and 18.2.3.11 of **IEEE Standard 802.11b-1999** and section 19.3.2.1 of **IEEE Standard 802.11g-2003**. Details
Coding:Header Encoder Enabled	Specifies whether to enable convolutional encoding of the OFDM SIGNAL field, as defined in section 17.3.5.5 of **IEEE Standard 802.11a-1999**. Details
Coding:Header Interleaver Enabled	Specifies whether to enable interleaving for the OFDM SIGNAL field, as defined in section 17.3.5.6 of **IEEE Standard 802.11a-1999**. Details
Coding:Payload Scrambler Enabled	Specifies whether to enable scrambling of the payload for OFDM packets and the entire burst for direct sequence spread spectrum (DSSS) packets, as defined in section 17.3.5.4 of **IEEE Standard 802.11a-1999** and section 18.2.4 of **IEEE Standard 802.11b-1999**. Details
Coding:Payload Scrambler Seed	Specifies the initial state of the scrambler seed. Details
Coding:Payload Encoder Enabled	Specifies whether to enable convolutional encoding of the OFDM payload, as defined in section 17.3.5.5 of **IEEE Standard 802.11a-1999**. Details
Coding:FEC Coding Type	Specifies the type of forward error correction (FEC) coding to use if you set the Standard property to 80211N MIMOOFDM, 80211AC MIMOOFDM, 80211AF MIMOOFDM or 80211AX MIMOOFDM, as defined in section 20.3.11.3 of **IEEE Standard 802.11n-2009**. Details

Coding:Payload Interleaver Enabled	Specifies whether to enable interleaving for the OFDM payload, as defined in section 17.3.5.6 of **IEEE Standard 802.11a-1999**. Details
Data Rate and Frame Format:PPDU Type	Specifies the type of physical layer convergence procedure (PLCP) protocol data unit (PPDU), if the Standard property is set to **80211AC MIMO OFDM** or **80211AX MIMOOFDM**. Details
Data Rate and Frame Format:RU Allocation Settings:Mode	Specifies how to configure the multi-user allocation in a 802.11ax MU PPDU signal. Details
Data Rate and Frame Format:RU Allocation Settings:RU Size	Specifies the size of resource unit (RU) in terms of the number of subcarriers for the 802.11ax signal. You must configure this property when you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**. Details
Data Rate and Frame Format:RU Allocation Settings:RU Offset	Specifies the location of the resource unit (RU), in terms of the index of 26-tone RU, assuming the entire bandwidth is composed of 26-tone RUs in the 802.11ax signal. For example, in the following figure, to specify the 106-tone RU second from left, you must configure this property to 5. Details
Data Rate and Frame Format:RU Allocation Settings:RU Allocation	Specifies the common field of the HE-SIG-B to be used for multi-user allocation in a 802.11ax MU PPDU signal. Details
Data Rate and Frame Format:RU Allocation Settings:User Enabled	Specifies whether to enable the user in a 802.11ax MU PPDU signal. Details
Data Rate and Frame Format:Power Boost Factor	Specifies the factor, per resource unit (RU), by which the amplitude of the HE modulated fields in 802.11ax signals are scaled, when you set the PPDU Type property to **MU PPDU**. The value of this property must be the same across all users within the RU. An RU is defined by the RU Size and RU Offset properties. Details
Data Rate and Frame Format:Relative Power (dB)	Specifies the per user power scaling value of the 802.11ax signal when you set the PPDU Type pr

	operty to **Trigger-Based PPDU**. This value is expressed in dB. The power scaling value is with reference to the user with index 0. The toolkit ignores the property value, if it is specified for the user with index 0. Details
Data Rate and Frame Format:Non-HT Modulation Mode	Specifies whether the format of the incoming OFDM signal is non-high throughput (HT). This property is valid only if the Standard property is set to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, or **80211AF MIMOOFDM**. Details
Data Rate and Frame Format:OFDM Data Rate (Mbps)	Specifies the data rate for the OFDM payload, as defined in section 17.3.2.2 of **IEEE Standard 802.11-2007**. This value is expressed in Mbps. Details
Data Rate and Frame Format:Actual OFDM Data Rate (Mbps)	Returns the OFDM data rate depending upon the values of the Channel Bandwidth and OFDM Data Rate properties. This query is only successful if you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**, or if you set the Non-HT Modulation Mode property to **On**. For more information about the OFDM data rate, refer to section 17.2.3.3 of **IEEE Standard 802.11-2007**. This value is expressed in Mbps. Details
Data Rate and Frame Format:DSSS Data Rate (Mbps)	Specifies the data rate for the direct sequence spread spectrum (DSSS) payload, as defined in **IEEE Standard 802.11b-1999** and the extended rate physical layer-packet binary convolutional coding (ERP-PBCC) mode in **IEEE Standard 802.11g-2003**. This value is expressed in Mbps. Details
Data Rate and Frame Format:DSSS Preamble Type	Specifies whether to use a long or short preamble for direct sequence spread spectrum (DSSS) and DSSS-OFDM packets, as defined in **IEEE Standard 802.11b-1999**. Details

Data Rate and Frame Format:STA-ID	Specifies 11 LSBs of the association identifier (AID) in a 802.11ax signal when you set the PPDU Type property to MU PPDU. Details
Data Rate and Frame Format:MCS Index	Specifies the value of the modulation and coding scheme (MCS) index. The MCS index is a compact representation that determines the modulation scheme, coding rate, and number of spatial streams, as specified in section 20.3.5 of **IEEE Standard 802.11n-2009**, section 22.5 of **IEEE Standard 802.11ac-2013**, section 24.5 of **IEEE P802.11ah/D1.3**, section 23.5 of **IEEE Standard 802.11af-2013**, and section 27.5 of **IEEE P802.11ax/D6.0**. Details
Data Rate and Frame Format:Dual Carrier Modulation Enabled	Specifies whether the dual carrier modulation (DCM) is applied to the data part of the 802.11ax signals or not. The property can be set to True only when the MCS index is 0, 1, 3 or 4, and the number of spatial streams is 1 or 2. Details
Data Rate and Frame Format:HE-SIG-B MCS Index	Specifies the value of the modulation and coding scheme (MCS) index of the HE-SIG-B field of 802.11ax signal when you set the PPDU Type property to MU PPDU. Details
Data Rate and Frame Format:HE-SIG-B Dual Carrier Modulation Enabled	Specifies whether the dual carrier modulation (DCM) is applied on the HE-SIG-B field of 802.11ax signals or not. The property can be set to True only when HE-SIG-B MCS index is 0, 1, 3 or 4. Details
Data Rate and Frame Format:80211n PLCP Frame Format	Specifies the format of the physical layer convergence protocol (PLCP) frame structure. The frame structure determines the arrangement of preambles, header (SIGNAL field), and payload in a frame, as defined in section 20.3.2 of **IEEE Standard 802.11n-2009**. Details
Data Rate and Frame Format:80211ah Preamble Type	Specifies the preamble type of packet if you set the Standard property to 80211AH MIMOOFDM. For more information about 80211ah Pream

	ble Type, refer to section 24.3.8.2 of **IEEE Standard P802.11ah/D1.3**. Details
Data Rate and Frame Format:Transmission Mode	Specifies the value of the uplink indication field of the S1G-SIG field when you set the Standard property to 80211AH MIMOOFDM. This property also specifies whether the packet is uplink or downlink when you set the Standard property to 80211AX MIMOOFDM. Details
Data Rate and Frame Format:Preamble Puncturing:Enabled	Specifies whether to enable preamble puncturing (channel puncturing) on 802.11ax MU PPDU signals. Preamble puncturing is valid only when you set the Channel Bandwidth property to 80 MHz or 160 MHz. Details
Data Rate and Frame Format:Preamble Puncturing:Primary 20 MHz Channel Index	Specifies the index of the primary 20 MHz sub-channel in the channel bandwidth. This property along with the puncturing information defines the mode of puncturing. Details
Data Rate and Frame Format:Preamble Puncturing:Mask	Specifies the 20 MHz sub-channels to be punctured in the 802.11ax MU PPDU signal when preamble puncturing is enabled. The mask value specified here is a binary mask represented as an integer, where bit '0' represents the punctured sub-channel. Details
Data Rate and Frame Format:BSS Color	Specifies the identifier of the BSS (Basic Service Set) from which the 802.11ax PPDU is transmitted. Details
Data Rate and Frame Format:Not Sounding Bit	Specifies the value of the Not Sounding field of the HT-SIG field when you set the Standard property to 80211N MIMOOFDM. For more information about Not Sounding Bit, refer to section 20.3.9.4.3 of **IEEE Standard 802.11n-2009**. Details
Data Rate and Frame Format:Guard Interval Type	Specifies the type of guard interval (cyclic prefix) in an OFDM symbol. Details
Data Rate and Frame Format:HE-LTF Size	Specifies the HE-LTF symbol size in the 802.11ax signals. The IEEE Standard 802.11ax specifies th

	e following combinations of the HE-LTF symbol size and the Guard Interval property. Details
Data Rate and Frame Format:Midamble Periodic ity	Specifies the interval, in number of data symbol s, after which the midambles are inserted in the data field of the 802.11ax signals. You must set t his property to **None** when the value of the Number of Space Time Streams property is grea ter than 4. Details
Data Rate and Frame Format:L-SIG Length	Specifies the value of the UL-LENGTH field in the trigger frame that is used for 802.11ax Trigger-B ased PPDU generation. Details
Data Rate and Frame Format:Pre-FEC Padding F actor	Specifies the value of pre-FEC padding factor su b-field in the trigger frame that is used for 802.1 1ax Trigger-Based PPDU generation. Details
Data Rate and Frame Format:PE Disambiguity	Specifies the value of the PE disambiguity sub-fi eld in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation. Details
Data Rate and Frame Format:LDPC Extra Symbo l Segment	Specifies the value of the LDPC extra symbol seg ment field in the trigger frame which is used for 802.11ax Trigger-Based PPDU generation. Details
Data Rate and Frame Format:Nominal Packet Pa dding	Specifies the nominal packet padding value use d for determining the packet extension duration when you set the Standard property to **80211A X MIMOOFDM**. Details
Data Rate and Frame Format:Packet Extension D uration (s)	Returns the duration of packet extension in the waveform when you set the Standard property t o **80211AX MIMOOFDM**. This value is expresse d in seconds. Details
MIMO:Number of Segments	Specifies the number of frequency segments for 802.11ac or 802.11ax signals. Details
MIMO:Number of Transmit Channels	Specifies the number of transmit channels for m ultiple input multiple output (MIMO) signals, as

	defined in section 20.3.3 of **IEEE Standard 802.11n-2009**. Details
MIMO:Number of Space Time Streams	Specifies the number of space-time streams into which the data is divided. Details
MIMO:Space Time Stream Offset	Specifies the space time stream offset which is used for 802.11ax Trigger-Based PPDU generation. Details
MIMO:Number of HE-LTF Symbols	Specifies the number of HE-LTF symbols in the transmitted 802.11ax signal when you set the PPDU Type property to **Trigger-Based PPDU**. Details
MIMO:MU-MIMO LTF Mode Enabled	Specifies whether the HE-LTF sequence corresponding to each space time stream is masked by a distinct orthogonal code if you set the Standard property to **80211AX MIMOOFDM** and PPDU Type property to **Trigger-Based PPDU**. Details
MIMO:Number of Users	Specifies the number of users in a multi-user (MU) physical layer convergence procedure (PLCP) protocol data unit (PPDU). Details
MIMO:Multi-segment Generation Mode	Specifies whether to use a single generator or two generators for each channel of a multi-segment (80+80) MHz 802.11ac signal. This property is applicable when the Number of Segments property is 2 and the channel bandwidth is 80 MHz. When you set this property to **Single Generator**, you have to specify the value of the Carrier Frequency property for both the segments. Details
MIMO:Spatial Mapping Mode	Specifies whether the spatial mapping is created from a single global matrix or per user. This property is valid, when you set the PPDU Type property to **Trigger-Based PPDU**. Details
MIMO:Mapping Matrix Type	Specifies the mapping matrix type for mapping space-time streams to transmit channels as defi

	ned in section 20.3.11.10.1 of **IEEE Standard 802.11n-2009**. Details
MIMO:STBC Index	Specifies the difference between the number of space-time streams and the number of spatial streams, as defined in section 20.3.9.4.3 of **IEEE Standard 802.11n-2009**. The toolkit derives the number of spatial streams from the specified value of the MCS Index property. Different space-time coding schemes are defined in section 20.3.11.8.1 of **IEEE Standard 802.11n-2009**. Details
MIMO:STBC All Streams Enabled	Specifies whether space-time block coding (STBC) is performed at the transmitter when the Standard property is set to 80211AC MIMOOFDM or 80211AX MIMOOFDM. Whenever STBC is performed, the number of space-time streams is equal to two times the number of spatial streams. Details
MIMO:Number of Extension Spatial Streams	Specifies the number of extension spatial streams (**N_{ESS}**) as defined in section 20.3.9.4.6 of **IEEE Standard 802.11n-2009**. Details
Impairments:Carrier Frequency Offset (Hz)	Specifies the offset in the center frequency of the complex baseband signal from the carrier frequency. This value is expressed in Hz. Details
Impairments:Sample Clock Offset (ppm)	Specifies the offset in the Sample Clock frequency from the sampling frequency defined by the following equation: **Sampling Frequency** = max (maximum hardware I/Q Rate, **Oversampling Factor** * **Nyquist Sampling Rate**) where **Maximum Hardware I/Q** is the value of the Maximum Hardware IQ Rate property **Oversampling Factor** is the value of the Oversampling Factor property

	This value is expressed in parts per million (ppm). Details
Impairments:IQ Impairments:Enabled	Specifies whether to apply I/Q impairments such as I DC offset, Q DC offset, quadrature skew, and I/Q gain imbalance to the waveform. Details
Impairments:IQ Impairments:IQ Gain Imbalance (dB)	Specifies the ratio of the mean amplitude of the in-phase (I) signal to the mean amplitude of the quadrature-phase (Q) signal. This value is expressed in dB. Details
Impairments:IQ Impairments:I DC Offset (%)	Specifies the value of the DC offset in the in-phase (I) channel as a percentage of the RMS magnitude of the unaltered I channel. Details
Impairments:IQ Impairments:Q DC Offset (%)	Specifies the value of the DC offset in the quadrature-phase (Q) channel as a percentage of the RMS magnitude of the unaltered Q channel. Details
Impairments:IQ Impairments:Quadrature Skew (deg)	Specifies the deviation in angle from 90 degrees between the in-phase (I) and quadrature-phase (Q) signals. This value is expressed in degrees. Details
Impairments:IQ Impairments:Timing Skew (s)	Specifies the difference between the sampling instants of I and Q streams. This value is expressed in seconds. Details
Impairments:Time Delay (s)	Specifies the time delay for each user within an 802.11ax Trigger-Based signal. This value is expressed in seconds. You must set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **Trigger-Based PPDU**. Use this property to introduce relative time delays between multiple users within an 802.11ax Trigger-Based signal. Details
Impairments:AWGN:Enabled	Specifies whether to introduce additive white Gaussian noise (AWGN) to the baseband waveform. The toolkit uses the value specified in the

	Carrier to Noise Ratio property to add the AWGN. Details
Impairments:AWGN:Carrier to Noise Ratio (dB)	Specifies the carrier-to-noise ratio (CNR) of the waveform generated. Noise bandwidth is equal to the value of the IQ Rate property. The toolkit ignores the Carrier to Noise Ratio property if you set the AWGN Enabled property to **False**. This value is expressed in dB. Details
Spectrum Control:Pulse Shaping Filter Enabled	Specifies whether to apply a pulse-shaping filter to the generated signal. Details
Spectrum Control:Pulse Shaping Filter Type	Specifies the pulse-shaping filter type to use to ensure that the signal spectrum meets the spectral mask criteria as defined in section 17.3.9.2 of **IEEE Standard 802.11a-1999**, section 18.4.7.3 of **IEEE Standard 802.11b-1999**, and section 20.3.21.1 of **IEEE Standard 802.11n-2009**. Details
Spectrum Control:Pulse Shaping Filter Parameter	Specifies the value of the rolloff factor (alpha) if you set the Pulse Shaping Filter Type property to **Raised Cosine** or **Root Raised Cosine**. If you set the Pulse Shaping Filter Type property to **Gaussian**, you can calculate the Pulse Shaping Filter Parameter property by multiplying **B** and **T**, where **B** is the 3 dB bandwidth and **T** is the symbol period for a Gaussian filter. If you set the Pulse Shaping Filter Type property to **Rectangular**, the toolkit ignores the Pulse Shaping Filter Parameter property. Details
Spectrum Control:Pulse Shaping Filter Length	Specifies the length of the pulse-shaping filter. This value is expressed in symbols. Details
Spectrum Control:Windowing Method	Specifies the method of applying window to the baseband signal if the Standard property is set to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211G DSSSOFDM**, **80211N MIMO OFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**. This property is ignored if the S

	tandard property is set to **80211B/G DSSS**. Details
Spectrum Control:DSSS Window Length (s)	Specifies the window length for direct spread sp ectrum signals If you do not want windowing, se t this property to 0. This value is expressed in se conds. Details
Spectrum Control:OFDM Window Length (sampl es)	Specifies the window length, for OFDM signals a t the sampling rate equal to the channel bandwi dth. For example, if the window length is 2, the c hannel bandwidth is 20 MHz and the oversampli ng factor is 4, the samples over which windowin g is applied is 8. This value is expressed in sampl es. Details
Advanced:Swap I and Q Enabled	Specifies whether to swap the data in the I and Q streams. Details
Advanced:Toolkit Compatibility Version	Indicates the **toolkit compatibility version** p arameter of the niWLANG Open Session VI. Details
Advanced:IQ Waveform Size	Returns the size of the generated I/Q waveform. This value is expressed in samples. Details
Advanced:Sample Clock Rate Factor	Specifies the factor by which the Sample Clock r ate is multiplied to generate a signal that is com pressed in the frequency domain and expanded in the time domain. Details
Obsolete:Legacy Scaling Enabled	Specifies whether to enable standard-defined tr ansmit chain scaling of the legacy part of the hig h throughput (HT) and very high throughput (VH T) frames. Details
Obsolete:Power Level (dBm)	Specifies the average power level of the active p ortion of the burst for signal generation. The act ive portion of the burst is the WLAN packet exclu ding the interframe spacing. This value is expres sed in dBm. Details
Obsolete:Waveform Scaling Factor	Specifies the scaling factor for the waveform, as a percentage of the maximum sample magnitud

	e, to reduce the overshoot associated with the d igital-to-analog converter (DAC) interpolation fil ter and other finite impulse response (FIR) filter s in the NI RF vector signal generators. Details
Obsolete:Peak to Average Power Ratio (dB)	Returns the peak-to-average power ratio (PAPR) of the output complex waveform. This value is e xpressed in dB. Details
Obsolete:Guard Interval (s)	Specifies the length of the cyclic prefix (CP) of a n OFDM symbol, as specified in section 20.1.1 of **IEEE Standard 802.11n-2009** and section 22.3. 6 of **IEEE Standard 802.11ac-2013**. This value is expressed in nanoseconds. Details

Active Channel Property

Short Name: Active Channel

Property of niWLANGeneration

Specifies the active channel string used to access all subsequent properties in a property node until a new active channel is specified.

If the property you want to use is channel-specific, you must first select the Active Channel property and then pass the name of the specific channel. If you specify an active channel for a property that is not channel-specific, the toolkit returns an error.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Write Only
High-level VIs	N/A
Resettable	No

Standard Property

Short Name: Standard

Property of niWLANGeneration

Specifies the IEEE 802.11 standard, which indicates the type of physical layer, for signal generation.

Note If you do not select a standard, the toolkit returns an error.

80211A/G OFDM (0)	Corresponds to the OFDM mode defined in **IEEE Standard 802.11a-1999**, and the extended rate physical layer-OFDM (ERP-OFDM) mode defined in **IEEE Standard 802.11g-2003**.
80211J OFDM (7)	Corresponds to the OFDM mode defined in **IEEE Standard 802.11j 2004**.
80211P OFDM (8)	Corresponds to the OFDM mode defined in **IEEE Standard 802.11p 2010**.
80211B/G DSSS (1)	Corresponds to all the compulsory and optional modes defined in **IEEE Standard 802.11b-1999** and the ERP-packet binary convolutional coding (ERP-PBCC) mode defined in **IEEE Standard 802 .11g-2003**.
80211G DSSSOFDM (2)	Corresponds to the optional direct sequence spr ead spectrum-OFDM (DSSS-OFDM) mode define d in **IEEE Standard 802.11g-2003**.
80211N MIMOOFDM (3)	Corresponds to **IEEE Standard 802.11n-2009**. T o use this option, you must set the toolkit com patibility version parameter on the niWLANG Open Session VI to 2.0.0, 3.0.0, 4.0.0, or 5.0.0.
80211AC MIMOOFDM (4)	Corresponds to **IEEE Standard 802.11ac-2013**. To use this option, you must set the toolkit co mpatibility version parameter on the niWLAN G Open Session VI to 3.0.0, 4.0.0, or 5.0.0.

80211AH MIMOOFDM (5)	Corresponds to **IEEE P802.11ah/D1.3**. To use this option, you must set the **toolkit compatibility version** parameter on the niWLANG Open Session VI to **3.0.0**, **4.0.0**, or **5.0.0**.
80211AF MIMOOFDM (6)	Corresponds to **IEEE Standard 802.11af-2013**. To use this option, you must set the **toolkit compatibility version** parameter on the niWLANG Open Session VI to **3.0.0**, **4.0.0**, or **5.0.0**.
80211AX MIMOOFDM (9)	Corresponds to **IEEE P802.11ax/D6.0**. To use this option, you must set the **toolkit compatibility version** parameter on the niWLANG Open Session VI to **3.0.0**, **4.0.0** , or **5.0.0**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Standard
Resettable	No

Channel Bandwidth (Hz) Property

Short Name: Channel Bandwidth (Hz)

Property of niWLANGeneration

Specifies the channel width used for transmitting the signal. This value is expressed in Hz.

The valid values are shown in the following table.

Standard Property Value	Channel Bandwidth (MHz)
80211A/G OFDM	20
80211J OFDM	10 or 20
80211P OFDM	5, 10 or 20

80211N MIMOOFDM	20 or 40
80211AC MIMOOFDM	20, 40, 80, or 160
80211AX MIMOOFDM	20, 40, 80, 160, or 320
80211AH MIMOOFDM	1, 2, 4, 8, or 16
80211AF MIMOOFDM	6, 7, or 8

For OFDM signals, channel bandwidth determines the number of pilot and data subcarriers used. The toolkit ignores the Channel Bandwidth property for other values of the Standard property.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niWLANG Set Channel Bandwidth
Resettable	No

Carrier Frequency (Hz) Property

Short Name: Carrier Frequency (Hz)

Property of niWLANGeneration

Specifies the carrier frequency for signal generation. If you set the Multi-segment Generation Mode property to **Single Generator**, you must set this property for each segment using the active channel string "segment**y**". This value is expressed in Hz.

The default value is 2.412 GHz.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write

High-level VIs	N/A
Resettable	No

Oversampling Factor Property

Short Name: Oversampling Factor

Property of niWLANGeneration

Specifies the number of times the toolkit increases the Nyquist sample rate to obtain the final sample rate of the signal.

The default value is 4, and the minimum value is 2.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Oversampling Factor
Resettable	No

Idle Interval (s) Property

Short Name: Idle Interval (s)

Property of niWLANGeneration

Specifies the interframe spacing for signal generation. If you set the Idle Interval Mode property to **Split**, the toolkit places half of the interframe spacing on either side of the burst in the generated waveform. If you set the Idle Interval Mode property to **Post Burst**, the toolkit places the interframe spacing at the end of the waveform. The waveform contains zeros for the duration of the interframe spacing. This value is expressed in seconds.

Tip For higher values of idle interval, LabVIEW may run out of memory. For a large idle interval, you can create a small idle waveform and

	generate the waveform multiple times using scripting.

The default value is 100 microseconds. Valid values are 0 to 1, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	niWLANG Set Idle Interval
Resettable	No

Idle Interval Mode Property

Short Name: Idle Interval Mode

Property of niWLANGeneration

Specifies how the idle interval is placed in the generated waveform. You cannot set this property to **Split** if you set the RF Blanking Enabled property to **True**.

The default value is **Split**.

Split (0)	Half the idle interval is placed on either side of t he burst in the generated waveform.
Post Burst (1)	Entire idle interval is placed at the end of the bu rst in the generated waveform.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Number of Frames Property

Short Name: Number of Frames

Property of niWLANGeneration

Specifies the number of frames to generate. Each iteration of the niWLANG Create Waveform VI generates only one frame along with the idle interval that you specify using the Idle Interval property.

If you set the Number of Frames property to a value greater than 1, create a loop around the niWLANG Create Waveform VI and concatenate the output values from different iterations. If you encounter memory usage issues, download a single frame to the NI RF vector signal generator memory on each iteration.

To generate the required number of frames, wire the Number of Frames property to the loop iteration count. You also can use the **generation done?** parameter on the niWLANG Create Waveform VI as a termination signal.

You can use the niRFSG Allocate Arb Waveform VI to preallocate arb memory and then download the waveform frame-by-frame.

The default value is 1. Valid values are 1 to 1,000 (inclusive).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Number of Frames
Resettable	No

Frame Duration (s) Property

Short Name: Frame Duration (s)

Property of niWLANGeneration

Specifies the duration of a generated frame excluding idle interval. This property is applicable only if you set Auto Data Length Mode property to **Frame Duration**.

When the value of **Frame Duration** results in fractional **Number of Data Symbols**, and if it results in generated frame duration less than 5.484 milliseconds, the number of data symbols is rounded to the next highest integer, otherwise, the number of symbols is rounded to the next lowest integer.

The default value is 1 millisecond. Valid values are 100 microseconds to 5.484 milliseconds.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Unframed Data Modulation Enabled Property

Short Name: Unframed Data Modulation Enabled

Property of niWLANGeneration

Specifies whether to enable Unframed Data Modulation.

If you set this property to **True**, only the data portion is present in the frame, and the idle interval is coerced to 0.

False (0)	Specifies that unframed data modulation is disabled.
True (1)	Specifies that unframed data modulation is enabled.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:Auto Headroom Enabled Property

Short Name: Auto Headroom Enabled

Property of niWLANGeneration

Specifies whether the toolkit calculates the headroom or uses the value that you specify. For multiframe generation, the toolkit uses the headroom calculated on the first frame to scale the waveform for every frame. NI recommends that you do not set this property to **True** for multiframe generation because variation of the peak-to-average power ratio (PAPR) across frames may lead to excessive clipping. To avoid excessive clipping, set this property to **False** and use the default values for the Headroom property.

Note This property is supported only when you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **3.0.0**, **4.0.0**, or **5.0.0**.

The default value is **True**.

False (0)	Specifies that the toolkit uses the headroom that you specify in the Headroom property.
True (1)	Specifies that the toolkit calculates the headroom automatically.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Auto Headroom Enabled
Resettable	No

Hardware Settings:Headroom (dB) Property

Short Name: Headroom (dB)

Property of niWLANGeneration

Specifies the headroom per transmit channel. This value represents the maximum peak-to-average power ratio (PAPR) allowed in the generated signal. This value is expressed in dB.

The toolkit clips any portion of the signal that exceeds the peak power corresponding to this value. The toolkit ignores this property if you set the Auto Headroom Enabled property to **True**. If you set the Auto Headroom Enabled property to **False** and you do not specify the headroom property the toolkit uses default values based on the Standard property.

If you set the Standard property to **80211AX MIMOOFDM**, the headroom applied on the signal with respect to the average power computed is based on the Average Power Reference property.

Note If you specify a value that is more than the actual PAPR of the signal, there is loss of dynamic range of the digital-to-analog converter (DAC). If you specify a value that is less than the actual PAPR of the signal, the toolkit clips the generated signal.

Note In toolkit version 2.0.0, the Headroom property was called Max Expected PAPR. To get the same behavior of this property as in version 2.0.0, you must set the Auto Headroom Enabled property to **False** in version **3.0.0**, **4.0.0**, or **5.0.0**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"
80211AF MIMOOFDM, 80211AX MIMOOFDM	"[segment**y**/]channel**x**" ("segment0/" is optional if the Number of Segments property is set to 1)
80211AC MIMOOFDM	"[segment**y**/]channel**x**" ("[segment0/]" is optional if the Number of Segments property is set to 1)
	"channel**x**" when the Multi-segment Generation Mode property is set to **Single Generator**

If you set the Standard property to **80211B/G DSSS**, the default value is 5. Otherwise, the default value is 12.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niWLANG Set Headroom
Resettable	No

Hardware Settings:Fullscale Backoff (dB) Property

Short Name: Fullscale Backoff (dB)

Property of niWLANGeneration

Specifies the additional scaling factor applied to the waveform when you set the Auto Headroom Enabled property to **True**. This value is expressed in dB.

The toolkit ignores this property, if you set Auto Headroom Enabled property to **False**.

The default value is 2.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:Average Power Reference Property

Short Name: Average Power Reference

Property of niWLANGeneration

Specifies the portions of the packet used to compute the average power value to apply waveform power scaling and additive white Gaussian noise (AWGN). This property is valid when you set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **MU PPDU** or **Extended Range SU PPDU**.

The default value is **Non-boosted Fields**.

Non-boosted Fields (0)	Specifies that the toolkit uses the packet fields where no power boosting is applied.
Power Boosted Fields (1)	Specifies that the toolkit uses the packet fields where power boosting is applied.
Entire Packet (2)	Specifies that the toolkit uses the complete pac ket for computing the average power value.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:Maximum Hardware IQ Rate (S/s) Property

Short Name: Maximum Hardware IQ Rate (S/s)

Property of niWLANGeneration

Specifies the maximum I/Q rate that the NI vector signal generator supports.

This property will be set according to the device model in the niWLANG RFSG Create and Download Waveform VI.

Device	Valid Values (MS/s)
PXIe-5840/PXIe-5841/PXIe-5841 with PXIe-5655/ PXIe-5820/PXIe-5830/PXIe-5831	1250
PXIe-5646	250
PXIe-5644/PXIe-5645	120
PXIe-5673/5673E	200

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:RF Blanking Enabled Property

Short Name: RF Blanking Enabled

Property of niWLANGeneration

Specifies whether to enable RF blanking.

If you want to attenuate the RF OUT signal during the idle interval, set this property to **True**. This behavior prevents any DC leakage from the local oscillator of the signal generator from appearing at the RF OUT signal. RF blanking attenuates the RF OUT signal of signal generators quickly.

For more information about RF blanking, refer to the Burst Start Locations property, the Burst Stop Locations property, the niWLANG RFSG Create and Download Waveform VI, and the niWLANG RFSG Configure Script VI.

The default value is **False**.

<table>
<tr><td>False (0)</td><td>Specifies that the toolkit does not enable RF bla nking.</td></tr>
<tr><td>True (1)</td><td>Specifies that the toolkit enables RF blanking. If you select this option, the toolkit completes the following actions:
The toolkit returns the value of the RF B lanking Marker Positions property.
The toolkit queries the burst start locati ons and the burst stop locations to get ma rker positions that can be used to toggle t he state of RF blanking. These marker posi tions are generated such that RF blanking is ON during the idle interval.
The niWLANG RFSG Create and Downlo ad Waveform VI sets the niRFSG RF Blanki ng Source property to “marker0”, if it is no t set already.</td></tr>
</table>

	▪ The niWLANG RFSG Configure Script VI modifies the input from the **script** parameter to specify marker events.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:LO Sharing Enabled Property

Short Name: LO Sharing Enabled

Property of niWLANGeneration

Specifies whether to enable configuration for sharing of local oscillator (LO) signal for multiple NI RF vector signal analyzers or vector signal transceivers.

This property is read when you call the niWLANG RFSG Configure Multiple Device Synchronization VI.

The default value is **False**.

False (0)	Disables LO sharing.
True (1)	Enables LO sharing.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:LO Frequency Offset Mode Property

Short Name: LO Frequency Offset Mode

Property of niWLANGeneration

Specifies how the LO frequency offset is derived to configure frequency on the NI RF vector signal generators and the NI synthesizers in the niWLANG RFSG Configure Frequency VI. The toolkit ignores this property if you do not use PXIe-5840, PXIe-5841, PXIe-5841 with PXIe-5655, PXIe-5646, PXIe-5830, or PXIe-5831 devices.

The default value is **Auto**.

Auto (0)	Specifies that the LO frequency offset value is computed for optimal EVM performance.
User defined (1)	Sets the LO frequency offset value to the value you specified in the LO Frequency Offset property.
Disabled (2)	Specifies that the LO frequency offset is not set by the toolkit.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:LO Frequency Offset (Hz) Property

Short Name: LO Frequency Offset (Hz)

Property of niWLANGeneration

Specifies the LO frequency offset to be used when you set the LO Frequency Offset Mode property to **User Defined**.

The default value is 0 Hz.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Hardware Settings:Recommended Settings:IQ Rate (S/s) Property

Short Name: IQ Rate (S/s)

Property of niWLANGeneration

Returns the recommended sample rate for the current signal configuration. The **dt** parameter of the created waveform is the inverse of the recommended sample rate. If you set the **use waveform dt for IQ rate?** parameter of the niRFSG Write Arb Waveform VI to **False**, wire the IQ Rate property to the niRFSG IQ Rate property. This value is expressed in samples per second.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only
High-level VIs	niWLANG Get IQ Rate
Resettable	No

Hardware Settings:Recommended Settings:Actual Headroom (dB) Property

Short Name: Actual Headroom (dB)

Property of niWLANGeneration

Returns the actual headroom that the toolkit applies to the waveform. Use an active channel string to query this property for a transmit channel. This value is expressed in dB.

You must use the following active channel string formats to query this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"
80211AF MIMOOFDM, 80211AX MIMOOFDM	"[segment**y**/]channel**x**" ("segment0/" is optional if the Number of Segments property is set to 1)
80211AC MIMOOFDM	"[segment**y**/]channel**x**" ("[segment0/]" is optional if the Number of Segments property is set to 1)
	“channel**x**” when the Multi-segment Generation Mode property is set to **Single Generator**

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	niWLANG Get Actual Headroom
Resettable	No

Hardware Settings:Recommended Settings:RF Blanking Marker Positions [] Property

Short Name: RF Blanking Marker Positions []

Property of niWLANGeneration

Returns the array of sample positions of marker events, which are used to toggle the state of RF blanking, within the waveform.

The marker positions are such that RF blanking is enabled during the idle interval. This property is applicable only if you set the RF Blanking Enabled property to **True**. The RF Blanking Marker Positions property is queried by the niWLANG RFSG Create and Download Waveform VI to store RF blanking marker positions in the RFSG database.

Remarks

The following table lists the characteristics of this property.

Datatype	[I32]
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Hardware Settings:Recommended Settings:Burst Start Locations [] Property

Short Name: Burst Start Locations []

Property of niWLANGeneration

Returns the array of sample positions of start of the burst, within the waveform.

Remarks

The following table lists the characteristics of this property.

Datatype	[I32]
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Hardware Settings:Recommended Settings:Burst Stop Locations [] Property

Short Name: Burst Stop Locations []

Property of niWLANGeneration

Returns the array of sample positions of end of the burst, within the waveform.

Remarks

The following table lists the characteristics of this property.

Datatype	[I32]
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Hardware Settings:Recommended Settings:Signal Bandwidth (Hz) Property

Short Name: Signal Bandwidth (Hz)

Property of niWLANGeneration

Returns the Signal Bandwidth value that needs to be configured on the NI RFSG session for the PXIe-5820, PXIe-5830, PXIe-5831, PXIe-5841, or PXIe-5841 with PXIe-5655 devices.

The toolkit computes this value using the following equation.

Signal Bandwidth (Hz) = 2 * {(Channel Bandwidth/2) + |Maximum Carrier Frequency Offset|}

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Payload:A-MPDU Enabled Property

Short Name: A-MPDU Enabled

Property of niWLANGeneration

Specifies whether all medium access control (MAC) protocol data units (MPDUs) are transmitted as aggregate-MPDU (A-MPDU).

Note This property is applicable only if you set the Standard property to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM** and the Non-HT Modulation Mode property to **Off**.

The default value is **False**, if you set the Standard property to **80211N MIMOOFDM** or **80211AH MIMOOFDM**. The default value is **True**, if you set the Standard property to **80211AC MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**.

False (0)	The toolkit does not transmit an A-MPDU.
True (1)	The toolkit transmits an A-MPDU.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Payload AMPDU Enabled
Resettable	No

Payload:Auto Number of MPDUs Property

Short Name: Auto Payload Number of MPDUs

Property of niWLANGeneration

Specifies whether to compute the number of MPDUs in an AMPDU of 802.11ax Trigger-Based PPDU using the following trigger frame parameters; L-SIG Length Property, Pre-FEC Padding Factor Property, PE Disambiguity Property, and LDPC Extra Symbol Segment Property.

The default value is **False**.

False (0)	Specifies that the number of MPDUs is not auto computed.
True (1)	Specifies that the number of MPDUs is auto com puted.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Frame Type Property

Short Name: MAC Frame Type

Property of niWLANGeneration

Specifies the type of frame for the MPDU. This property can be set to Trigger Frame only if you set the Standard property to **80211A/G OFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, or **80211AX MIMOOFDM**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"", if you set the A-MPDU Enabled property to **False**
	"mpdux", if you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**.
	"mpdux", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**.
	"user**x**/mpduy", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**.
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**.
	"mpdux", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**.
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**.
	"mpdux", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**.
	"user**x**/mpduy", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**.

The default value is **General Frame**.

General Frame (0)	Specifies that the MAC frame type is General.
Data Frame (1)	Specifies that the MAC frame type is Data.
Trigger Frame (2)	Specifies that the MAC frame type is Trigger Fra me.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Number of MPDUs Property

Short Name: Payload Number of MPDUs

Property of niWLANGeneration

Specifies the number of medium access control (MAC) protocol data units (MPDUs) to combine into one aggregate-MPDU (A-MPDU).

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**	"" (empty string)
80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type prop erty to **SU PPDU**
	"user**x**", if you set the PPDU Type property to **M U PPDU**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type prop erty to **SU PPDU** or **Extended Range SU PPD U**

	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**
	Note The toolkit ignores this property if you set the A-MPDU Enabled property to **False** or if you set the Standard property to a value other than **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**; or if you set the Non-HT Modulation Mode property to **On**; or if you set the Auto Number of MPDUs property to **True**.

The default value is 1.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Payload Number of MPDUs
Resettable	No

Payload:Auto Data Length Mode Property

Short Name: Auto Payload Data Length Mode

Property of niWLANGeneration

Specifies whether to use the value specified in Data Length (bytes) property or automatically compute the data length of the MPDUs in an AMPDU.

The default value is **Disabled**.

Disabled (0)	Specifies that the toolkit uses the value specified in Data Length (bytes) property.

L-SIG Length (1)	Specifies that the toolkit computes the data length of the MPDUs in an AMPDU of 802.11ax Trigger-Based PPDU using the following trigger frame parameters; L-SIG Length, Pre-FEC Padding Factor, PE Disambiguity, and LDPC Extra Symbol Segment properties.
Frame Duration (2)	Specifies that the toolkit computes the data length of the MPDUs in an AMPDU using the value specified in Frame Duration (s) property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Data Length (bytes) Property

Short Name: Payload Data Length (bytes)

Property of niWLANGeneration

Specifies the length of the payload, excluding the lengths of the medium access control (MAC) header and frame check sequence (FCS). The payload length encoded into the physical layer (PHY) header is the sum of the payload data length and the lengths of the MAC header and FCS. This value is expressed in bytes.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, **80211AH MIMOOFDM**	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled property to **True**

80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**
	"user**x**", if you set the A-MPDU Enabled property to **False** and use the niWLANG Create Trigger Frame MSDU VI to generate the trigger frame

The default values are as follows:

- If you set the Standard property to **80211N MIMOOFDM** and the A-MPDU Enabled property to **True**, the default value is 1,024, the limits on MPDU length are 0 to 4,095 (inclusive), and the limits on PSDU length are 0 to 65,535 (inclusive).

- If you set the Standard property to **80211N MIMOOFDM** and the A-MPDU Enabled property to **False**, the default value is 4,096 and the limits on PSDU length are 0 to 65,535 (inclusive).
- If you set the Standard property to **80211AC MIMOOFDM** and the A-MPDU Enabled property to **True**, the default value is 4,096, the limits on MPDU length are 0 to 16,383 (inclusive), and the limits on PSDU length are 0 to 46,92,480 (inclusive).
- If you set the Standard property to **80211AC MIMOOFDM** and the A-MPDU Enabled property to **False**, the default value is 4,096, and the limits on PSDU length are 0 to 4692480 (inclusive).
- If you set the Standard property to **80211AH MIMOOFDM** and the A-MPDU Enabled property to **True**, the default value is 256, the limits on MPDU length are 0 to 16,383 (inclusive), and the limits on PSDU length are 0 to 7,97,159 (inclusive).
- If you set the Standard property to **80211AH MIMOOFDM** and the A-MPDU Enabled property to **False**, the default value is 256, and the limits on PSDU length are 0 to 511 (inclusive).
- If you set the Standard property to **80211AF MIMOOFDM** and the A-MPDU Enabled property to **True**, the default value is 4096, the limits on MPDU length are 0 to 16,383 (inclusive), and the limits on PSDU length are 0 to 1,065,600 (inclusive).
- If you set the Standard property to **80211AF MIMOOFDM** and the A-MPDU Enabled property to **False**, the default value is 4096, and the limits on PSDU length are 0 to 1,065,600 (inclusive).
- If you set the Standard property to **80211AX MIMOOFDM** and the A-MPDU Enabled property to **True**, the default value is 4096, the limits on MPDU length are 0 to 16,383 (inclusive), and the limits on PSDU length are 0 to 65,00,631 (inclusive).
- If you set the Standard property to **80211AX MIMOOFDM** and the A-MPDU Enabled property to **False**, the default value is 4096, and the limits on PSDU length are 0 to 65,00,631 (inclusive).

If you set the Standard property to any other value, the default value is 1,024, and valid values are 0 to 4,095 (inclusive).

The toolkit ignores this property, if you set the Auto Data Length property to **L–SIG Length or Frame Duration**.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Data Type Property

Short Name: Payload Data Type

Property of niWLANGeneration

Specifies the type of payload for waveform generation.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **PN Sequence**.

User Defined Pattern (0)	Specifies that the toolkit uses the sequence that you specify using the Payload User Defined Bits property.
PN Sequence (1)	Specifies that the toolkit generates the bits using a pseudonoise (PN) sequence with the PN seed and order you specify using the Payload PN Seed and Payload PN Order properties, respectively.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Payload:PN Order Property

Short Name: Payload PN Order

Property of niWLANGeneration

Specifies the order (length of memory) of the pseudorandom bit sequence (PRBS) generator.

The generated sequence is repeated (2) - 1 bits. If you set the Payload Data Type property to **User Defined Pattern**, the toolkit ignores the PN Order property.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 9. Valid values are 5 to 31, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:PN Seed Property

Short Name: Payload PN Seed

Property of niWLANGeneration

Specifies the initialization seed used for the pseudorandom bit sequence (PRBS) generator. If you set the Payload Data Type property to **User Defined Pattern**, the toolkit ignores the Payload PN Seed property.

If you set the Number of Frames property to a value greater than 1, and set the **reset?** parameter of the niWLANG Create Waveform VI to FALSE, the toolkit uses the PRBS generator state at the end of the payload in frame **n** as the seed for frame **n** + 1. If you set the **reset?** parameter of the niWLANG Create Waveform VI to TRUE, all frames use the value of the Payload PN Seed property.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format

80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0xD6BF7DF2.

Remarks

The following table lists the characteristics of this property.

Datatype	I32

Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:User Defined Bits [] Property

Short Name: Payload User Defined Bits []

Property of niWLANGeneration

Specifies a user-defined bit pattern as an array of zeros and ones. If the array length is greater than the configured payload data length, the toolkit uses a subset of the required length from the beginning of the array for waveform generation. If the array length is less than the configured payload length, the toolkit repeats the user-defined bit pattern until the required length is achieved. If you set the Payload Data Type property to **PN Sequence**, the toolkit ignores the Payload User Defined Bits property.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, **80211AH MIMOOFDM**	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**

	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default is an empty array. Valid values include arrays of zeros and ones.

Remarks

The following table lists the characteristics of this property.

Datatype	[I32]
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Frame Format Property

Short Name: MAC Frame Format

Property of niWLANGeneration

Specifies whether the medium access control (MAC) frame is long or short.

Note Configure this property only when you set the Standard property to **80211AH MIMOOFDM**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format

80211AH MIMOOFDM	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled proper ty to **True**

The default value is **Long**.

Long (0)	Specifies that the MAC frame format is long. The long format follows the general frame structure as defined in section 8.2.3 of **IEEE Standard 802 .11-2012**.
Short (1)	Specifies that the MAC frame format is short. Th e short format follows the short frame structure as defined in section 8.8 of **IEEE Standard P802 .11ah/D1.3**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Enabled Property

Short Name: MAC Header Enabled

Property of niWLANGeneration

Specifies whether to enable the medium access control (MAC) header, as defined in section 7.1.2 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3** and section 9.2.4 of **IEEE P802.11ax/D6.0**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format

80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

False (0)	Disables the MAC FCS field.
True (1)	Enables the MAC FCS field.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Frame Control Property

Short Name: MAC Header Frame Control

Property of niWLANGeneration

Specifies the two-byte frame control field of the medium access control (MAC) header as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) at the rightmost position.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**

80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default values are as follows:

- If you set the MAC Frame Type Property to General frame, the default value is 0x0.
- If you set the MAC Frame Type Property to Data frame, the default value is 0x8.
- If you set the MAC Frame Type Property to Trigger frame, the default value is 0x24.

Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Duration/ID Property

Short Name: MAC Header Duration/ID

Property of niWLANGeneration

Specifies the two-byte duration ID field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) in the rightmost position.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, 80211AH MIMOOFDM	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

Note The toolkit ignores this property if you set the MAC Frame Format property to **Short**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address1 Enabled Property

Short Name: MAC Header Address1 Enabled

Property of niWLANGeneration

Specifies whether to enable the Address1 field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

False (0)	Disables the Address1 field of the MAC header.
True (1)	Enables the Address1 field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address1 Length (bytes) Property

Short Name: Address1 Length (bytes)

Property of niWLANGeneration

Specifies the length of Address1 field when you set the MAC Frame Format property to **Short**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AH MIMOOFDM	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled proper ty to **True**

The default value is 2. Valid values are 2 and 6.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address1 Property

Short Name: MAC Header Address1

Property of niWLANGeneration

Specifies the Address1 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. The length of this field is defined by the Address1 Length property if you set the MAC Frame Format property to **Short**. In all other instances, the length of this field is six bytes. If this field is a MAC address, it is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position.

For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0 x 123456789ABC.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address2 Enabled Property

Short Name: MAC Header Address2 Enabled

Property of niWLANGeneration

Specifies whether to enable the Address2 field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

False (0)	Disables the Address2 field of the MAC header.
True (1)	Enables the Address2 field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address2 Length (bytes) Property

Short Name: Address2 Length (bytes)

Property of niWLANGeneration

Specifies the length of Address2 field when the MAC Frame Format is **Short**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AH MIMOOFDM	"" (empty string), if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the A-MPDU Enabled property to **True**

The default value is 2. Valid values are 2 and 6.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address2 Property

Short Name: MAC Header Address2

Property of niWLANGeneration

Specifies the Address2 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. The length of this field is defined by the Address2 Length property if you set the MAC Frame Format property to **Short**. In all other instances, the length of the field is six bytes. If this field is a MAC address, it is represented by the least significant byte in the leftmost position, and each byte is represented by the least significant bit in the rightmost position.

For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0x123456789ABC.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address3 Enabled Property

Short Name: MAC Header Address3 Enabled

Property of niWLANGeneration

Specifies whether to enable the Address3 field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Disables the Address3 field of the MAC header.
True (1)	Enables the Address3 field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address3 Property

Short Name: MAC Header Address3

Property of niWLANGeneration

Specifies the six-byte Address3 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position.

For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0x123456789ABC.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**

	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Sequence Control Enabled Property

Short Name: MAC Header Sequence Control Enabled

Property of niWLANGeneration

Specifies whether to enable the Sequence Control field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Disables the Sequence Control field of the MAC header.
True (1)	Enables the Sequence Control of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Sequence Control Property

Short Name: MAC Header Sequence Control

Property of niWLANGeneration

Specifies the two-byte Sequence Control field as defined in section 7.1.3 of **IEEE Standard 802.11-2007**, **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**

80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Payload:MAC Header:Address4 Enabled Property

Short Name: MAC Header Address4 Enabled

Property of niWLANGeneration

Specifies whether to enable the Address4 field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**

	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Disables the Address4 field of the MAC header.
True (1)	Enables the Address4 field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:Address4 Property

Short Name: MAC Header Address4

Property of niWLANGeneration

Specifies the six-byte Address4 field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position.

For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0x123456789ABC.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:QoS Control Enabled Property

Short Name: MAC Header QoS Control Enabled

Property of niWLANGeneration

Specifies whether to enable the QoS Control field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

Note The toolkit ignores this property if you set the MAC Frame Format property to **Short**.

The default value is **False**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Disables the QoS Control field of the MAC header.
True (1)	Enables the QoS Control field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	N/A
Resettable	No

Payload:MAC Header:QoS Control Property

Short Name: MAC Header QoS Control

Property of niWLANGeneration

Specifies the two-byte QoS Control field as defined in section 7.1.3 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**

80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

Note The toolkit ignores this property if you set the MAC Frame Format property to **Short**.

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:HT Control Enabled Property

Short Name: MAC Header HT Control Enabled

Property of niWLANGeneration

Specifies whether to enable the HT Control field of the medium access control (MAC) header.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

Note The toolkit ignores this property if you set the MAC Frame Format property to **Short** or if you set the MAC Frame Type property to **Trigger Frame**.

The default value is **False**.

False (0)	Disables the HT Control field of the MAC header.
True (1)	Enables the HT Control field of the MAC header.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MAC Header:HT Control Property

Short Name: MAC Header HT Control

Property of niWLANGeneration

Specifies the four-byte HT Control field as defined in section 7.1.3 of **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013** and **IEEE P802.11ah/D1.3**, and section 9.2.4 of **IEEE P802.11ax/D6.0**. This field is represented with the least significant bit in the rightmost position.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**

	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

Note The toolkit ignores this property if you set the MAC Frame Format property to **Short**.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFF. For values outside this range, the toolkit uses the least significant four bytes.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Sequence Number Increment Enabled

Payload:MAC Header:Sequence Control:Sequence Number Increment Enabled Property

Short Name: MAC Header Seq Num Increment Enabled

Property of niWLANGeneration

Specifies whether to increment the sequence number in a sequence of frames.

The default value is **False**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Does not increment the sequence number.
True (1)	Increments the sequence number.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Sequence Number Increment Interval (frames)

Payload:MAC Header:Sequence Control:Sequence Number Increment Interval (frames) Property

Short Name: MAC Header Seq Num Increment Interval

Property of niWLANGeneration

Specifies the number of frames after which the sequence number is incremented by 1. The starting number is the value represented by the sequence number sub-field of the Sequence Control field. The sequence number is wrapped to 0 after reaching the value 4,095 or (2 -1).

The default value is 1. Valid values are 0 to 4,095 (inclusive).

The toolkit ignores this property if you set the MAC Header Seq Num Increment Enabled property to **False** and the MAC Frame Type property to **Trigger Frame**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Fragment Number Increment Enabled

Payload:MAC Header:Sequence Control:Fragment Number Increment Enabled Property

Short Name: MAC Header Frag Num Increment Enabled

Property of niWLANGeneration

Specifies whether to increment the fragment number in a sequence of frames. The starting number is the value represented by the fragment number sub-field of the MAC Header Sequence Control property. If you set the MAC Header Frag Num Increment Enabled property to **True**, the toolkit increments the fragment number by 1 for every successive frame having the same sequence number. The fragment number wraps to the starting number when the sequence number increments. The fragment number wraps to 0 after reaching the value 15.

The default value is **False**.

The toolkit ignores this property if you set the MAC Frame Type property to **Trigger Frame**.

False (0)	Does not increment the fragment number.
True (1)	Increments the fragment number.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:MPDU Length (bytes) Property

Short Name: MPDU Length (bytes)

Property of niWLANGeneration

Returns the length of the medium access control (MAC) protocol data unit (MPDU). An MPDU comprises of a MAC header, a frame body, and a frame check sequence (FCS). The MPDU Length property is the sum of the length of MAC header, the value of the Payload Data Length property, and the length of FCS, which is equal to four bytes. If you disable the MAC Header Enabled and MAC FCS Enabled properties, the lengths of MAC header and FCS are zero. This value is expressed in bytes.

You must use the following active channel string formats to query this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**
80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Payload:MAC FCS Enabled Property

Short Name: MAC FCS Enabled

Property of niWLANGeneration

Specifies whether to enable the medium access control (MAC) frame check sequence (FCS), as defined in section 7.1.2 of **IEEE Standard 802.11-2007** and **IEEE Standard 802.11n-2009**, section 8.2.4 of **IEEE Standard 802.11ac-2013**, **IEEE P802.11ah/D1.3** and **IEEE Standard P802.11af-2013**, and section 9.2.4 of **IEEE P802.11ax/D6.0**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdu**x**", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** and the A-MPDU Enabled property to **True**

80211AF MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **True**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdu**x**", if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU** and the A-MPDU Enabled property to **True**
	"user**x**/mpdu**y**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** and the A-MPDU Enabled property to **True**

The default value is **True**.

False (0)	Disables the MAC FCS field.
True (1)	Enables the MAC FCS field.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Trigger Frame:AID12 Property

Short Name: AID12

Property of niWLANGeneration

Specifies the value of the AID12 field in the trigger frame.

The valid values are 1 to 2007, which are used for indication of RUs used by Trigger-based PPDU.

The values 0 and 2045 are used for Random Access RU information specification, the value 2046 is used for unassigned RU location indication, and the value 4095 is reserved for trigger frame padding indication.

You must use the following active channel string format to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"userx", if you set the PPDU Type property to **Trigger-Based PPDU**.

The default value is 1.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Trigger Frame:AP Tx Power Property

Short Name: AP Tx Power

Property of niWLANGeneration

Specifies the value of the AP Tx Power field of the trigger frame. The power values -20 dBm to 40 dBm are mapped to the field values 0 to 60, respectively.

The default value is 0. The valid values are 0 to 63, both inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	N/A
Resettable	No

Payload:Trigger Frame:CS Required Property

Short Name: CS Required

Property of niWLANGeneration

Specifies the CS required sub-field in the 802.11ax Trigger Frame.

The default value is 0. The valid values are 0 and 1.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Trigger Frame:Target RSSI Property

Short Name: Target RSSI

Property of niWLANGeneration

Specifies the value of the UL-Target RSSI field of the trigger frame. The power values -110 dBm to -20 dBm are mapped to the field values 0 to 90, respectively. To specify the maximum transmit power for the assigned MCS you must set the value of 127.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"userx", if you set the PPDU Type property to **Trigger-Based PPDU**

The default value is 78, which is corresponding power value -32 dBm. The valid values are 0 to 127, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Trigger Frame:MAC Padding Duration (s) Property

Short Name: MAC Padding Duration (s)

Property of niWLANGeneration

Specifies the padding duration when the MAC Frame Type property is set to **Trigger Frame**. This property is valid if you set the Standard property to **80211A/G OFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, or **80211AX MIMOOFDM**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM	"" (empty string)
80211N MIMOOFDM	"", if you set the A-MPDU Enabled property to **False**
	"mpdux", If you set the A-MPDU Enabled property to **True**
80211AC MIMOOFDM, 80211AX MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU** and the A-MPDU Enabled property to **False**
	"mpdux", If you set the A-MPDU Enabled property to **True**

The default value is **0us**.

0us (0)	Specifies that there is no padding.
8us (1)	Specifies that the padding duration is 8us.

16us (2)	Specifies that the padding duration is 16us.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Payload:Number of Data Symbols Property

Short Name: Number of Data Symbols

Property of niWLANGeneration

Returns the number of symbols in the data portion of the WLAN frame. The symbol refers to the chip if the Standard property is set to **80211B/G DSSS**, and the symbol refers to the OFDM symbol for other values of the Standard property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Coding:Subcarrier Mask Property

Short Name: Subcarrier Mask

Property of niWLANGeneration

Specifies the sequence of attenuation values on each subcarrier in the signal and payload symbols if you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**. You must specify a 64-element array. The first element of the array corresponds to subcarrier index-32, and the 64th element corresponds to subcarrier index 31, as defined in section 17.3.2.5 of **IEEE Standard 802.11a-1999**.

If an element has a value of 0, this indicates that the corresponding subcarrier is absent in the generated signal. If an element has a value of 1, this indicates that the corresponding subcarrier is present in the generated signal.

The default value is a 64-element array of all ones. Valid values are 64-element arrays of zeros and ones.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Locked Clocks Bit Enabled Property

Short Name: Locked Clocks Bit Enabled

Property of niWLANGeneration

Specifies whether to enable the Locked Clocks Bit flag for the direct sequence spread spectrum (DSSS) header, as defined in sections 18.2.3.4 and 18.2.3.11 of **IEEE Standard 802.11b-1999** and section 19.3.2.1 of **IEEE Standard 802.11g-2003**.

Note Configure this property only when you set the Standard property to **80211B/G** or **80211G DSSSOFDM**.

The default value is **True**.

False (0)	Sets the locked clocks bit to 0.
True (1)	Sets the locked clocks bit to 1.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Header Encoder Enabled Property

Short Name: Header Encoder Enabled

Property of niWLANGeneration

Specifies whether to enable convolutional encoding of the OFDM SIGNAL field, as defined in section 17.3.5.5 of **IEEE Standard 802.11a-1999**.

Note Configure this property only when you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**.

The default value is **True**.

False (0)	Disables convolutional encoding and repeats ea ch input bit twice.
True (1)	Enables convolutional encoding.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Header Interleaver Enabled Property

Short Name: Header Interleaver Enabled

Property of niWLANGeneration

Specifies whether to enable interleaving for the OFDM SIGNAL field, as defined in section 17.3.5.6 of **IEEE Standard 802.11a-1999**.

Note Configure this property only when you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**.

The default value is **True**.

False (0)	Disables header interleaving.
True (1)	Enables header interleaving.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Payload Scrambler Enabled Property

Short Name: Payload Scrambler Enabled

Property of niWLANGeneration

Specifies whether to enable scrambling of the payload for OFDM packets and the entire burst for direct sequence spread spectrum (DSSS) packets, as defined in section 17.3.5.4 of **IEEE Standard 802.11a-1999** and section 18.2.4 of **IEEE Standard 802.11b-1999**.

Note If you set the Standard property to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**, the toolkit ignores the Payload Scrambler Enabled property and always enables the payload scrambler.

The default value is **True**.

False (0)	Disables scrambling of the payload.
True (1)	Enables scrambling of the payload.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Payload Scrambler Seed Property

Short Name: Payload Scrambler Seed

Property of niWLANGeneration

Specifies the initial state of the scrambler seed.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, **80211G DSSSOFDM**, **80211N MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**	"" (empty string)

80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**

For direct sequence spread spectrum (DSSS) packets, the default value follows the requirements defined in sections 18.2.3.1 and 18.2.3.8 of **IEEE Standard 802.11b-1999**. For OFDM and DSSS-OFDM packets, the default value is 93.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Payload Encoder Enabled Property

Short Name: Payload Encoder Enabled

Property of niWLANGeneration

Specifies whether to enable convolutional encoding of the OFDM payload, as defined in section 17.3.5.5 of **IEEE Standard 802.11a-1999**.

Note Configure this property only when you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**.

The default value is **True**.

False (0)	Disables payload encoding.
True (1)	Enables payload encoding.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:FEC Coding Type Property

Short Name: FEC Coding Type

Property of niWLANGeneration

Specifies the type of forward error correction (FEC) coding to use if you set the Standard property to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AF MIMOOFDM** or **80211AX MIMOOFDM**, as defined in section 20.3.11.3 of **IEEE Standard 802.11n-2009**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, **80211AF MIMOOFDM**	""(empty string)
80211AC MIMOOFDM	"", if you set the PPDU Type property to **SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**

The default value is **LDPC**, if you set the Standard property to **80211AX MIMOOFDM**, or **BCC** otherwise.

BCC (0)	Specifies that the FEC coding type is binary con volutional code (BCC).
LDPC (1)	Specifies that the FEC coding type is low-densit y parity check (LDPC).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Coding:Payload Interleaver Enabled Property

Short Name: Payload Interleaver Enabled

Property of niWLANGeneration

Specifies whether to enable interleaving for the OFDM payload, as defined in section 17.3.5.6 of **IEEE Standard 802.11a-1999**.

Note Configure this property only when you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**.

The default value is **True**.

False (0)	Disables interleaving.
True (1)	Enables interleaving.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:PPDU Type Property

Short Name: PPDU Type

Property of niWLANGeneration

Specifies the type of physical layer convergence procedure (PLCP) protocol data unit (PPDU), if the Standard property is set to **80211AC MIMOOFDM** or **80211AX MIMOOFDM**.

The default value is **SU PPDU**.

SU PPDU (0)	Specifies that the toolkit generates a single user (SU) PPDU.
MU PPDU (1)	Specifies that the toolkit generates a multi-user (MU) PPDU.
Extended Range SU PPDU (2)	Specifies that the toolkit generates an extended range single user (ER SU) PPDU.
Trigger-Based PPDU (3)	Specifies that the toolkit generates a trigger-based (TB) PPDU.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	niWLANG Set PPDU Type
Resettable	No

Data Rate and Frame Format:RU Allocation Settings:Mode Property

Short Name: RU Allocation Mode

Property of niWLANGeneration

Specifies how to configure the multi-user allocation in a 802.11ax MU PPDU signal.

The default value is **Individual**.

Individual (0)	Users are configured using the RU Size and the RU Offset properties.
Group (1)	Users are configured as per the common field of the HE-SIG-B using the RU Allocation property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:RU Allocation Settings:RU Size Property

Short Name: RU Size

Property of niWLANGeneration

Specifies the size of resource unit (RU) in terms of the number of subcarriers for the 802.11ax signal. You must configure this property when you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**.

When you set the PPDU Type property to **Extended Range SU PPDU**, this property only supports **242** and **106** as valid values.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**.
	"" (empty string), if you set the PPDU Type property to **Extended Range SU PPDU**

The default value is **26**.

26 (26)	Specifies that the RU size is 26 subcarriers.
52 (52)	Specifies that the RU size is 52 subcarriers.
106 (106)	Specifies that the RU size is 106 subcarriers.
242 (242)	Specifies that the RU size is 242 subcarriers.
484 (484)	Specifies that the RU size is 484 subcarriers.
996 (996)	Specifies that the RU size is 996 subcarriers.
2x996 (1992)	Specifies that the RU size is 1992 subcarriers.

Remarks

The following table lists the characteristics of this property.

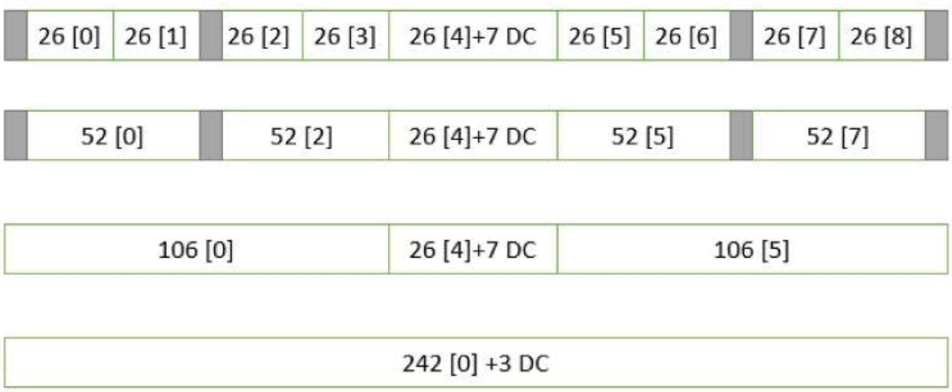
Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:RU Allocation Settings:RU Offset Property

Short Name: RU Offset

Property of niWLANGeneration

Specifies the location of the resource unit (RU), in terms of the index of 26-tone RU, assuming the entire bandwidth is composed of 26-tone RUs in the 802.11ax signal. For example, in the following figure, to specify the 106-tone RU second from left, you must configure this property to 5.



In the preceding image,

- The shaded area indicates null subcarriers
- The square bracket value indicates the RU offset

This property is valid only if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**.

The valid values are as follows:

Channel Bandwidth (MHz)	Number of Segments	Valid Values (all inclusive)
20	1	0 to 8
40	1	0 to 17
80	1	0 to 36
160	1	0 to 73
80	2	0 to 73

The default value is 0.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:RU Allocation Settings:RU Allocation Property

Short Name: RU Allocation

Property of niWLANGeneration

Specifies the common field of the HE-SIG-B to be used for multi-user allocation in a 802.11ax MU PPDU signal.

Each element of the array represents the RU allocation subfield of the common field corresponding to each 20 MHz sub-channel in the channel bandwidth.

The RU Size property and the RU Offset property of the configured users are derived based on the value of this property. Query the number of users using the niWLANG Get Number of Users from RU Allocation VI to configure the user related properties. The users are indexed in the increasing order of frequency.

When the Channel Bandwidth property is set to 80 MHz or 160 MHz, the 26 Resource Units (RU) at the center are enabled by default. These users can be disabled by using the User Enabled property.

The toolkit ignores this property if you set the RU Allocation Mode property to **Individual**.

The default value in the array is 0d128.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:RU Allocation Settings:User Enabled Property

Short Name: User Enabled

Property of niWLANGeneration

Specifies whether to enable the user in a 802.11ax MU PPDU signal.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **MU PPDU**.

The default value is **True**.

False (0)	User is disabled.
True (1)	User is enabled.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Power Boost Factor Property

Short Name: Power Boost Factor

Property of niWLANGeneration

Specifies the factor, per resource unit (RU), by which the amplitude of the HE modulated fields in 802.11ax signals are scaled, when you set the PPDU Type property to **MU PPDU**. The value of this property must be the same across all users within the RU. An RU is defined by the RU Size and RU Offset properties.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **MU PPDU**.

The default value is 1. The valid values are 0.1 to 10, inclusive. To convert this value to dB, use the following formula:

Power Boost Factor (dB) = 20 * log (Power Boost Factor)

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Data Rate and Frame Format:Relative Power (dB) Property

Short Name: Relative Power (dB)

Property of niWLANGeneration

Specifies the per user power scaling value of the 802.11ax signal when you set the PPDU Type property to **Trigger-Based PPDU**. This value is expressed in dB. The power scaling value is with reference to the user with index 0. The toolkit ignores the property value, if it is specified for the user with index 0.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **Trigger-Based PPDU**.

The default value is 0. The valid values are -100 to 100, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Non-HT Modulation Mode Property

Short Name: Non-HT Modulation Mode

Property of niWLANGeneration

Specifies whether the format of the incoming OFDM signal is non-high throughput (HT). This property is valid only if the Standard property is set to **80211N MIMOOFDM**, **80211AC MIMOOFDM**, or **80211AF MIMOOFDM**.

A non-HT format signal has only L-LTF, L-STF, and L-SIG symbols in the preamble, which are similar to 802.11n, 802.11ac, or 802.11af signals with same bandwidth. The payload is modulated in the same manner as an 802.11a signal with a bandwidth of 20 MHz. The payload is repeated with appropriate tone rotation to fill the channel bandwidth.

The default value is **Off**.

Off (0)	Specifies that the signal is not of non-HT format .
On (1)	Specifies that the signal is of non-HT format.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:OFDM Data Rate (Mbps) Property

Short Name: OFDM Data Rate (Mbps)

Property of niWLANGeneration

Specifies the data rate for the OFDM payload, as defined in section 17.3.2.2 of **IEEE Standard 802.11-2007**. This value is expressed in Mbps.

Note Configure this property only when you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or

	80211G DSSSOFDM, or if you set the Non-HT Modulation Mode property to **On**.

The default value is **6**.

6 (0)	Specifies a data rate of 1.5 Mbps, 3 Mbps, and 6 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
9 (1)	Specifies a data rate of 2.25 Mbps, 4.5 Mbps, and 9 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
12 (2)	Specifies a data rate of 3 Mbps, 6 Mbps, and 12 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
18 (3)	Specifies a data rate of 4.5 Mbps, 9 Mbps, and 18 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
24 (4)	Specifies a data rate of 6 Mbps, 12 Mbps, and 24 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
36 (5)	Specifies a data rate of 9 Mbps, 18 Mbps, and 36 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
48 (6)	Specifies a data rate of 12 Mbps, 24 Mbps, and 48 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
54 (7)	Specifies a data rate of 13.5 Mbps, 27 Mbps, and 54 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.

Remarks

The following table lists the characteristics of this property.

Datatype	I32

Permissions	Read/Write
High-level VIs	niWLANG Set OFDM Data Rate
Resettable	No

Data Rate and Frame Format:Actual OFDM Data Rate (Mbps) Property

Short Name: Actual OFDM Data Rate (Mbps)

Property of niWLANGeneration

Returns the OFDM data rate depending upon the values of the Channel Bandwidth and OFDM Data Rate properties. This query is only successful if you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM** or **80211G DSSSOFDM**, or if you set the Non-HT Modulation Mode property to **On**. For more information about the OFDM data rate, refer to section 17.2.3.3 of **IEEE Standard 802.11-2007**. This value is expressed in Mbps.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:DSSS Data Rate (Mbps) Property

Short Name: DSSS Data Rate (Mbps)

Property of niWLANGeneration

Specifies the data rate for the direct sequence spread spectrum (DSSS) payload, as defined in **IEEE Standard 802.11b-1999** and the extended rate physical layer-packet binary convolutional coding (ERP-PBCC) mode in **IEEE Standard 802.11g-2003**. This value is expressed in Mbps.

Note Configure this property only if you set the Standard property to 80211B/G DSSS.

The default value is **1**.

1 (0)	Specifies a data rate of 1 Mbps, as defined in sections 18.4.6.3 and 18.4.6.4 of **IEEE Standard 802.11b-1999**.
2 (1)	Specifies a data rate of 2 Mbps, as defined in sections 18.4.6.3 and 18.4.6.4 of **IEEE Standard 802.11b-1999**.
5.5 CCK (2)	Specifies a data rate of 5.5 Mbps complementary code keying (CCK), as defined in section 18.4.6.5 of **IEEE Standard 802.11b-1999**.
5.5 PBCC (3)	Specifies a data rate of 5.5 Mbps PBCC, as defined in section 18.4.6.6 of **IEEE Standard 802.11b-1999**.
11 CCK (4)	Specifies a data rate of 11 Mbps CCK, as defined in section 18.4.6.5 of **IEEE Standard 802.11b-1999**.
11 PBCC (5)	Specifies a data rate of 11 Mbps PBCC, as defined in section 18.4.6.6 of **IEEE Standard 802.11b-1999**.
22 (6)	Specifies a data rate of 22 Mbps, as defined in section 19.3.3.2 of **IEEE Standard 802.11g-2003**.
33 (7)	Specifies a data rate of 33 Mbps, as defined in section 19.3.3.2 of **IEEE Standard 802.11g-2003**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set DSSS Data Rate
Resettable	No

Data Rate and Frame Format:DSSS Preamble Type Property

Short Name: DSSS Preamble Type

Property of niWLANGeneration

Specifies whether to use a long or short preamble for direct sequence spread spectrum (DSSS) and DSSS-OFDM packets, as defined in **IEEE Standard 802.11b-1999**.

Note Configure this property only if you set the Standard property to **80211B/G DSSS** or **80211G DSSSOFDM**.

The default value is **Long Preamble**.

Short Preamble (0)	Uses a short preamble as defined in section 18.2.2.2 of **IEEE Standard 802.11b-1999**.
Long Preamble (1)	Uses a long preamble as defined in section 18.2.2.1 of **IEEE Standard 802.11b-1999**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set DSSS Preamble Type
Resettable	No

Data Rate and Frame Format:STA-ID Property

Short Name: STA-ID

Property of niWLANGeneration

Specifies 11 LSBs of the association identifier (AID) in a 802.11ax signal when you set the PPDU Type property to **MU PPDU**.

When the PPDU Type property is set to **MU PPDU**, the valid values are 0 to 2047.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **MU PPDU**.

The default value is 0.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:MCS Index Property

Short Name: MCS Index

Property of niWLANGeneration

Specifies the value of the modulation and coding scheme (MCS) index. The MCS index is a compact representation that determines the modulation scheme, coding rate, and number of spatial streams, as specified in section 20.3.5 of **IEEE Standard 802.11n-2009**, section 22.5 of **IEEE Standard 802.11ac-2013**, section 24.5 of **IEEE**

P802.11ah/D1.3, section 23.5 of **IEEE Standard 802.11af-2013**, and section 27.5 of **IEEE P802.11ax/D6.0**.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211N MIMOOFDM, 80211AH MIMOOFDM , 80211AF MIMOOFDM	"" (empty string)
80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**

The default value is 0. For **80211AC MIMOOFDM**, valid values are 0 to 11, inclusive. For **80211AX MIMOOFDM**, valid values are 0 to 13, inclusive. For **80211N MIMOOFDM**, valid values are 0 to 32, inclusive. For **80211AH MIMOOFDM**, valid values are 0 to 10, inclusive. For **80211AF MIMOOFDM**, valid values are 0 to 9, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set MCS Index
Resettable	No

Data Rate and Frame Format:Dual Carrier Modulation Enabled Property

Short Name: DCM Enabled

Property of niWLANGeneration

Specifies whether the dual carrier modulation (DCM) is applied to the data part of the 802.11ax signals or not. The property can be set to **True** only when the MCS index is 0, 1, 3 or 4, and the number of spatial streams is 1 or 2.

You must use the following active channel string formats to configure this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**

The default value is **False**.

False (0)	Disables the DCM.
True (1)	Enables the DCM.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:HE-SIG-B MCS Index Property

Short Name: HE-SIG-B MCS Index

Property of niWLANGeneration

Specifies the value of the modulation and coding scheme (MCS) index of the HE-SIG-B field of 802.11ax signal when you set the PPDU Type property to **MU PPDU**.

The default value is 0. The valid values are 0 to 5, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:HE-SIG-B Dual Carrier Modulation Enabled Property

Short Name: HE-SIG-B DCM Enabled

Property of niWLANGeneration

Specifies whether the dual carrier modulation (DCM) is applied on the HE-SIG-B field of 802.11ax signals or not. The property can be set to **True** only when HE-SIG-B MCS index is 0, 1, 3 or 4.

The default value is **False**.

False (0)	Dual carrier modulation is not applied on the HE-SIG-B field.

True (1)	Dual carrier modulation is applied on the HE-SIG-B field.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:80211n PLCP Frame Format Property

Short Name: 80211n PLCP Frame Format

Property of niWLANGeneration

Specifies the format of the physical layer convergence protocol (PLCP) frame structure. The frame structure determines the arrangement of preambles, header (SIGNAL field), and payload in a frame, as defined in section 20.3.2 of **IEEE Standard 802.11n-2009**.

The default value is **Mixed Format**.

Mixed Format (0)	Specifies that the PLCP frame structure consists of non-high throughput (HT) preamble and header (SIGNAL field) followed by HT header, preambles, and payload as specified in **IEEE Standard 802.11n-2009**.
Greenfield Format (1)	Specifies that the PLCP frame structure does not support non-HT and that it starts with HT preamble, followed by HT SIGNAL field and payload.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set PLCP Frame Format (802 11N)
Resettable	No

Data Rate and Frame Format:80211ah Preamble Type Property

Short Name: 80211ah Preamble Type

Property of niWLANGeneration

Specifies the preamble type of packet if you set the Standard property to **80211AH MIMOOFDM**. For more information about 80211ah Preamble Type, refer to section 24.3.8.2 of **IEEE Standard P802.11ah/D1.3**.

The default value is **Short Preamble**.

Short Preamble (0)	Specifies that the preamble type is S1G_Short.
Long Preamble (1)	Specifies that the preamble type is S1G_Long.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Preamble Type (802_11AH)
Resettable	No

Data Rate and Frame Format:Transmission Mode Property

Short Name: Transmission Mode

Property of niWLANGeneration

Specifies the value of the uplink indication field of the S1G-SIG field when you set the Standard property to **80211AH MIMOOFDM**. This property also specifies whether the packet is uplink or downlink when you set the Standard property to **80211AX MIMOOFDM**.

Refer to section 24.3.8.2.1.4 of IEEE Standard P802.11ah/D1.3 for more information about the 80211ah Uplink Indication.

The default value is **DL**.

DL (0)	Specifies that the transmission mode is downlink (DL).
UL (1)	Specifies that the transmission mode is uplink (UL).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Preamble Puncturing:Enabled Property

Short Name: Preamble Puncturing Enabled

Property of niWLANGeneration

Specifies whether to enable preamble puncturing (channel puncturing) on 802.11ax MU PPDU signals. Preamble puncturing is valid only when you set the Channel Bandwidth property to 80 MHz or 160 MHz.

The default value is **False**.

False (0)	Specifies that preamble puncturing is disabled.

True (1)	Specifies that preamble puncturing is enabled.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Preamble Puncturing:Primary 20 MHz Channel Index Property

Short Name: Primary 20 MHz Channel Index

Property of niWLANGeneration

Specifies the index of the primary 20 MHz sub-channel in the channel bandwidth. This property along with the puncturing information defines the mode of puncturing.

The default value is 0. For 80 MHz, valid values are 0 to 3, inclusive. For (80 + 80) MHz and 160 MHz, valid values are 0 to 7, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Preamble Puncturing:Mask Property

Short Name: Preamble Puncturing Mask

Property of niWLANGeneration

Specifies the 20 MHz sub-channels to be punctured in the 802.11ax MU PPDU signal when preamble puncturing is enabled. The mask value specified here is a binary mask represented as an integer, where bit '0' represents the punctured sub-channel.

In the binary value, the least significant bit represents the 20 MHz sub-channel lower in frequency and the most significant bit represents the 20 MHz sub-channel higher in frequency. For (80 + 80) MHz case, LSB represents the lowest sub-channel in first segment. For 80M case, the toolkit considers the least significant 4 bits as the mask value. For (80 + 80) MHz and 160 MHz cases, the toolkit considers the least significant 8 bits.

The default value is 255 (1111 1111) which represents full preamble bandwidth.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:BSS Color Property

Short Name: BSS Color

Property of niWLANGeneration

Specifies the identifier of the BSS (Basic Service Set) from which the 802.11ax PPDU is transmitted.

The default value is 63. The valid values are 0 to 63, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Not Sounding Bit Property

Short Name: Not Sounding Bit

Property of niWLANGeneration

Specifies the value of the Not Sounding field of the HT-SIG field when you set the Standard property to **80211N MIMOOFDM**. For more information about Not Sounding Bit, refer to section 20.3.9.4.3 of **IEEE Standard 802.11n-2009**.

The default value is 1. Valid values are 0 or 1.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Guard Interval Type Property

Short Name: Guard Interval Type

Property of niWLANGeneration

Specifies the type of guard interval (cyclic prefix) in an OFDM symbol.

The following table lists the guard interval length values for different standards.

Standard Property Value	Bandwidth (MHz)	1/4th Guard Interval Length	1/8th Guard Interval Length	1/16th Guard Interval Length
80211N MIMOOFDM	All	0.8 microseconds	0.4 microseconds	N.A
80211AC MIMO OFDM	All	0.8 microseconds	0.4 microseconds	N.A
80211AH MIMO OFDM	All	8 microseconds	4 microseconds	N.A
80211AF MIMOOFDM	6,7	6 microseconds	3 microseconds	N.A
	8	4.5 microseconds	2.25 microseconds	N.A
80211AX MIMOOFDM	All	3.2 microseconds	1.6 microseconds	0.8 microseconds

The default value is **1/4**.

1/4 (0)	Specifies that guard interval length is equal to one-fourth of the IDFT/DFT period.
1/8 (1)	Specifies that guard interval length is equal to one-eighth of the IDFT/DFT period.
1/16 (2)	Specifies that guard interval length is equal to one-sixteenth of the IDFT/DFT period.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Guard Interval Type
Resettable	No

Data Rate and Frame Format:HE-LTF Size Property

Short Name: HE-LTF Size

Property of niWLANGeneration

Specifies the HE-LTF symbol size in the 802.11ax signals. The IEEE Standard 802.11ax specifies the following combinations of the HE-LTF symbol size and the Guard Interval property.

PPDU Type Property Value	HE-LTF Size Property Value	Guard Interval Type Property Value
SU PPDU, Extended Range SU PPDU	**4x**	1/4
	2x	1/8
	2x, 4x, 1x	1/16
MU PPDU	**4x**	1/4
	2x	1/8
	2x, 4x	1/16
Trigger-Based PPDU	**4x**	1/4
	2x, 1x	1/8

The default value is **Auto**.

Auto (-1)	Specifies 4x as HE-LTF symbol duration for the guard interval type value of 1/4; or 2x otherwise.
4x (0)	Specifies 4x as the HE-LTF symbol duration.
2x (1)	Specifies 2x as the HE-LTF symbol duration.
1x (2)	Specifies 1x as the HE-LTF symbol duration.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Midamble Periodicity Property

Short Name: Midamble Periodicity

Property of niWLANGeneration

Specifies the interval, in number of data symbols, after which the midambles are inserted in the data field of the 802.11ax signals. You must set this property to **None** when the value of the Number of Space Time Streams property is greater than 4.

The default value is **None**.

None (0)	Specifies that the midamble is not inserted.
10 Symbols (1)	Specifies that the midamble is inserted after every 10 data symbols.
20 Symbols (2)	Specifies that the midamble is inserted after every 20 data symbols.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:L-SIG Length Property

Short Name: L-SIG Length

Property of niWLANGeneration

Specifies the value of the UL-LENGTH field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of UL-LENGTH is derived from the payload settings.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Pre-FEC Padding Factor Property

Short Name: Pre-FEC Padding Factor

Property of niWLANGeneration

Specifies the value of pre-FEC padding factor sub-field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of pre-FEC padding factor is derived from the payload settings.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:PE Disambiguity Property

Short Name: PE Disambiguity

Property of niWLANGeneration

Specifies the value of the PE disambiguity sub-field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of PE Disambiguity is derived from the payload settings.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:LDPC Extra Symbol Segment Property

Short Name: LDPC Extra Symbol Segment

Property of niWLANGeneration

Specifies the value of the LDPC extra symbol segment field in the trigger frame which is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of the LDPC extra symbol segment field is derived from the payload settings.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Nominal Packet Padding Property

Short Name: Nominal Packet Padding

Property of niWLANGeneration

Specifies the nominal packet padding value used for determining the packet extension duration when you set the Standard property to **80211AX MIMOOFDM**.

When you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU** , this property value corresponds to the maximum nominal packet padding across all users. When you set the PPDU Type property to **Trigger-Based PPDU**; and none of the L-SIG Length property, Pre-FEC Padding Factor property, PE Disambiguity property, and LDPC Extra Symbol Segment property are set to -1, the toolkit ignores this property.

The default value is **Auto**.

Auto (-1)	Specifies that the nominal packet padding is an y of the following: - 0us, if all the resource units are of size l ess than 242 and DCM is disabled. - 0us, if all the resource units are of size l ess than 106 and DCM is enabled. - 16us, otherwise.
0us (0)	Specifies that the nominal packet padding is 0u s.

8us (1)	Specifies that the nominal packet padding is 8u s.
16us (2)	Specifies that the nominal packet padding is 16 us.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Data Rate and Frame Format:Packet Extension Duration (s) Property

Short Name: Packet Extension Duration (s)

Property of niWLANGeneration

Returns the duration of packet extension in the waveform when you set the Standard property to **80211AX MIMOOFDM**. This value is expressed in seconds.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only
High-level VIs	N/A
Resettable	No

MIMO:Number of Segments Property

Short Name: Number of Segments

Property of niWLANGeneration

Specifies the number of frequency segments for 802.11ac or 802.11ax signals.

For 80 MHz + 80 MHz transmission of 802.11ac or 802.11ax signals, set this property to 2 and the Channel Bandwidth property to 80 MHz.

The default value is 1.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Number of Segments
Resettable	No

MIMO:Number of Transmit Channels Property

Short Name: Number of Transmit Channels

Property of niWLANGeneration

Specifies the number of transmit channels for multiple input multiple output (MIMO) signals, as defined in section 20.3.3 of **IEEE Standard 802.11n-2009**.

If you set the Mapping Matrix Type property to **Direct**, the number of transmit channels must be equal to the number of space-time streams. If you set the Mapping Matrix Type property to values other than **Direct**, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the Number of Extension Spatial Streams property. The toolkit defines the number of space-time streams using the MCS Index property and the STBC Index property, as specified in **IEEE Standard 802.11n-2009**.

If you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211B/G DSSS**, or **80211G DSSSOFDM** and the Number of Transmit Channels property to greater than one, the same waveform is generated on multiple RFSG devices inside the niWLANG RFSG Create and Download Waveforms (Multiple Channel) VI.

The default value is 1. If you set the Standard property to **80211N MIMOOFDM** or **80211AH MIMOOFDM**, the valid values are 1 to 4, inclusive. If you set the Standard property to any other value, the valid values are 1 to 8, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Number of Transmit Channels
Resettable	No

MIMO:Number of Space Time Streams Property

Short Name: Number of Space Time Streams

Property of niWLANGeneration

Specifies the number of space-time streams into which the data is divided.

You must use the following active channel string formats to set this property.

Standard Property Value	Active Channel String Format
80211AC MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU**
80211AX MIMOOFDM	"" (empty string), if you set the PPDU Type property to **SU PPDU** or **Extended Range SU PPDU**
	"user**x**", if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**
80211N MIMOOFDM, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**	"" (empty string)

The default value is 1. The valid values for this property are as follows:

Standard Property Value	Valid Values(Inclusive)

80211N MIMOOFDM	1 to 4
80211AC MIMOFDM	1 to 8, if you set the PPDU Type property to **SU PPDU**
	1 to 4, if you set the PPDU Type property to **MU PPDU**
80211AX MIMOFDM	1 to 8, if you set the PPDU Type property to **SU PPDU**
	1 to 4, if you set the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**
	1 to 2, if you set the PPDU Type property to **Exte nded Range SU PPDU**

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Number of Space Time Streams
Resettable	No

MIMO:Space Time Stream Offset Property

Short Name: Space Time Stream Offset

Property of niWLANGeneration

Specifies the space time stream offset which is used for 802.11ax Trigger-Based PPDU generation.

You must use the following active channel string formats to set this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to **Tri gger-Based PPDU**

The default value is 0.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:Number of HE-LTF Symbols Property

Short Name: Number of HE-LTF Symbols

Property of niWLANGeneration

Specifies the number of HE-LTF symbols in the transmitted 802.11ax signal when you set the PPDU Type property to **Trigger-Based PPDU**.

The default value is -1, which indicates that the value is derived from the maximum index of space time streams across users.

The valid values are -1, 1, 2, 4, 6, and 8.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:MU-MIMO LTF Mode Enabled Property

Short Name: MU-MIMO LTF Mode Enabled

Property of niWLANGeneration

Specifies whether the HE-LTF sequence corresponding to each space time stream is masked by a distinct orthogonal code if you set the Standard property to **80211AX MIMOOFDM** and PPDU Type property to **Trigger-Based PPDU**.

The default value is **False**.

False (0)	Specifies that single stream pilots are used in LT F sequence.
True (1)	Specifies that the LTF sequence is HE masked.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:Number of Users Property

Short Name: Number of Users

Property of niWLANGeneration

Specifies the number of users in a multi-user (MU) physical layer convergence procedure (PLCP) protocol data unit (PPDU).

The default value is 1. You can set this property only when you set the Standard property to **80211AC MIMOOFDM** and the PPDU Type property to **MU PPDU**, or when you set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **MU PPDU** or **Trigger-Based PPDU**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Number of Users
Resettable	No

MIMO:Multi-segment Generation Mode Property

Short Name: Multi-segment Generation Mode

Property of niWLANGeneration

Specifies whether to use a single generator or two generators for each channel of a multi-segment (80+80) MHz 802.11ac signal. This property is applicable when the Number of Segments property is 2 and the channel bandwidth is 80 MHz. When you set this property to **Single Generator**, you have to specify the value of the Carrier Frequency property for both the segments.

The default value is **Multiple Generators**.

Single Generator (0)	Specifies that one vector signal generator is used to generate the (80+80) MHz 802.11ac signal.
Multiple Generators (1)	Specifies that two vector signal generators are used to generate the (80+80) MHz 802.11ac signal.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:Spatial Mapping Mode Property

Short Name: Spatial Mapping Mode

Property of niWLANGeneration

Specifies whether the spatial mapping is created from a single global matrix or per user. This property is valid, when you set the PPDU Type property to **Trigger-Based PPDU**.

The default value is **Common**.

Common (0)	Specifies that a single global matrix is used for s patial mapping of all users.
User Specific (1)	Specifies that per user spatial mapping is done.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:Mapping Matrix Type Property

Short Name: Mapping Matrix Type

Property of niWLANGeneration

Specifies the mapping matrix type for mapping space-time streams to transmit channels as defined in section 20.3.11.10.1 of **IEEE Standard 802.11n-2009**.

Note If you set the Mapping Matrix Type property to values other than **Direct**, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the Number of Extension Spatial Streams property if the Standard property is set to **80211N MIMOOFDM**.

The default value is **Direct**.

If you set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **Trigger-Based PPDU**, the N STS value is dependent on the Spatial Mapping Mode property.

Spatial Mapping Mode property value	N_STS value
Common	Maximum number of space time streams across Resource Units (RUs)
User Specific	Number of space time streams for the specified user

You must use the following active channel string formats when you set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **Trigger-Based PPDU**.

Spatial Mapping Mode property value	Active channel string format
Common	"" (empty string)
User Specific	"user**x**"

<table>
<tr>
<td>Direct (0)</td>
<td>
Specifies direct mapping is used for space-time streams.
For direct mapping matrices $\mathbf{N_{Tx}}$ must be equal to $\mathbf{N_{STS}}$. A direct mapping matrix is derived by taking the subset $\mathbf{N_{STS}}$ * $\mathbf{N_{STS}}$ of the 8 × 8 identity matrix.
<table>
<tr><td>where</td><td>$\mathbf{N_{Tx}}$ is the number of transmit channels specified by the Number of Transmit Channels property.</td></tr>
<tr><td></td><td>$\mathbf{N_{STS}}$ is the number of space-time streams. If the Standard property is set to 80211N MIMOOFDM, the value of $\mathbf{N_{STS}}$ is determined by the MCS index and the STBC index</td></tr>
</table>
The following matrix is an example of a 4 × 4 identity matrix.
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

</td>
</tr>
<tr>
<td>Hadamard (1)</td>
<td>

Specifies Hadamard mapping is used for space-time streams and the extension spatial streams.

The Hadamard mapping matrix is of size $\mathbf{N_{Tx}}$ * ($\mathbf{N_{STS}}$ + $\mathbf{N_{ESS}}$), where $\mathbf{N_{Tx}}$ >= ($\mathbf{N_{STS}}$ + $\mathbf{N_{ESS}}$) if the Standard property is set to **80211N MIMOOFDM**.

The Hadamard mapping matrix is of size $\mathbf{N_{Tx}}$ *($\mathbf{N_{STS}}$), where $\mathbf{N_{Tx}}$ >= $\mathbf{N_{STS}}$ if the Standard property is set to **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, or **80211AX MIMOOFDM**.

The Hadamard mapping matrix is derived by taking subset of the 8 × 8 Hadamard matrix.

<table>
<tr><td>where</td><td>$\mathbf{N_{Tx}}$ is the number of transmit channels specified by the Number of Transmit Channels property.</td></tr>
<tr><td></td><td>$\mathbf{N_{STS}}$ is the number of space-time streams. If the Standard property is set to 80211N MIMOOFDM, the value of $\mathbf{N_{STS}}$ is determined by the MCS index and the STBC index.</td></tr>
<tr><td></td><td>$\mathbf{N_{ESS}}$ is the number of extension spatial streams</td></tr>
</table>

The following matrix is an example of a 4 × 4 Hadamard matrix.

</td>
</tr>
</table>

<table>
<tr><td></td><td>$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$</td></tr>
<tr><td>Fourier (2)</td><td>Specifies Fourier mapping is used for space-time streams and the extension spatial streams.

The Fourier mapping matrix is given by the following equation:

$$\{a_{nk}\} = \frac{1}{\sqrt{N}} e^{\left[\frac{j2\pi kn}{N}\right]}$$
<table>
<tr><td>where</td><td>n= 0 … (N - 1)</td></tr>
<tr><td></td><td>N = N_{Tx} , where N_{Tx} is the number of transmit channels specified by the Number of Transmit Channels property.</td></tr>
<tr><td></td><td>k = 0…((N_{STS} + N_{ESS}) - 1) if the Standard property is set to 80211N MIMOOFDM, or k = 0 …(N_{STS} - 1) if the Standard property is set to 80211AC MIMOOFDM, 80211AH MIMOOFDM, or 80211AX MIMOOFDM, N_{STS} is the number of space-time streams. If the Standard property is set to 80211N MIMOOFDM, the value of N_{STS} is determined by the MCS index and the STBC index. N_{ESS} is the number of extension spatial streams.</td></tr>
</table></td></tr>
</table>

	The following matrices are examples of 2 × 2, 3 × 3, and 4 × 4 matrices. $\frac{1}{\sqrt{2}}\begin{bmatrix}1+i0 & 1+i0\\1+i0 & 1+i0\end{bmatrix}$ $\frac{1}{\sqrt{3}}\begin{bmatrix}1+i0 & 1+i0 & 1+i0\\1+i0 & -0.5+i0.9 & -0.5-i0.9\\1+i0 & -0.5-i0.9 & -0.5+i0.9\end{bmatrix}$ $\frac{1}{2}\begin{bmatrix}1+i0 & 1+i0 & 1+i0 & 1+i0\\1+i0 & 0+i1 & -1+i0 & 0-i1\\1+i0 & -1+i0 & 1+i0 & -1+i0\\1+i0 & 0-i1 & -1+i0 & 0+i1\end{bmatrix}$
User Defined (3)	Specifies a user-defined mapping for space-time streams and extension spatial streams. You can set this matrix using the niWLANG Set Mapping Matrix VI.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set Mapping Matrix Type
Resettable	No

MIMO:STBC Index Property

Short Name: STBC Index

Property of niWLANGeneration

Specifies the difference between the number of space-time streams and the number of spatial streams, as defined in section 20.3.9.4.3 of **IEEE Standard 802.11n-2009**. The toolkit derives the number of spatial streams from the specified value of the MCS Index property. Different space-time coding schemes are defined in section 20.3.11.8.1 of **IEEE Standard 802.11n-2009**.

The default value is 0. Valid values are 0 to 2, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niWLANG Set STBC Index
Resettable	No

MIMO:STBC All Streams Enabled Property

Short Name: STBC All Streams Enabled

Property of niWLANGeneration

Specifies whether space-time block coding (STBC) is performed at the transmitter when the Standard property is set to **80211AC MIMOOFDM** or **80211AX MIMOOFDM**. Whenever STBC is performed, the number of space-time streams is equal to two times the number of spatial streams.

The default value is **False**.

False (0)	Specifies that STBC is not performed at the transmitter.
True (1)	Specifies that STBC is performed at the transmitter.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

MIMO:Number of Extension Spatial Streams Property

Short Name: Number of Extension Spatial Streams

Property of niWLANGeneration

Specifies the number of extension spatial streams (**N_{ESS}**) as defined in section 20.3.9.4.6 of **IEEE Standard 802.11n-2009**.

The value of **N_{ESS}** follows the definition shown in the following equation: **N_{STS}** + **N_{ESS}** ≤ **N_{Tx}**

where	**N_{STS}** is the number of space-time streams and is determined by the MCS index and STBC index
	N_{Tx} is the number of transmit channels specified by the Number of Transmit Channels property

The default value is 0. Valid values are 0 to 3, inclusive.

Note **N_{Tx}** must be greater than or equal to **N_{STS}**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:Carrier Frequency Offset (Hz) Property

Short Name: Carrier Frequency Offset (Hz)

Property of niWLANGeneration

Specifies the offset in the center frequency of the complex baseband signal from the carrier frequency. This value is expressed in Hz.

You must use the following active channel string formats to configure this attribute.

Standard Property Value	Active Channel String Format
80211AC MIMOOFDM, 80211AF MIMOOFDM	"segment**x**"
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS , 80211G DSSSOFDM, 80211N MIMOOFDM, 80211AH MIMOOFDM	"" (empty string)
80211AX MIMOOFDM	"segment**x**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**", if you set the PPDU Type property to **Trigger-Based PPDU** (user**x** is optional if you want to apply to all users)

The default value is 0. Valid value is -(1/2) * **IQ Rate** to (1/2) * **IQ Rate**, inclusive, where **IQ Rate** is the value of the IQ Rate property.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:Sample Clock Offset (ppm) Property

Short Name: Sample Clock Offset (ppm)

Property of niWLANGeneration

Specifies the offset in the Sample Clock frequency from the sampling frequency defined by the following equation:

Sampling Frequency = max (maximum hardware I/Q Rate, **Oversampling Factor** * **Nyquist Sampling Rate**)

where

Maximum Hardware I/Q is the value of the Maximum Hardware IQ Rate property

Oversampling Factor is the value of the Oversampling Factor property

This value is expressed in parts per million (ppm).

For large offset values, with large waveform sizes, clock cycle slips may occur. Clock cycle slips can cause the final waveform size to be different from the expected size, given the ideal burst length and the interframe spacing.

Standard Property Value	Active Channel String Format
80211AC MIMOOFDM, 80211AF MIMOOFDM	"segment**x**"
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B/G DSSS , 80211G DSSSOFDM, 80211N MIMOOFDM, 80211AH MIMOOFDM	"" (empty string)
80211AX MIMOOFDM	"segment**x**", if you set the PPDU Type property to SU PPDU, Extended Range SU PPDU or MU PPDU
	"[user**x**]/segment**y**", if you set the PPDU Type property to Trigger-Based PPDU (user**x** is optional if you want to apply to all users)

The default value is 0. Valid values are -1000 to 1000, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Impairments:IQ Impairments:Enabled Property

Short Name: IQ Impairments Enabled

Property of niWLANGeneration

Specifies whether to apply I/Q impairments such as I DC offset, Q DC offset, quadrature skew, and I/Q gain imbalance to the waveform.

The default value is **True**.

False (0)	Does not apply I/Q impairments.
True (1)	Applies I/Q impairments.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:IQ Impairments:IQ Gain Imbalance (dB) Property

Short Name: IQ Gain Imbalance (dB)

Property of niWLANGeneration

Specifies the ratio of the mean amplitude of the in-phase (I) signal to the mean amplitude of the quadrature-phase (Q) signal. This value is expressed in dB.

Note If you set this property to a large value, you may experience dynamic range loss at the digital-to-analog converter (DAC).

I/Q gain imbalance follows the definition shown in the following equation:

$$\mathbf{I'} = \mathbf{I} - \boldsymbol{\gamma} * \sin(\phi) * \mathbf{Q} + \mathbf{I}_0$$

$$\mathbf{Q'} = \boldsymbol{\gamma} * \cos(\phi) * \mathbf{Q} + \mathbf{Q}_0$$

where	$\boldsymbol{\gamma} = 10$
	$\boldsymbol{\phi}$ is the quadrature skew
	I is the in-phase component of the waveform before applying impairments
	Q is the quadrature component of the waveform before applying impairments
	I' is the in-phase component of the waveform after applying impairments
	Q' is the quadrature component of the waveform after applying impairments
	$\mathbf{I}_0 = (\mathbf{I}_{RMS} * \mathbf{I}_{DC\ Offset})/100$
	$\mathbf{Q}_0 = (\mathbf{Q}_{RMS} * \mathbf{Q}_{DC\ Offset})/100$

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"
80211AC MIMOOFDM , 80211AF MIMOOFDM	"segment**x**/channel**y**"
80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 0. Valid values are -6 to +6, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL

Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:IQ Impairments:I DC Offset (%) Property

Short Name: I DC Offset (%)

Property of niWLANGeneration

Specifies the value of the DC offset in the in-phase (I) channel as a percentage of the RMS magnitude of the unaltered I channel.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"channel**x**"
80211AC MIMOOFDM, **80211AF MIMOOFDM**	"segment**x**/channel**y**"
80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 0. Valid values are -100 to +100, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:IQ Impairments:Q DC Offset (%) Property

Short Name: Q DC Offset (%)

Property of niWLANGeneration

Specifies the value of the DC offset in the quadrature-phase (Q) channel as a percentage of the RMS magnitude of the unaltered Q channel.

Standard Property Value	Active Channel String Format
80211A/G OFDM, **80211J OFDM**, **80211P OFDM**, **80211B DSSS**, **80211G DSSSOFDM**	"" (empty string)
80211N MIMOOFDM, **80211AH MIMOOFDM**	"channel**x**"
80211AC MIMOOFDM, **80211AF MIMOOFDM**	"segment**x**/channel**y**"
80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 0. Valid values are -100 to +100, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:IQ Impairments:Quadrature Skew (deg) Property

Short Name: Quadrature Skew (deg)

Property of niWLANGeneration

Specifies the deviation in angle from 90 degrees between the in-phase (I) and quadrature-phase (Q) signals. This value is expressed in degrees.

Quadrature skew follows the definition shown in the following equation:

$\mathbf{I'} = \mathbf{I} - \boldsymbol{\gamma} * \sin(\phi) * \mathbf{Q} + \mathbf{I}_0$

$\mathbf{Q'} = \boldsymbol{\gamma} * \cos(\boldsymbol{\phi}) * \mathbf{Q} + \mathbf{Q}_0$

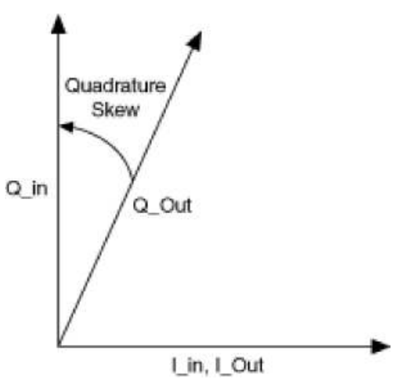
where $\boldsymbol{\gamma} = 10$

ϕ = quadrature skew

$\mathbf{I}_0 = (\mathbf{I}_{RMS} * \mathbf{I}_{DC\ Offset})/100$

$\mathbf{Q}_0 = (\mathbf{Q}_{RMS} * \mathbf{Q}_{DC\ Offset})/100$

The following figure is a graphical representation of the quadrature skew.



Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"

80211AC MIMOOFDM, 80211AF MIMOOFDM	"segment**x**/channel**y**"
80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 0. Valid values are -30 to +30, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:IQ Impairments:Timing Skew (s) Property

Short Name: Timing Skew (s)

Property of niWLANGeneration

Specifies the difference between the sampling instants of I and Q streams. This value is expressed in seconds.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"
80211AC MIMOOFDM, 80211AF MIMOOFDM	"segment**x**/channel**y**"

80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 0. Valid values are -1 microsecond to 1 microsecond, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:Time Delay (s) Property

Short Name: Time Delay (s)

Property of niWLANGeneration

Specifies the time delay for each user within an 802.11ax Trigger-Based signal. This value is expressed in seconds. You must set the Standard property to **80211AX MIMOOFDM** and the PPDU Type property to **Trigger-Based PPDU**. Use this property to introduce relative time delays between multiple users within an 802.11ax Trigger-Based signal.

You must use the following active channel string formats to set this property.

Standard Property Value	Active Channel String Format
80211AX MIMOOFDM	"user**x**", if you set the PPDU Type property to Trigger-Based PPDU.

The default value is 0.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:AWGN:Enabled Property

Short Name: AWGN Enabled

Property of niWLANGeneration

Specifies whether to introduce additive white Gaussian noise (AWGN) to the baseband waveform. The toolkit uses the value specified in the Carrier to Noise Ratio property to add the AWGN.

The default value is **False**.

False (0)	Disables AWGN addition.
True (1)	Enables AWGN addition.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Impairments:AWGN:Carrier to Noise Ratio (dB) Property

Short Name: Carrier to Noise Ratio (dB)

Property of niWLANGeneration

Specifies the carrier-to-noise ratio (CNR) of the waveform generated. Noise bandwidth is equal to the value of the IQ Rate property. The toolkit ignores the Carrier to Noise Ratio property if you set the AWGN Enabled property to **False**. This value is expressed in dB.

Standard Property Value	Active Channel String Format
80211A/G OFDM, 80211J OFDM, 80211P OFDM, 80211B DSSS, 80211G DSSSOFDM	"" (empty string)
80211N MIMOOFDM, 80211AH MIMOOFDM	"channel**x**"
80211AC MIMOOFDM, 80211AF MIMOOFDM	"segment**x**/channel**y**"
80211AX MIMOOFDM	"segment**x**/channel**y**", if you set the PPDU Type property to **SU PPDU**, **Extended Range SU PPDU** or **MU PPDU**
	"[user**x**]/segment**y**/channel**z**", if you set the PPDU Type property to **Trigger-Based PPDU** (user **x** is optional if you want to apply to all users)

The default value is 50. Valid values are -100 to 100, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:Pulse Shaping Filter Enabled Property

Short Name: Pulse Shaping Filter Enabled

Property of niWLANGeneration

Specifies whether to apply a pulse-shaping filter to the generated signal.

Note The toolkit ignores this property and enables the pulse shaping filter, if you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **1.0.0**.

If you set the Standard property to **80211B/G DSSS** or **80211P OFDM**, the default value is **True**. If you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211G DSSSOFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**, the default value is **False**.

False (0)	Disables the pulse-shaping filter.
True (1)	Enables the pulse-shaping filter.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:Pulse Shaping Filter Type Property

Short Name: Pulse Shaping Filter Type

Property of niWLANGeneration

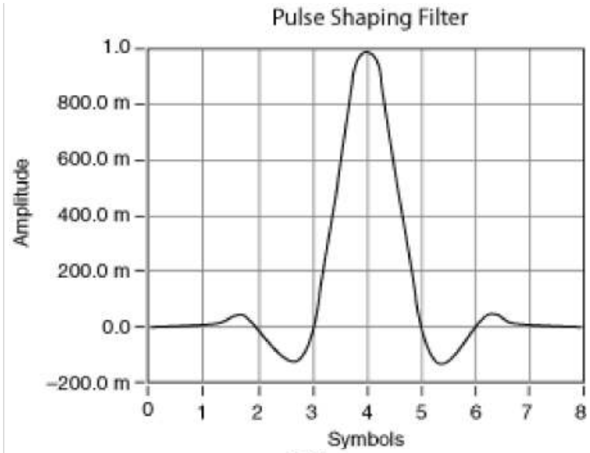
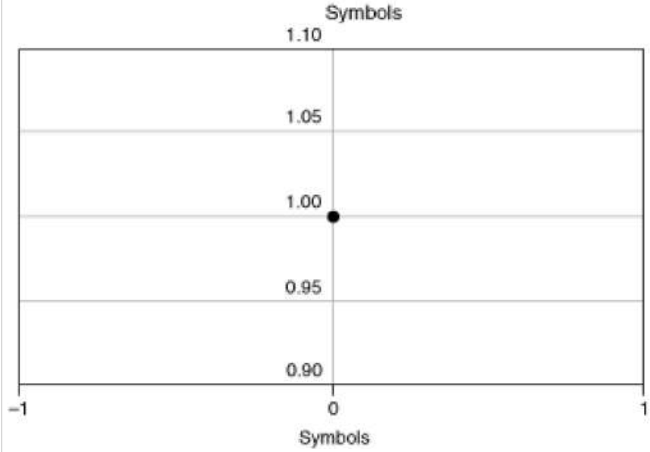
Specifies the pulse-shaping filter type to use to ensure that the signal spectrum meets the spectral mask criteria as defined in section 17.3.9.2 of **IEEE Standard 802.11a-1999**, section 18.4.7.3 of **IEEE Standard 802.11b-1999**, and section 20.3.21.1 of **IEEE Standard 802.11n-2009**.

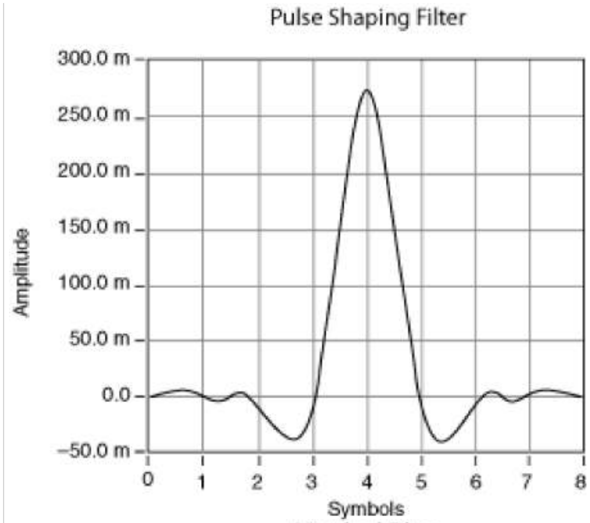
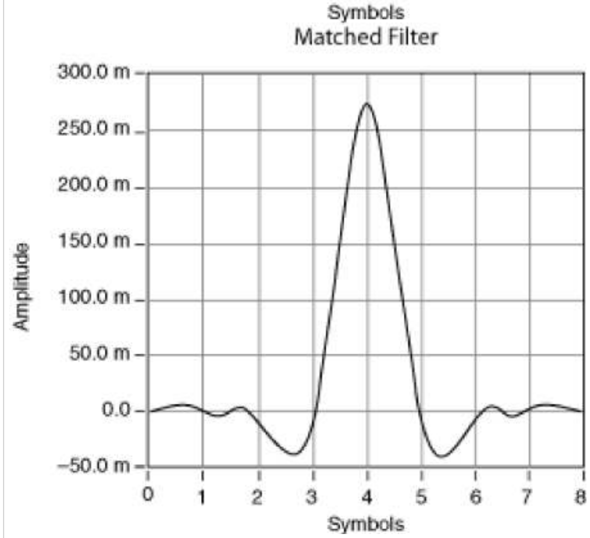
The default value is **Rectangular** if you set the Standard property to **80211A/G OFDM**, **80211J OFDM**, **80211G DSSSOFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**.

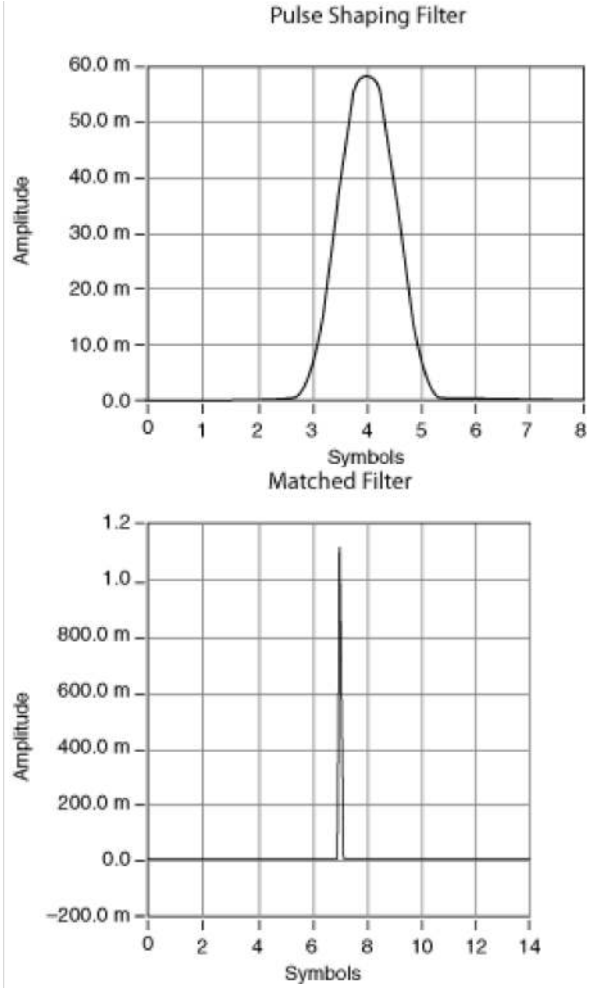
The default value is **Root Raised Cosine** if you set the Standard property to **80211B/G DSSS**.

The default value is **Raised Cosine** if you set the Standard property to **80211P DSSS**.

Rectangular (0)	Generates a rectangular filter that is always one symbol long.
Raised Cosine (1)	Generates a filter with a frequency-domain resp onse in the transition band that has the shape o f a raised cosine. The filter is defined by a roll-off factor configured using the Pulse Shaping Filter Parameter property.

	 
Root Raised Cosine (2)	Generates a filter with a frequency-domain response in transition band that has the shape of a square root of a raised cosine. The filter is defined by a roll-off factor configured using the Pulse Shaping Filter Parameter property.

	
Gaussian (3)	Generates a filter with a frequency-domain response and time-domain response that are Gaussian. The filter is defined by the product of 3 dB bandwidth and symbol time, configured using the Pulse Shaping Filter Parameter property.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:Pulse Shaping Filter Parameter Property

Short Name: Pulse Shaping Filter Parameter

Property of niWLANGeneration

Specifies the value of the rolloff factor (alpha) if you set the Pulse Shaping Filter Type property to **Raised Cosine** or **Root Raised Cosine**. If you set the Pulse Shaping Filter Type property to **Gaussian**, you can calculate the Pulse Shaping Filter Parameter property by multiplying **B** and **T**, where **B** is the 3 dB bandwidth and **T** is the symbol period for a Gaussian filter.

If you set the Pulse Shaping Filter Type property to **Rectangular**, the toolkit ignores the Pulse Shaping Filter Parameter property.

If you set the Standard property to **80211P OFDM**, the default value is 0.1. In all other instances, the default value is 0.5. Valid values are 0.1 to 0.95, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:Pulse Shaping Filter Length Property

Short Name: Pulse Shaping Filter Length

Property of niWLANGeneration

Specifies the length of the pulse-shaping filter. This value is expressed in symbols.

The length affects the frequency response of the filter. The toolkit ignores this property when the Pulse Shaping Filter Enabled property is set to **False**.

If you set the Standard property to **80211P OFDM**, the default value is 100. In all other instances, the default value is 8.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:Windowing Method Property

Short Name: Windowing Method

Property of niWLANGeneration

Specifies the method of applying window to the baseband signal if the Standard property is set to **80211A/G OFDM**, **80211J OFDM**, **80211P OFDM**, **80211G DSSSOFDM**, **80211N MIMOOFDM**, **80211AC MIMOOFDM**, **80211AH MIMOOFDM**, **80211AF MIMOOFDM**, or **80211AX MIMOOFDM**. This property is ignored if the Standard property is set to **80211B/G DSSS**.

Refer to the Windowing topic for more information about windowing for OFDM signals.

The default value is **Centered At Symbol Boundary**.

Centered At Symbol Boundary (0)	Specifies that the window is applied with its cen ter at the boundary between two OFDM symbol s.
Starting At Symbol Boundary (1)	Specifies that the window is applied with its star ting position at the boundary between two OFD M symbols.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	N/A
Resettable	No

Spectrum Control:DSSS Window Length (s) Property

Short Name: DSSS Window Length (s)

Property of niWLANGeneration

Specifies the window length for direct spread spectrum signals If you do not want windowing, set this property to 0. This value is expressed in seconds.

This property provides power ramp-up and ramp-down for the entire burst.

Refer to the Windowing help topic for more information about windowing for DSSS signals.

The default value is 2 microseconds.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Spectrum Control:OFDM Window Length (samples) Property

Short Name: OFDM Window Length (samples)

Property of niWLANGeneration

Specifies the window length, for OFDM signals at the sampling rate equal to the channel bandwidth. For example, if the window length is 2, the channel bandwidth is 20 MHz and the oversampling factor is 4, the samples over which windowing is applied is 8. This value is expressed in samples.

This property provides a smooth, spurious-free transition from the end of one OFDM symbol to the cyclic prefix of the next symbol. If you do not want windowing, set this property to 0.

Refer to the Windowing help topic for more information about windowing for OFDM signals.

The default value is 2.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Advanced:Swap I and Q Enabled Property

Short Name: Swap I and Q Enabled

Property of niWLANGeneration

Specifies whether to swap the data in the I and Q streams.

The default value is **False**.

False (0)	Specifies that the toolkit does not swap the dat a in the I and Q streams.
True (1)	Specifies that the toolkit swaps the data in the I and Q streams.

Remarks

The following table lists the characteristics of this property.

Datatype	I32

Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Advanced:Toolkit Compatibility Version Property

Short Name: Toolkit Compatibility Version

Property of niWLANGeneration

Indicates the **toolkit compatibility version** parameter of the niWLANG Open Session VI.

1.0.0 (010000)	Indicates that the **toolkit compatibility version** parameter of the niWLANG Open Session VI is set to **1.0.0**.
2.0.0 (020000)	Indicates that the **toolkit compatibility version** parameter of the niWLANG Open Session VI is set to **2.0.0**.
3.0.0 (030000)	Indicates that the **toolkit compatibility version** parameter of the niWLANG Open Session VI is set to **3.0.0**.
4.0.0 (040000)	Indicates that the **toolkit compatibility version** parameter of the niWLANG Open Session VI is set to **4.0.0**.
5.0.0 (050000)	Indicates that the **toolkit compatibility version** parameter of the niWLANG Open Session VI is set to **5.0.0**.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A

Resettable	No

Advanced:IQ Waveform Size Property

Short Name: IQ Waveform Size

Property of niWLANGeneration

Returns the size of the generated I/Q waveform. This value is expressed in samples.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	niWLANG Get IQ Waveform Size
Resettable	No

Advanced:Sample Clock Rate Factor Property

Short Name: Sample Clock Rate Factor

Property of niWLANGeneration

Specifies the factor by which the Sample Clock rate is multiplied to generate a signal that is compressed in the frequency domain and expanded in the time domain.

For example, a 40 MHz 802.11n signal can be compressed in the frequency domain to 20 MHz, if the Sample Clock rate is reduced to half. In this case, you must set this property to 0.5 to generate the signal.

The default value is 1. Valid values are 0.001 to 1, inclusive.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A

Resettable	No

Obsolete:Legacy Scaling Enabled Property

Short Name: Legacy Scaling Enabled

Property of niWLANGeneration

Specifies whether to enable standard-defined transmit chain scaling of the legacy part of the high throughput (HT) and very high throughput (VHT) frames.

Note Configure this property only when you set the Standard property to **80211N MIMOOFDM** or **80211AC MIMOOFDM**. The toolkit ignores the Legacy Scaling Enabled property for all other standards.

Note This property is available only when you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **2.0.0** or **3.0.0**.

The default value is **True**.

False (0)	Disables the legacy part of the HT and VHT fram e.
True (1)	Enables the legacy part of the HT and VHT frame .

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Obsolete:Power Level (dBm) Property

Short Name: Power Level (dBm)

Property of niWLANGeneration

Specifies the average power level of the active portion of the burst for signal generation. The active portion of the burst is the WLAN packet excluding the interframe spacing. This value is expressed in dBm.

Note This property is available only if you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **1.0.0** or **2.0.0**.

If you set the Standard property to **80211N MIMOOFDM**, use an active channel string to configure this property.

The default value is -10.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Obsolete:Waveform Scaling Factor Property

Short Name: Waveform Scaling Factor

Property of niWLANGeneration

Specifies the scaling factor for the waveform, as a percentage of the maximum sample magnitude, to reduce the overshoot associated with the digital-to-analog converter (DAC) interpolation filter and other finite impulse response (FIR) filters in the NI RF vector signal generators.

Note This property is available only when you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **1.0.0**.

The default value is 99. Valid values are 1 to 100, inclusive. You can reduce the value to avoid clipping, but you may lose dynamic range.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

Obsolete:Peak to Average Power Ratio (dB) Property

Short Name: PAPR (dB)

Property of niWLANGeneration

Returns the peak-to-average power ratio (PAPR) of the output complex waveform. This value is expressed in dB.

Note This property is available only when you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **1.0.0**.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read Only

High-level VIs	N/A
Resettable	No

Obsolete:Guard Interval (s) Property

Short Name: Guard Interval (s)

Property of niWLANGeneration

Specifies the length of the cyclic prefix (CP) of an OFDM symbol, as specified in section 20.1.1 of **IEEE Standard 802.11n-2009** and section 22.3.6 of **IEEE Standard 802.11ac-2013**. This value is expressed in nanoseconds.

The guard interval can be 800 nanoseconds (long) or 400 nanoseconds (short).

Note The toolkit does not support short guard interval if the modulation and coding scheme (MCS) index is 0-7 and the physical layer convergence protocol (PLCP) frame format is Greenfield format.

Note This property is available only when you set the **toolkit compatibility version** parameter of the niWLANG Open Session VI to **2.0.0** or **3.0.0**.

The default value is 800 ns. Valid values are 400 nanoseconds and 800 nanoseconds.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Resettable	No

NI WLAN Generation Toolkit C Reference

This help file contains information about the WLAN Generation functions, attributes, and values that you can use when programming your application.

WLAN Generation Functions

Contains functions to set various parameters used for generating WLAN standard-compliant bursts and configuring the NI RF vector signal generators.

niWLANG_OpenSession

int32 __stdcall niWLANG_OpenSession (char sessionName[], int32 compatibilityVersion, niWLANGenerationSession *session, int32 *isNewSession);

Purpose

Looks up an existing niWLAN generation session using the sessionName parameter and returns the refnum that you can pass to subsequent niWLAN generation functions. If the lookup fails, the `niWLANG_OpenSession` function creates a new niWLAN generation session and returns a new refnum.

Parameters

Input

Name	Type	Description
sessionName	char[]	Specifies the name of the session that you are looking up or creating. If a session with the same name already exists, this function returns a reference to that session. To get the reference to an already-opened session x, specify x as the session name. You can obtain the reference to an existing session multiple times if you have not called the niWLANG_CloseSession function in that session. You do not need to close the session

<table>
<tr><td></td><td></td><td>multiple times. To create an unnamed session, pass an empty string or NULL to the sessionName parameter.

Tip NI recommends that you call the niWLANG_CloseSession function for each unique named instance of the niWLANG_OpenSession function or each instance of the niWLANG_OpenSession function with an unnamed session.</td></tr>
<tr><td>compatibilityVersion</td><td>int32</td><td>Specifies the version of the toolkit to which the current version of the toolkit is compatible. If the behavior of the toolkit changes in a new version, use this parameter to specify that you want to continue using the behavior of the previous release. The default value is NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

<table><tr><td>NIWLANG_VAL_COMPATIBILI</td><td>Specifies that the toolkit exhibits</td></tr></table></td></tr>
</table>

TY_VERSION_010000(10000)	version 1.0.0 behavior, and all new features in later releases are unavailable. Select this option if you purchased version 1.0.0 and want to maintain functional behavior. Refer to the NI WLAN Generation Toolkit Readme for a list of changes between versions 1.0.0 and later versions.
NIWLANG_VAL_COMPATIBILITY_VERSION_020000(20000)	Specifies that the toolkit exhibits version 2.0.0 behavior, and all new features in later releases are unavailable. Select this option if you purchased version 2.0.0 and want to maintain functional

	behavior. Refer to the NI WLAN Generation Toolkit Readme for a list of changes between versions 1.0.0 and 2.0.0.
NIWLANG_VAL_COMPATIBILITY_VERSION_030000(30000)	Specifies that the toolkit exhibits version 3.0.0 behavior. Select this option to if you want 3.0.0 behavior and access to new features and bug fixes. Refer to the NI WLAN Generation Toolkit Readme for a list of changes between versions 2.0.0 and 3.0.0.
NIWLANG_VAL_COMPATIBILITY_VERSION_040000(40000)	Specifies that the toolkit exhibits version 4.0.0 behavior. Select this option to if you want 4.0.0 behavior and access to new features and

	bug fixes. Refer to the NI WLAN Generation Toolkit Readme for a list of changes between versions 3.0.0 and 4.0.0.
NIWLANG_VAL_COMPATIBILITY_VERSION_050000(50000)	Specifies that the toolkit exhibits version 5.0.0 behavior. Select this option to if you want 5.0.0 behavior and access to new features and bug fixes. Refer to the NI WLAN Generation Toolkit Readme for a list of changes between versions 4.0.0 and 5.0.0.

Output

Name	Type	Description
session	niWLANGenerationSession*	Returns the niWLAN generation session refnum. Use this parameter to configure the behavior and operation of the appropriate NI WLAN Generation Toolkit functions that accept the session

		reference as an input. Note Close the niWLAN generation session reference using the niWLANG Close Session function before the completion of execution to avoid possible memory leak issues.
isNewSession	int32*	Indicates whether the function creates a new session. NIWLANG_VAL_FALSE (0): Indicates that the function returns a reference to an existing session. NIWLANG_VAL_TRUE (1): Indicates that the function creates a new session.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the

function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Configure Generation

Contains functions to configure various parameters for WLAN signals.

niWLANG_SetStandard

int32 __stdcall niWLANG_SetStandard (niWLANGenerationSession session, char channelString[], int32 standard);

Purpose

Specifies the IEEE 802.11 standard, which indicates the type of physical layer, for signal generation.

Note If you do not select a standard, the toolkit returns an error.

Parameters

Input

Name	Type	Description

session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
standard	int32	Specifies the standard for signal generation.

`NIWLANG_VAL_STANDARD_80211AG_OFDM` (0)	Corresponds to the OFDM mode defined in IEEE Standard 802.11a-1999, IEEE Standard 802.11j-2004, IEEE Standard 802.11p-2010, and the extended rate physical layer-OFDM (ERP-OFDM) mode, as defined in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211J_OFDM` (7)	Corresponds to the OFDM mode defined in IEEE Standard 802.11j-2004.
`NIWLANG_VAL_STANDARD_80211P_OFDM` (8)	Corresponds to the OFDM mode defined in IEEE Standard 802.11p-2010.
`NIWLANG_VAL_STANDARD_80211BG_DSSS` (1)	Corresponds to all the compulsory and optional modes defined in IEEE Standard 802.11b-1999 and th

	e ERP-packet binary convolutional coding (ERP-PBCC) mode defined in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM` (2)	Corresponds to the optional direct sequence spread spectrum-OFDM (DSSS-OFDM) mode defined in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM` (3)	Corresponds to IEEE Standard 802.11n-2009. To use this option, you must set the compatibilityVersion parameter of the [niWLANG_OpenSession]() function to `NIWLANG_VAL_COMPATIBILITY_VERSION_020000`, NIWLANG_VAL_COMPATIBILITY_VERSION_030000, `NIWLANG_VAL_COMPATIBILITY_VERSION_040000`, or `NIWLANG_VAL_COMPATIBILIT`

	Y_VERSION_050000.
NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM (4)	Corresponds to IEEE Standard 802.11ac-2013. To use this option, you must set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000 , NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM (5)	Corresponds to IEEE P802.11ah/D1.3. To use this option, you must set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG

	_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM (6)	Corresponds to IEEE Standard 802.11af-2013. To use this option, you must set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM (9)	Corresponds to IEEE P802.11ax/D6.0. To use this option, you must set the compatibilityVersion parameter of the niWLANG_OpenSe

		ssion function to `NIWLANG_VAL_COMPATIBILITY_VERSION_030000`, `NIWLANG_VAL_COMPATIBILITY_VERSION_040000`, or `NIWLANG_VAL_COMPATIBILITY_VERSION_050000`.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_SetOFDMDataRate

int32 __stdcall niWLANG_SetOFDMDataRate (niWLANGenerationSession session, char channelString[], int32 ofdmDataRate);

Purpose

Specifies the data rate for the OFDM payload, as defined in section 17.3.2.2 of IEEE Standard 802.11a-1999. This value is expressed is Mbps.

	Note Configure this function only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
ofdmDataRate	int32	Specifies the data rate for the OFDM payload. This value is expressed in Mbps. The default value is NIWLANG_VAL_OFDM_DATA_RATE_6.

NIWLANG_VAL_OFDM_DATA_RATE_6 (0)	Specifies a data rate of 1.5 Mbps, 3 Mbps, and 6 Mbps for respective channel bandwidths of 5 MHz, 10

	MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_9 (1)	Specifies a data rate of 2.25 Mbps, 4.5 Mbps, and 9 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_12 (2)	Specifies a data rate of 3 Mbps, 6 Mbps, and 12 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_18 (3)	Specifies a data rate of 4.5 Mbps, 9 Mbps, and 18 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_24 (4)	Specifies a data rate of 6 Mbps, 12 Mbps, and 24 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DAT	Specifies a data rate of 9 Mbps, 18 Mbps, an

A_RATE_36 (5)	d 36 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_48 (6)	Specifies a data rate of 12 Mbps, 24 Mbps, and 48 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_54 (7)	Specifies a data rate of 13.5 Mbps, 27 Mbps, and 54 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetDSSSDataRate

int32 __stdcall niWLANG_SetDSSSDataRate (niWLANGenerationSession session, char channelString[], int32 dsssDataRate);

Purpose

Specifies the data rate for the direct sequence spread spectrum (DSSS) payload, as defined in IEEE Standard 802.11b-1999 and the extended rate physical layer-packet binary convolutional coding (ERP-PBCC) mode in IEEE Standard 802.11g-2003.

	Note Configure this function only if you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211BG_DSSS`.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
dsssDataRate	int32	Specifies the data rate for the DSSS payload. The default value is NIWLANG_VAL_DSSS_DATA_RATE_1.

NIWLANG_VAL_DSSS_DATA_RATE_1 (0)	Specifies a data rate of 1, as defined in sections 18.4.6.3 and 18.4.6.4 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_2 (1)	Specifies a data rate of 2, as defined in sections 18.4.6.3 and 18.4.6.4 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_5p5_CCK (2)	Specifies a data rate of 5.5 complementary code keying (CCK), as defined in section 18.4.6.5 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_5p5_PBCC (3)	Specifies a data rate of 5.5 PBCC, as defined in section 18.4.6.6 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_11_CCK (4)	Specifies a data rate of 11 CCK, as defined in section 18.4.6.5 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_11_PBCC (5)	Specifies a data rate of 11 PBCC, as defined in section 18.4.

	6.6 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_22 (6)	Specifies a data rate of 22, as defined in section 19.3.3.2 of IEEE Standard 802.11g-2003.
NIWLANG_VAL_DSSS_DATA_RATE_33 (7)	Specifies a data rate of 33, as defined in section 19.3.3.2 of IEEE Standard 802.11g-2003.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetDSSSPreambleType

int32 __stdcall niWLANG_SetDSSSPreambleType (niWLANGenerationSession session, char channelString[], int32 preambleFormat);

Purpose

Specifies whether to use a long or short preamble for direct sequence spread spectrum (DSSS) and DSSS-OFDM packets as defined in IEEE Standard 802.11b-1999.

	Note Configure this function only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
preambleFormat	int32	Specifies whether to use a long or short preamble for DSSS and DSSS-OFDM packets. The default value is NIWLANG_VAL_PREAMBLE_TYPE_LONG_PREAMBLE.

NIWLANG_VAL_PREAMBLE_TYPE_LONG_PREAMBLE (0)	Uses a long preamble, as defined in section 18.2.2.1 of IEEE Standard 802.11b-1999.

NIWLANG_VAL_PREAMBLE_TYPE_SHORT_PREAMBLE (1)	Uses a short preamble, as defined in section 18.2.2.2 of IEEE Standard 802.11b-1999.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

niWLANG_SetChannelBandwidth

int32 __stdcall niWLANG_SetChannelBandwidth (niWLANGenerationSession session, char channelString[], float64 channelBandwidth);

Purpose

Specifies the channel width used for transmitting the signal. This value is expressed in Hz.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, the channel bandwidth must be equal to 20 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211J_OFDM, the channel bandwidth must be equal to 10 MHz or 20 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211P_OFDM, the channel bandwidth must be equal to 5 MHz, 10 MHz, or 20 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, the channel bandwidth must be either 20 MHz or 40 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the channel bandwidth must be equal to 20 MHz, 40 MHz, 80 MHz, or 160 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the channel bandwidth must be equal to 1 MHz, 2 MHz, 4 MHz, 8 MHz, or 16 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, the channel bandwidth must be equal to 6 MHz, 7 MHz, or 8 MHz. For OFDM signals, channel bandwidth determines the number of pilot and data subcarriers used. The toolkit ignores the niWLANG_SetChannelBandwidth function for other values of the NIWLANG_STANDARD attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
channelBandwidth	float64	Specifies the channel width used for transmitting the signal. The default value is 20 MHz. Valid values are 1 MHz, 2 MHz, 4

		MHz, 5 MHz, 6 MHz, 7 MHz, 8 MHz, 10 MHz, 16 MHz, 20 MHz, 40 MHz, 80 MHz, and 160 MHz. This value is expressed in Hz.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_SetNumberOfTransmitChannels

int32 __stdcall niWLANG_SetNumberOfTransmitChannels (niWLANGenerationSession session, char channelString[], int32 numberOfTransmitChannels);

Purpose

Specifies the number of transmit channels for multiple input multiple output (MIMO) signals, as defined in section 20.3.3 of IEEE Standard 802.11n-2009.

The number of transmit channels must be greater than or equal to the number of space-time streams if you set the NIWLANG_MAPPING_MATRIX_TYPE attribute to NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT. If you set the NIWLANG_MAPPING_MATRIX_TYPE attribute to values other than NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the NIWLANG_NUMBER_OF_EXTENSION_SPATIAL_STREAMS attribute. The toolkit defines the number of space-time streams using the NIWLANG_MCS_INDEX attribute and the NIWLANG_STBC_INDEX attribute as specified in IEEE Standard 802.11n-2009.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM and the niWLANG_SetNumberOfTransmitChannels function to greater than one, the same waveform is generated on multiple RFSG devices inside the niWLANG_RFSGCreateAndDownloadMIMOWaveforms function.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
numberOfTransmitChannels	int32	Specifies the number of transmit channels for MIMO signals. The default value is 1. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_802

		11N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the valid values are 1 to 4, inclusive. If you set the `NIWLANG_STANDARD` attribute to any other value, the valid values are 1 to 8, inclusive.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetMCSIndex

int32 __stdcall niWLANG_SetMCSIndex (niWLANGenerationSession session, char channelString[], int32 MCSIndex);

Purpose

Specifies the value of the modulation and coding scheme (MCS) index. The MCS index is a compact representation that determines the modulation scheme, coding rate, and number of spatial streams as specified in section 20.3.5 of IEEE Standard 802.11n-2009, section 22.5 of IEEE Standard 802.11ac-2013, section 24.5 of IEEE Standard P802.11ah/D1.3, section 23.5 of IEEE Standard 802.11af-2013, and section 28.5 of IEEE P802.11ax/D6.0.

To understand which active channel strings are required to configure this function, refer to the `NIWLANG_MCS_INDEX` attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
MCSIndex	int32	Specifies the value of the MCS index. The default value is 0. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM`, valid values are 0 to 11, inclusive. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM`, valid values are 0 to 13, inclusive. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`, valid values are 0 to 32, inclusive. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM`, valid values

		are 0 to 10, inclusive. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM`, valid values are 0 to 9, inclusive.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLAN_SetSTBCIndex

int32 __stdcall niWLANG_SetSTBCIndex (niWLANGenerationSession session, char channelString[], int32 STBCIndex);

Purpose

Specifies the difference between the number of space-time streams and the number of spatial streams, as defined in section 20.3.9.4.3 of IEEE Standard 802.11n-2009. The toolkit derives the number of spatial streams from the specified value of the NI WLANG_MCS_INDEX attribute. Different space-time coding schemes are defined in section 20.3.11.8.1 of IEEE Standard 802.11n-2009.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
STBCIndex	int32	Specifies the value of the space-time block coding (STBC) index. The default value is 0. Valid values are 0 to 2, inclusive.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_Set80211nPLCPFrameFormat

int32 __stdcall niWLANG_Set80211nPLCPFrameFormat (niWLANGenerationSession session, char channelString[], int32 frameFormat);

Purpose

Specifies the format of the physical layer convergence protocol (PLCP) frame structure. The frame structure determines the arrangement of preambles, header (SIGNAL field), and payload in a frame as defined in section 20.3.2 of IEEE Standard 802.11n-2009.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
frameFormat	int32	Specifies the PLCP frame structure to use. The default value is `NIWLANG_VAL_80211N_PLCP_FRAME_FORMAT_MIXED`.

`NIWLANG_VAL_80211N_PLCP_FRAME_`	Specifies that the PLCP frame

FORMAT_MIXED (0)	structure consists of non-HT preamble and header (Signal field) followed by high throughput (HT) header, preambles, and payload as specified in IEEE Standard 802.11n-2009.
NIWLANG_VAL_80211N_PLCP_FRAME_FORMAT_GREENFIELD (1)	Specifies that the PLCP frame structure does not support non-HT, and starts with HT preamble, followed by HT SIGnal field, and payload.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetOverSamplingFactor

int32 __stdcall niWLANG_SetOverSamplingFactor (niWLANGenerationSession session, char channelString[], int32 overSamplingFactor);

Purpose

Specifies the number of times the toolkit increases the Nyquist sample rate to obtain the final sample rate of the signal.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
overSamplingFactor	int32	Specifies the oversampling factor. The default value is 4 and the minimum value is 2.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status

code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetOFDMGuardIntervalType

int32 __stdcall SetOFDMGuardIntervalType (niWLANGenerationSession session, char channelString[], int32 value);

Purpose

Specifies the type of guard interval (cyclic prefix) in a OFDM symbol.

The following table lists the guard interval length values for different standards.

NIWLANG_STANDARD Attribute Value	Bandwidth (MHz)	NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR length	NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT length	NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_SIXTEEN length
NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM	All	800 nanoseconds	400 nanoseconds	N.A

NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM	All	800 nanoseconds	400 nanoseconds	N.A
NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM	All	8 microseconds	4 microseconds	N.A
NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM	6,7	6 microseconds	3 microseconds	N.A
	8	4.5 microseconds	2.25 microseconds	N.A
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM	All	3.2 microseconds	1.6 microseconds	800 nanoseconds

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
value	int32	Specifies whether to use a long or short guard interval in an OFDM symbol. The default value is NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR.

`NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR` (0)	Specifies that guard interval length is equal to one-fourth of IDFT/DFT period.
`NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT` (1)	Specifies that guard interval length is equal to one-eighth

	of the IDFT/DFT period.
NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_SIXTEEN (1)	Specifies that guard interval length is equal to one-sixteenth of the IDFT/DFT period.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetMappingMatrixType

int32 __stdcall niWLANG_SetMappingMatrixType (niWLANGenerationSession session, char channelString[], int32 mappingMatrixType);

Purpose

Specifies the mapping matrix type for mapping space-time streams to transmit channels as defined in section 20.3.11.10.1 of IEEE Standard 802.11n-2009.

	Note If you set this function to values other than NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the NIWLANG_NUMBER_OF_EXTENSION_SPATIAL_STREAMS attribute if the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
mappingMatrixType	int32	Specifies the mapping matrix type for mapping space-time streams to transmit channels. The default value is NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT. `NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT` (0): Specifies direct mapping for space-time streams. The direct mapping matrix is given by N_{Tx} =

N_{STS}. Direct mapping matrix is derived by taking the subset (N_{STS} * N_{STS}) of the 8 × 8 identity matrix.

where N_{Tx} is the number of transmit channels

N_{STS} is the number of space-time streams. If the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, this value is determined by the modulation and coding scheme (MCS) index and space-time block coding (STBC) index

The following is an example of a 4 × 4 identity matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

NIWLANG_VAL_MAPPING_MATRIX_TYPE_HADAMARD (1)	Specifies Hadamard mapping for space-time streams and the extension spatial streams. The Hadamard mapping matrix is of size N_{Tx} * (N_{STS} + N_{ESS}), where $N_{Tx} \geq$ (N_{STS} + N_{ESS}), if the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM. The Hadamard mapping matrix is of size N_{Tx} * (N_{STS} + N_{ESS}), where $N_{Tx} \geq$ (N_{STS} if the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AC_MIMOOFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM. Hadamard mapping matrix is deriv

	ed by taking the subset ((N_{Tx} x (N_{STS} + N_{ESS})) of the 8 × 8 Hadamard matrix. where N_{Tx} is the number of transmit channels N_{STS} is the number of space-time streams. If the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, the value of N_{STS} is determined by the MCS index and STBC index N_{ESS} is the number of extension spatial streams The following matrix is an example of a 4 × 4 matrix. $\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$
`NIWLANG_VAL_MAPPING_MATRIX_TYP`	Specifies Fourier mapping for space-time str

E_FOURIER (2)	eams and the extension spatial streams. The Fourier mapping matrix is given by $\{a_{nk}\} = \frac{1}{\sqrt{N}} e^{\left[\frac{j2\pi kn}{N}\right]}$ where n= 0 … (N -1) N = N_{Tx}, where N_{Tx} is the number of transmit channels k = 0...((N_{STS} + N_{ESS}) - 1), if the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, or k = 0...((N_{STS} - 1) if the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AC_MIMOOFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, N_{STS} is the nu

	mber of space-time streams. If the NIWLANA_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, this value is determined by MCS index and STBC index, and N_{ESS} is the number of extension spatial streams The following matrices are examples of 2 × 2, 3 × 3 and 4 × 4 matrices. $\frac{1}{\sqrt{2}}\begin{bmatrix} 1+i0 & 1+i0 \\ 1+i0 & 1+i0 \end{bmatrix}$ $\frac{1}{\sqrt{3}}\begin{bmatrix} 1+i0 & 1+i0 & 1+i0 \\ 1+i0 & -0.5+i0.9 & -0.5-i0.9 \\ 1+i0 & -0.5-i0.9 & -0.5+i0.9 \end{bmatrix}$ $\frac{1}{2}\begin{bmatrix} 1+i0 & 1+i0 & 1+i0 & 1+i0 \\ 1+i0 & 0+i1 & -1+i0 & 0-i1 \\ 1+i0 & -1+i0 & 1+i0 & -1+i0 \\ 1+i0 & 0-i1 & -1+i0 & 0+i1 \end{bmatrix}$
`NIWLANG_VAL_MAPPING_MATRIX_TYPE_USER_DEFINED` (3)	Specifies user-defined mapping for space-time streams and extension spatial steams. You can set this matrix using the `niWLANG_SetMappingMatrix` function.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_Set80211ahPreambleType

int32 __stdcall niWLANG_Set80211ahPreambleType (niWLANGenerationSession session, char channelString[], int32 frameFormat);

Purpose

Specifies the preamble type of packet if you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM` as defined in section 24.3.8.2 of IEEE Standard P802.11ah/D1.3.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
frameFormat	int32	Specifies the preamble type to use. The default value is NIWLANG_VAL_80211AH_PREAMBLE_TYPE_SHORT. NIWLANG_VAL_80211AH_PREAMBLE_TYPE_SHORT (0): Specifies that the preamble type is S1G_Short. NIWLANG_VAL_80211AH_PREAMBLE_TYPE_LONG (1): Specifies that the preamble type is S1G_Long.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning

0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetMappingMatrix

int32 __stdcall niWLANG_SetMappingMatrix (niWLANGenerationSession session, char channelString[], NIComplexNumber* mappingMatrix, int32 numMappingMatrixRows, int32 numMappingMatrixColumns);

Purpose

Specifies the matrix for mapping space-time streams to the transmit channels as specified in section 20.3.11.10.1 of IEEE Standard 802.11n-2009. The toolkit ignores this VI, if the `NIWLANG_MAPPING_MATRIX_TYPE` property is not set to User Defined.

If the mappingMatrixType parameter is not set to NIWLANG_VAL_MAPPING_MATRIX_TYPE_USER_DEFINED, the toolkit ignores the niWLANG_SetMappingMatrix function.

The dimensions of the matrix must be N_{Tx} * (N_{STS} + N_{ESS}) if you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`. The dimensions of the matrix must be N_{Tx} * N_{STS} if you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM`, or `NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM`.

where N_{Tx} is the number of transmit channels

N_{STS} is the number of space-time streams. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`, N_{STS} determined by the MCS index and the STBC index

N_{ESS} is the number of extension spatial streams

	Note For 802.11n signals, N_STS is determined by the MCS Index and the STBC Index. For

	802.11ac signals, N_STS is determined by the NIWLANG_MCS_INDEX and NIWLANG_STBC_INDEX attributes.

For one-to-one mapping (direct mapping), $N_{Tx} = N_{STS}$ and $N_{ESS} = 0$. For one-to-many mapping (spatial expansion), $N_{Tx} \geq (N_{STS} + N_{ESS})$.

Valid values are any matrix of size N_{Tx} * $(N_{STS} + N_{ESS})$.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, then N-STS value is dependent on the NIWLANG_SPATIAL_MAPPING_MODE attribute.

Spatial Mapping Mode Attribute Value	N STS Value
Common	Maximum number of space time streams across Resource Units (RUs)
User Specific	Number of space time streams for the specified user

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Specifies the channel string. You must use the following channel string formats when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU. Spatial Mapping Mode Attribute Value: Common — Active Channel String Format: ""(empty string)

		User Specific "userx"
mappingMatrix	NIComplexNumber*	Specifies the matrix for mapping spatial streams to the transmit channels.
numMappingMatrixRows	int32	Specifies the number of mapping matrix rows.
numMappingMatrixColumns	int32	Specifies the number of mapping matrix columns.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_SetIdleInterval

int32 __stdcall niWLANG_SetIdleInterval (niWLANGenerationSession session, char channelString[], float64 idleInterval);

Purpose

Specifies the interframe spacing for signal generation. If you set the NIWLANG_RF_BLANKING_ENABLED attribute to NIWLANA_VAL_FALSE, the toolkit places half of the interframe spacing on either side of the burst in the generated waveform. If you set the NIWLANG_RF_BLANKING_ENABLED attribute to NIWLANA_VAL_TRUE, the toolkit places the interframe spacing at the end of the waveform. The waveform contains zeros for the duration of the interframe spacing.

	Note For higher values of idle interval, LabWindows/CVI may run out of memory. For a large idle interval, you can create a small idle waveform and generate the waveform multiple times using scripting.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
idleInterval	float64	Specifies the interframe spacing for signal generation. The default value is 100 µs. Valid values are 0 to 1, inclusive. This value is expressed in seconds.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the

function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetNumberOfFrames

int32 __stdcall niWLANG_SetNumberOfFrames (niWLANGenerationSession session, char channelString[], int32 numberOfFrames);

Purpose

Specifies the number of frames to generate. Each iteration of the niWLANG_CreateWaveformComplexF64 or niWLANG_CreateMIMOWaveformsComplexF64 function generates only one frame along with the idle interval that you specify using the NIWLANG_IDLE_INTERVAL attribute.

If you set the numberOfFrames parameter to a value greater than 1, call the niWLANG_CreateWaveformComplexF64 function or the niWLANG_CreateMIMOWaveformsComplexF64 function in a loop and concatenate the output values from different iterations. If you encounter memory usage issues, download a single frame to the NI RF vector signal generator memory on each iteration.

To generate the required number of frames, wire the NIWLANG_NUMBER_OF_FRAMES attribute to the loop iteration count. You also can use the done parameter of the niWLANG_CreateWaveformComplexF64 or niWLANG_CreateMIMOWaveformsComplexF64 function as a termination signal.

You can use the `niRFSG_AllocateArbWaveform` function to preallocate arb memory and then download the waveform frame-by-frame.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
numberOfFrames	int32	Specifies the number of frames to generate. The default value is 1. Valid values are 1 to 1,000.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success

Positive Values	Warnings
Negative Values	Errors

niWLANG_SetAutoHeadroomEnabled

int32 __stdcall niWLANG_SetAutoHeadroomEnabled (niWLANGenerationSession session, char channelString[], int32 enabled);

Purpose

Specifies whether the toolkit calculates the headroom or uses the value that you specify. For multiframe generation, the toolkit uses the headroom calculated on the first frame to scale the waveform for every frame. NI recommends that you do not set the enabled parameter to `NIWLANG_VAL_TRUE` for multiframe generation because variation of the peak-to-average power ratio (PAPR) across frames may lead to excessive clipping. To avoid excessive clipping, set the enabled parameter to `NIWLANG_VAL_FALSE` and use the default values for the `NIWLANG_HEADROOM` attribute.

Note This function is available only if you set the compatibilityVersion parameter of the `niWLANG_OpenSession` function to `NIWLANG_VAL_COMPATIBILITY_VERSION_030000`, `NIWLANG_VAL_COMPATIBILITY_VERSION_040000`, or `NIWLANG_VAL_COMPATIBILITY_VERSION_050000`.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
enabled	int32	Specifies whether the toolkit calculates the headroom or

uses the value that you specify. The default value is NIWLANG_VAL_TRUE.

NIWLANG_VAL_FALSE (0)	Specifies that the toolkit uses the headroom that you specify in the NIWLANG_HEADROOM attribute.
NIWLANG_VAL_TRUE (1)	Specifies that the toolkit calculates the headroom automatically.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetHeadroom

int32 __stdcall niWLANG_SetHeadroom (niWLANGenerationSession session, char channelString[], float64 headroom);

Purpose

Specifies the headroom for each transmit channel. This value represents the maximum peak-to-average power ratio (PAPR) allowed in the generated signal. The toolkit clips any portion of the signal that exceeds the peak power corresponding to this value. The toolkit ignores the headroom parameter set by this function if you set the enabled parameter of the `niWLANG_SetAutoHeadroomEnabled` function to `NIWLANG_VAL_TRUE`. If you set the enabled parameter to `NIWLANG_VAL_FALSE`, the toolkit uses default values based on the `niWLANG_SetStandard` function.

	Note If you specify a value that is more than the actual PAPR of the signal, there is loss of dynamic range of the digital-to-analog converter (DAC). If you specify a value that is less than the actual PAPR of the signal, the toolkit clips the generated signal.

	Note In toolkit version NIWLANG_VAL_COMPATIBILITY_VERSION_020000, the `NIWLANG_HEADROOM` attribute was called NIWLANG_MAX_EXPECTED_PAPR. To get the same behavior of this attribute as in version NIWLANG_VAL_COMPATIBILITY_VERSION_020000, you must set the `NIWLANG_AUTO_HEADROOM_ENABLED` attribute to NIWLANG_VAL_FALSE in version NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

To understand which active channel strings are required to configure this function, refer to the `NIWLANG_HEADROOM` attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
headroom	float64	Specifies the headroom per transmit channel. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211BG_DSSS`, the default value is 5. Otherwise, the default value is 12. This value is expressed in dB.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetOFDMPacketExtensionThresholds

int32 __stdcall niWLANG_SetOFDMPacketExtensionThresholds (niWLANGenerationSession session, char channelString[], int32 PPET16[], int32 PPET8[], int32 numberOfSpaceTimeStreams[], int32 RUSize[], int32 PPET16ArraySize, int32 PPET8ArraySize, int32 numberOfSpaceTimeStreamsArraySize, int32 RUArraySize);

Purpose

Configures the packet extension (PE) thresholds that determine the nominal packet padding of the 802.11ax signal. This function configures the thresholds for each resource unit (RU) size and each space-time stream of the 802.11ax DUT. You must configure this table based on the nominal packet padding requirements of the DUT.

The PE field of in 802.11ax signal provides additional processing time to the receiver to decode the last symbol. The possible durations of the PE field are 0 microseconds, 4 microseconds, 8 microseconds, 12 microseconds, or 16 microseconds. The PE duration is determined by both the pre-FEC padding factor of the data field, and the nominal packet padding requested by the recipient, and the modulation scheme of the current PPDU. The nominal packet padding as defined by the HE Capabilities element are 0 microseconds, 8 microseconds, and 16 microseconds, which can be specified using this function.

The nominal packet padding is computed by comparing the constellation index and the threshold values. Packet extension device capability threshold values are defined for all RU sizes greater than or equal to 242 tones. No thresholds are defined for an RU size less than 242 tones. The supported constellations are assigned with a unique index value as shown in following table.

Modulation Scheme	Constellation Index Value
BPSK	0
QPSK	1
16-QAM	2
64-QAM	3
256-QAM	4
1024-QAM	5
4096-QAM	6
None	7

The PE can be defined in two modes.

1. Nominal PE mode 8 microseconds: PE duration is 0, 0, 4, and 8 microseconds for pre-FEC padding factor of 1, 2, 3, and 4 respectively.
2. Nominal PE mode 16 microseconds: PE duration is 4, 8, 12, and 16 microseconds for pre-FEC padding factor of 1, 2, 3, and 4 respectively.

The nominal packet padding is calculated from the threshold values and the constellation index for the specified MCS index as following.

- If the constellation is greater than or equal to PPET16, you must apply the value of the maximum PPET16 parameter, or if the constellation is greater than or equal to PPET8, you must apply the value of the maximum **PPET8** parameter; otherwise, there is no packet extension.
- If no PE is required for all constellations, set **PPET8** and **PPET16** to "" (empty array).
- If only the nominal PE 8 microseconds mode is required, set **PPET16** to be "" (empty array), and **PPET8** to be the constellation at which max PE 8 microseconds mode starts.
- If only the nominal PE 16 microseconds mode is required, set **PPET16** to be the constellation at which max PE 16 microseconds mode starts, and **PPET8** to be "" (empty array).

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
PPET16	int32[]	Specifies the 16 microsecond mode PE threshold values. The default value is "" (empty array).
PPET8	int32[]	Specifies the 8 microsecond mode PE threshold values. The default value is "" (empty array).
numberOfSpaceTimeStreams	int32[]	Specifies the number of space-time streams for the corresponding RU size. The default value is "" (empty array).
RUSize	int32[]	Specifies the size of the RU in terms of the number of subcarriers. The default value is "" (empty array).
PPET16ArraySize	int32	Specifies the size of PPET16 array.
PPET8ArraySize	int32	Specifies the size of PPET8 array.
numberOfSpaceTimeStreamsArraySize	int32	Specifies the size of **numberOfSpaceTimeStreams** array.
RUArraySize	int32	Specifies the size of RU size array.

Return Value

Name	Type	Description

status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

Set and Get Attributes

Contains functions to obtain the current value or set a new value to an attribute.

Set Attributes

Contains functions to set a new value to an attribute.

niWLANG_SetScalarAttributeI32

int32 __stdcall niWLANG_SetScalarAttributeI32 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int32 attributeValue);

Purpose

Sets the value of an niWLAN generation 32-bit integer (int32) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of an int32 niWLAN generation scalar attribute.
attributeValue	int32	Specifies the value to which you want to set the attribute.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success

Positive Values	Warnings
Negative Values	Errors

niWLANG_SetScalarAttributeF64

int32 __stdcall niWLANG_SetScalarAttributeF64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, float64 attributeValue);

Purpose

Sets the value of an niWLAN generation 64-bit floating point number (float64) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a float64 niWLAN generation scalar attribute.
attributeValue	float64	Specifies the value to which you want to set the attribute.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or

describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetScalarAttributeI64

int32 __stdcall niWLANG_SetScalarAttributeI64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int64 attributeValue);

Purpose

Sets the value of an niWLAN generation 64-bit integer (int64) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-

		specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a int64 niWLAN generation scalar attribute.
attributeValue	int64	Specifies the value to which you want to set the attribute.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Success</td></tr><tr><td>Positive Values</td><td>Warnings</td></tr><tr><td>Negative Values</td><td>Errors</td></tr></table>

niWLANG_SetVectorAttributeI32

int32 __stdcall niWLANG_SetVectorAttributeI32 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int32 data[], int32 dataArraySize);

Purpose

Sets the value of an niWLAN generation 32-bit integer (int32) vector attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of an int32 niWLAN generation vector attribute.
data	int32[]	Specifies the int32 array to which you want to set the attribute.
dataArraySize	int32	Specifies the number of elements in the int32 array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the

niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_SetVectorAttributeF64

int32 __stdcall niWLANG_SetVectorAttributeF64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, float64 data[], int32 dataArraySize);

Purpose

Sets the value of an niWLAN generation 64-bit floating point number (float64) vector attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a float64 niWLAN generation vector attribute.

data	float64[]	Specifies the float64 array to which you want to set the attribute.
dataArraySize	int32	Specifies the number of elements in the **data** array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

Get Attributes

Contains functions to obtain the current value of an attribute.

niWLANG_GetScalarAttributeI32

int32 __stdcall niWLANG_GetScalarAttributeI32 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int32* attributeValue);

Purpose

Queries the value of an niWLAN generation 32-bit integer (int32) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of an int32 niWLAN generation scalar attribute.

Output

Name	Type	Description
attributeValue	int32*	Returns the value of the attribute that you specify using the attributeID parameter.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the

		niWLANG_GetErrorString function. The general meaning of the status code is as follows: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Success</td></tr><tr><td>Positive Values</td><td>Warnings</td></tr><tr><td>Negative Values</td><td>Errors</td></tr></table>

niWLANG_GetScalarAttributeF64

int32 __stdcall niWLANG_GetScalarAttributeF64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, float64* attributeValue);

Purpose

Queries the value of an niWLAN generation 64-bit floating point number (float64) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a float64 niWLAN generation scalar attribute.

Output

Name	Type	Description
attributeValue	float64*	Returns the value to which you want to set the attribute.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0: Success Positive Values: Warnings Negative Values: Errors

niWLANG_GetScalarAttributeI64

int32 __stdcall niWLANG_GetScalarAttributeI64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int64* attributeValue);

Purpose

Queries the value of an niWLAN Generation 64-bit integer (int64) scalar attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a int64 niWLAN generation scalar attribute.

Output

Name	Type	Description
attributeValue	int64*	Returns the value to which you want to set the attribute.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_GetVectorAttributeI32

int32 __stdcall niWLANG_GetVectorAttributeI32 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, int32 data[], int32 dataArraySize, int32* actualNumDataArrayElements);

Purpose

Queries the value of an niWLAN generation 32-bit integer (int32) vector attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of an int32 niWLAN generation vector attribute.
dataArraySize	int32	Specifies the number of elements in the data array.

Output

Name	Type	Description

data	int32[]	Returns the current values of an int32 vector attribute. The array must have at least as many elements as are indicated in the dataArraySize parameter.
actualNumDataArrayElements	int32*	Returns the number of elements populated in the data array attribute. If the array is not large enough to hold all the samples, the function returns an error and actualNumDataArrayElements parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

niWLANG_GetVectorAttributeF64

int32 __stdcall niWLANG_GetVectorAttributeF64 (niWLANGenerationSession session, char channelString[], niWLANG_Attr attributeID, float64 data[], int32 dataArraySize, int32* actualNumDataArrayElements);

Purpose

Queries the value of an niWLAN generation 64-bit floating point number (float64) array attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	If the attribute is channel-specific, this parameter specifies the channel to which the attribute applies. If the attribute is not channel-specific, set this parameter to "" (empty string) or NULL.
attributeID	niWLANG_Attr	Specifies the ID of a float64 niWLAN generation vector attribute.
dataArraySize	int32	Specifies the number of elements in the data array.

Output

Name	Type	Description
data	float64[]	Returns the current value of a float64 vector attribute. The array must have at least as many elements as are indicated in the dataArraySize parameter.

actualNumDataArrayElements	int32*	Returns the actual number of elements populated in the data array parameter. If the array is not large enough to hold all the samples, the function returns an error and actualNumDataArrayElements parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGConfigureMultipleDeviceSynchronizatio

int32 __stdcall niWLANG_RFSGConfigureMultipleDeviceSynchronization (niWLANGenerationSession session, ViSession rfsgSessions[], int32 noOfChannels, char masterReferenceClockSource[], int32 triggerLines[], int32 noOfTriggerLines);

Purpose

Configures multiple NI RF vector signal generators for sharing the local oscillator (LO), configures Reference Clock settings, and synchronizes multiple devices.

For PXIe-5673/5673E/5840/5841/5830/5831 RF vector signal generators, this function configures LO sharing and clock settings in daisy-chained manner.

For PXIe-5644, PXIe-5645, and PXIe-5646 RF vector signal transceiver (VST) devices, this function configures daisy-chained LO sharing and low-level properties for multiple device synchronization. These VSTs do not support daisy-chained Reference Clocks for synchronization. You must set the **MasterReferenceClockSource** to PXI_CLK for these devices.

For PXIe-5840/5841/5830/5831 RF VST devices, this VI configures daisy-chained LO sharing and Reference Clocks for synchronization, and for PXIe-5820 baseband I/Q VST devices, this VI configures daisy-chained Reference Clocks for synchronization. For additional information, refer to Synchronization Using NI-RFSA and NI-RFSG topic in the NI RF Signal Generators Help.

This function assumes that the devices are interconnected in the order of elements in the instrument handles array. In time synchronization, the first device is assumed to be the master device, and the remaining devices are assumed to be slave devices. The devices are divided into two sets when the number of segments are two. In LO sharing, the first device in the each set is assumed to be the master device, and the remaining devices in the set are assumed to be slave devices. To interconnect multiple devices, refer to the following topics in the NI RF Signal Generators Help:

- Interconnecting Multiple PXIe-5673E Modules
- Interconnecting Multiple PXIe-5673 Modules
- Interconnecting Multiple PXIe-5644 RF Channels (Homogeneous Channel Types)

- Interconnecting Multiple PXIe-5645 RF Channels (Homogeneous Channel Types)
- Interconnecting Multiple PXIe-5646 RF Channels (Homogeneous Channel Types)
- Interconnecting Multiple PXIe-5840 RF Channels (Homogeneous Channel Types)
- Interconnecting Multiple PXIe-5841 RF Channels (Homogeneous Channel Types)

If you use the PXIe-5644, PXIe-5645, or PXIe-5646 RF vector signal transceiver (VST) devices, the function completes the following actions:

- Sets the `NIRFSG_ATTR_REF_CLOCK_SOURCE` attribute to the value you specify in **masterReferenceClockSource** parameter.
- If you set the `NIWLANG_LO_SHARING_ENABLED` attribute to `NIWLANG_VAL_TRUE`, the function completes the following actions:
  - For the master device, the function exports the LO by setting the `NIRFSG_ATTR_LO_EXPORT_ENABLED` attribute to `VI_TRUE`. The function also reads the `NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY` attribute so that the value can be set on slave devices.
  - For the slave devices, the function sets the `NIRFSG_ATTR_LO_SOURCE` attribute to `NIRFSG_VAL_LO_IN_STR` and the `NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY` attribute to the value read from the master device. With the exception of the last slave device, the function also exports the LO to the next device in the daisy chain by setting the `NIRFSG_ATTR_LO_EXPORT_ENABLED` attribute to `VI_TRUE`.
- The function configures the trigger synchronization for the master and slave devices.
- For the master device, the function completes the following actions:
  - Sets the `NIRFSG_ATTR_SYNC_START_TRIGGER_MASTER` attribute and the `NIRFSG_ATTR_SYNC_REF_TRIGGER_MASTER` attribute to `VI_TRUE`. For the PXIe-5646, it sets `NIRFSG_ATTR_SYNC_SAMPLE_CLOCK_MASTER` attribute to `VI_TRUE`.

- Sets the NIRFSG_ATTR_SYNC_START_TRIGGER_DIST_LINE attribute and the NIRFSG_ATTR_SYNC_REF_TRIGGER_DIST_LINE attribute to the first and second element of the **triggerLines** array, respectively. For the PXIe-5646, it sets the NIRFSG_ATTR_SYNC_SAMPLE_CLOCK_DIST_LINE attribute to the third element of the **triggerLines** array.
- Calls the niRFSG_Commit function to commit the synchronization settings to the device.

- For the slave devices, the function completes the following actions:
  - Sets the NIRFSG_ATTR_SYNC_START_TRIGGER_MASTER attribute and NIRFSG_ATTR_SYNC_REF_TRIGGER_MASTER attribute to VI_FALSE. For the PXIe-5646, it sets NIRFSG_ATTR_SYNC_SAMPLE_CLOCK_MASTER attribute to VI_FALSE.
  - Sets the NIRFSG_ATTR_SYNC_START_TRIGGER_DIST_LINE attribute and the NIRFSG_ATTR_SYNC_REF_TRIGGER_DIST_LINE attribute to the first and second element of the **triggerLines** array, respectively. For the PXIe-5646, it sets the NIRFSG_ATTR_SYNC_START_TRIGGER_DIST_LINE attribute to the third element of the **triggerLines** array.
  - Sets the NIRFSG_ATTR_REF_TRIGGER_TYPE attribute to NIRFSG_VAL_DIGITAL_EDGE, the NIRFSG_ATTR_DIGITAL_EDGE_REF_TRIGGER_SOURCE attribute to NIRFSG_VAL_SYNC_REF_TRIGGER_STR, and the NIRFSG_ATTR_DIGITAL_EDGE_START_TRIGGER_SOURCE attribute to NIRFSG_VAL_SYNC_START_TRIGGER_STR.

If you use the PXIe-5673/5673E RF vector signal analyzers, the function completes the following actions:

- Configures the devices for daisy-chained Reference Clock synchronization.
  - For the master device, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to the value you specify in **masterReferenceClockSource** and the NIRFSG_ATTR_EXPORTED_RE

F_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_CLK_OUT_STR.

- For the slave devices, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to NIRFSG_VAL_CLK_IN_STR and the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_CLK_OUT_STR. The function also sets the NIRFSG_ATTR_REF_CLOCK_RATE attribute according to the device model.
- If NIWLANG_LO_SHARING_ENABLED attribute is set to NIWLANG_VAL_TRUE, the function completes the following actions:
    - For the master device, it exports the LO by setting the NIRFSG_ATTR_LO_EXPORT_ENABLED attribute to VI_TRUE. The function also reads the NIRFSG_ATTR_LO_FREQUENCY attribute so that the value can be set on slave devices.
    - For the slave devices, the function sets the NIRFSG_ATTR_LO_SOURCE attribute to NIRFSG_VAL_LO_IN_STR, the NIRFSG_ATTR_LO_FREQUENCY attribute to the value read from the master and exports the LO to the next device in the daisy chain by setting the NIRFSG_ATTR_LO_EXPORT_ENABLED attribute to VI_TRUE.

If you use PXIe-5840, PXIe-5841, PXIe-5830 or PXIe-5831 RF vector signal generators, the function completes the following actions:

- For reference clock synchronization:
    - The function configures all the devices to use PXI_CLK as reference clock if you set the **MasterReferenceClockSource** parameter to NIRFSA_VAL_PXI_CLK_STR.
    - The function configures the devices for daisy-chained reference clock synchronization, if you set the **MasterReferenceClockSource** parameter to a value other than NIRFSA_VAL_PXI_CLK_STR.
        - For the master device, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to the value you specify in the **MasterReferenceClockSource** parameter and sets the NIRFSG_AT

TR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT_STR.

- For the slave devices, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to NIRFSG_VAL_REF_IN_STR. Except for the last slave device, this function also sets the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT_STR.

- If you set the NIWLANG_LO_SHARING_ENABLED attribute to NIWLANG_VAL_LO_SHARING_ENABLED_TRUE, the function completes the following actions:
  - For the master device, the function exports the LO by setting the NIRFSG_ATTR_LO_OUT_ENABLED attribute to VI_TRUE. The function also reads the NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY attribute so that the value can be set on slave devices.
  - For the slave devices, the function sets the NIRFSG_ATTR_LO_SOURCE attribute to NIRFSG_VAL_LO_IN_STR and the NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY attribute to the value read from the master device. Except for the last slave device, this function also exports the LO to the next device in the daisy chain by setting the NIRFSG_ATTR_LO_OUT_ENABLED attribute to VI_TRUE.

If you use PXIe-5820 baseband I/Q vector signal generators, the function completes the following actions:

- For reference clock synchronization:
  - The function configures all the devices to use PXI_CLK as reference clock if you set the **MasterReferenceClockSource** parameter to NIRFSA_VAL_PXI_CLK_STR.
  - The function configures the devices for daisy-chained reference clock synchronization, if you set the **MasterReferenceClockSource** parameter to a value other than NIRFSA_VAL_PXI_CLK_STR.
    - For the master device, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to the value you specify in the

MasterReferenceClockSource parameter and sets the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT.

- For the slave devices, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to NIRFSG_VAL_REF_IN. Except for the last slave device, this function also sets the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT.

If you use PXIe-5830 or PXIe-5831 RF vector signal generators, the function completes the following actions:

- For reference clock synchronization:
  - The function configures all the devices to use PXI_CLK as reference clock if you set the **MasterReferenceClockSource** parameter to NIRFSA_VAL_PXI_CLK_STR.
  - The function configures the devices for daisy-chained reference clock synchronization, if you set the **MasterReferenceClockSource** parameter to a value other than NIRFSA_VAL_PXI_CLK_STR.
    - For the master device, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to the value you specify in the **MasterReferenceClockSource** parameter and sets the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT_STR.
    - For the slave devices, the function sets the NIRFSG_ATTR_REF_CLOCK_SOURCE attribute to NIRFSG_VAL_REF_IN_STR. Except for the last slave device, this function also sets the NIRFSG_ATTR_EXPORTED_REF_CLOCK_OUTPUT_TERMINAL attribute to NIRFSG_VAL_REF_OUT_STR.
- If you set the NIWLANG_LO_SHARING_ENABLED attribute to NIWLANG_VAL_LO_SHARING_ENABLED_TRUE, the function completes the following actions:

- For the master device, the function exports the LO by setting the `NIRFSG_ATTR_LO_OUT_ENABLED` attribute to `VI_TRUE`. The function also reads the `NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY` attribute so that the value can be set on slave devices.
- For the slave devices, the function sets the `NIRFSG_ATTR_LO_SOURCE` attribute to `NIRFSG_VAL_LO_IN_STR` and the `NIRFSG_ATTR_UPCONVERTER_CENTER_FREQUENCY` attribute to the value read from the master device. Except for the last slave device, this function also exports the LO to the next device in the daisy chain by setting the `NIRFSG_ATTR_LO_OUT_ENABLED` attribute to `VI_TRUE`.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Specifies an array of references to multiple NI-RFSG instrument sessions. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions function and identifies a particular instrument session.
noOfChannels	int32	Specifies the number of Transmit chains. The value of this parameter should be product of the values configured using the `NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS` and `NIWLANG_NUMBER_OF_SEGMENTS` attributes.
masterReferenceClockSource	char[]	Specifies the device Reference Clock to configure on the master NI RF vector signal generator.

triggerLines	int32[]	Specifies trigger lines used for distribution of synchronized trigger signals.
noOfTriggerLines	int32	Specifies the actual number of elements populated in the **triggerLines** array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGConfigureFrequencySingleLO

int32 __stdcall niWLANG_RFSGConfigureFrequencySingleLO (niWLANGenerationSession session, ViSession rfsgSessions[], int32 numberOfRFSGSessions, int32 LOSource, ViSession externalLOHandle, float64 carrierFrequency, int32 rfsgLODaisyChainEnabled, int32 LOExportToExternalDevicesEnabled);

Purpose

Configures the frequency on NI RF vector signal generators and NI RF synthesizers. NI RF synthesizers are used as external local oscillator (LO) devices. This function also configures LO frequency offset based on the NIWLANG_LO_FREQUENCY_OFFSET_MODE and NIWLANG_LO_FREQUENCY_OFFSET attributes. It also configures the Signal Bandwidth and IQ Rate attributes on NI RF vector signal generators and the NIWLANG_CARRIER_FREQUENCY attribute.

You must ensure that after a frequency configuration change, the output of the external LO device settles before generating the WLAN signal. You must use one of the following programming flows:

- Before calling the niWLANG_RFSGConfigureFrequencySingleLO function, if the external LO device is in generation state, ensure that it is brought in configuration state, by stopping signal generation. Initiate signal generation on the external LO device after calling the niWLANG_RFSGConfigureFrequencySingleLO function.
- You may choose to do on-the-fly frequency change on the external LO device while the device is in generation state. You must allow it's output to settle by calling the niRFSG_WaitUntilSettled function after the niWLANG_RFSGConfigureFrequencySingleLO function.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions

		functions and identifies a particular instrument session.
numberOfRFSGSessions	int32	Specifies the number of Transmit chains. The value of this parameter should be product of the values configured using the `NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS` and `NIWLANG_NUMBER_OF_SEGMENTS` attributes.
LOSource	int32	Specifies whether to use the internal or the external LO source. This value is applicable only for the first RFSG device if the **rfsgLODaisyChainEnabled** parameter is set to TRUE. The default value is NIWLANG_VAL_LO_SOURCE_ONBOARD. `NIWLANG_VAL_LO_SOURCE_ONBOARD` (0): Uses an internal LO as the LO source. `NIWLANG_VAL_LO_SOURCE_EXTERNAL` (1): Uses an external LO as the LO source. `NIWLANG_VAL_LO_SOURCE_SG_SA_SHARED` (2): Shares the internal RFSG LO between RFSG and RFSA.
externalLOHandle	ViSession	Identifies the instrument session of the external LO device. This parameter is obtained from the niRFSG_Initialize function or the

		niRFSG_InitializeWithOptions function.
carrierFrequency	float64	Specifies the carrier frequency used to generate signals. This value is expressed in Hz.
rfsgLODaisyChainEnabled	int32	Specifies whether to export the LO signal from one RFSG device to the next. The default value is FALSE.
LOExportToExternalDevicesEnabled	int32	Specifies whether to export the LO signal from each RFSG device on its LO OUT terminal, which you can use to share the LO signal with an external device. An example of an external device would be an RFSA device. The default value is FALSE.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values	Errors

niWLANG_RFSGConfigureFrequencyMultipleLO

int32 __stdcall niWLANG_RFSGConfigureFrequencyMultipleLO (niWLANGenerationSession session, ViSession rfsgSessions[], int32 numberOfRFSGSessions, int32 LOSource, ViSession externalLOHandles[], int32 numberOfexternalLOHandles, float64 carrierFrequency[], int32 dataArraySize, int32 rfsgLODaisyChainEnabled, int32 LOExportToExternalDevicesEnabled);

Purpose

Configures the frequency on NI RF vector signal generators and NI RF synthesizers. NI RF synthesizers are used as external LO devices. This function also configures LO frequency offset based on the NIWLANG_LO_FREQUENCY_OFFSET_MODE and NIWLANG_LO_FREQUENCY_OFFSET attributes. This function equally divides the NI RF vector signal generators and NI RF synthesizers into sets such that each set corresponds to one carrier frequency. The number of sets is equal to the size of the carrier frequencies array. In each set, the first NI RF vector signal generator is used as the master device for LO daisy chaining. It also configures the Signal Bandwidth and IQ Rate attributes on NI RF vector signal generators and the NIWLANG_CARRIER_FREQUENCY attribute.

You must ensure that after a frequency configuration change, the output of the external LO device settles before generating the WLAN signal. You must use one of the following programming flows:

- Before calling the niWLANG_RFSGConfigureFrequencyMultipleLO function, if the external LO device is in generation state, ensure that it is brought in configuration state, by stopping signal generation. Initiate signal generation on the external LO device after calling the niWLANG_RFSGConfigureFrequencyMultipleLO function.
- You may choose to do on-the-fly frequency change on the external LO device while the device is in generation state. You must allow it's output to

settle by calling the niRFSG_WaitUntilSettled function after the niWLANG_RFSGConfigureFrequencyMultipleLO function.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
numberOfRFSGSessions	int32	Specifies the number of Transmit chains. The value of this parameter should be product of the values configured using the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS and NIWLANG_NUMBER_OF_SEGMENTS attributes.
LOSource	int16	Specifies whether to use the internal or the external LO source. This value is applicable only for the first RFSG device within a set if the **rfsgLODaisyChainEnabled** parameter is set to TRUE. The default value is NIWLANG_VAL_LO_SOURCE_ONBOARD. NIWLANG_VAL_LO_SOURCE_ONBOARD (0): Uses an internal LO as the LO source.

		NIWLANG_VAL_LO_SOURCE_EXTERNAL (1) — Uses an external LO as the LO source. NIWLANG_VAL_LO_SOURCE_SG_SA_SHARED (2) — Shares the internal RFSG LO between RFSG and RFSA.
externalLOHandles	ViSession[]	Identifies the instrument sessions of external LO devices. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions.
numberOfexternalLOHandles	int32	Specifies the number of external LO handles.
carrierFrequency	float64[]	Specifies an array of the carrier frequencies used to generate signals. This value is expressed in Hz.
dataArraySize	int32	Specifies the number of elements in the **carrierFrequency** array. This value should be the same as the value you set for the NIWLANG_NUMBER_OF_SEGMENTS attribute.
rfsgLODaisyChainEnabled	int32	Specifies whether to export the LO signal from one RFSG device to the next. The default value is FALSE.
LOExportToExternalDevicesEnabled	int32	Specifies whether to export the LO signal from each RFSG device on its LO OUT terminal, which you can use to share the LO signal with an external device. An example of an external device would be an

		RFSA device. The default value is FALSE.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGMultipleDeviceInitiate

int32 __stdcall niWLANG_RFSGMultipleDeviceInitiate (niWLANGenerationSession session, ViSession rfsgSessions[], int32 numberOfRFSGSessions, int32 forceSync);

Purpose

Commits settings to hardware, waits for hardware settling, and starts multi-device synchronized generation.

If you use PXIe-5644, PXIe-5645, or PXIe-5646, this function performs the following steps:

- Commits the device settings to the master device by calling the niRFSG Commit function.
- Initializes generation first for slave devices and then for the master device by calling the niRFSG Initiate function in a loop.

If you use PXIe-5673/5673E, PXIe-5840, PXIe-5841, PXIe-5820, PXIe-5830, or PXIe-5831, this function performs the following steps:

- If number of devices required is more than 1, this function uses the niWLANG RFSG Configure TClk For Homogeneous Triggers function, niWLANG RFSG Synchronize TClk function, and the niTClk Initiate function for synchronized generation. Refer to the NI-TClk Synchronization section of NI RF Vector Signal Generators help for more information about the niTClk functions.
- Otherwise, this function calls the niRFSG Initiate function for the first device.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
numberOfRFSGSessions	int32	Passes a reference of the WLAN generation session to the next function. Close the niWLAN generation session reference using the niWLANG_CloseSession function before the completion

		of execution to avoid possible memory leak issues.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

niWLANG_CreateWaveformComplexF64

int32 __stdcall niWLANG_CreateWaveformComplexF64 (niWLANGenerationSession session, int32 reset, int32 waveformSize, float64* t0, float64* dt, NIComplexNumber waveform[], int32* actualNumWaveformSamples, int32* done);

Purpose

Creates WLAN I/Q data and returns the data as a complex waveform. This function returns one frame, including the idle interval, at a time. For multi-frame generation, set the reset parameter to `NIWLANG_VAL_FALSE` and run the function in a loop for the specified number of times or until the done parameter is NIWLANG_VAL_TRUE.

	Note Use this function if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
reset	int32	Specifies whether to reset the internal states in the created waveform. Set this parameter to NIWLANG_VAL_TRUE for the first frame of the generation or if you want to reset the pseudonoise (PN) seed.
waveformSize	int32	Specifies the waveform size in samples. The size of the waveform is given by the NIWLANG_IQ_WAVEFORM_SIZE attribute.

Output

Name	Type	Description
t0	float64*	Returns the starting time. This value is expressed in seconds.
dt	float64*	Returns the time interval between baseband I/Q samples. This value is expressed in seconds.

waveform	NIComplexNumber[]	Returns the WLAN I/Q data. This parameter must be at least the size of the waveformSize parameter.
actualNumWaveformSamples	int32*	Returns the actual number of samples populated in the waveform array. If the array is not large enough to hold all the samples, the function returns an error and this parameter returns the minimum expected size of the output array.
done	int32*	Indicates whether the function has generated all the data. If you generate multiple frames, you can include this function in a while loop and use the done parameter as the terminating condition.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success

Positive Values	Warnings
Negative Values	Errors

niWLANG_CreateMIMOWaveformsComplexF64

int32 __stdcall niWLANG_CreateMIMOWaveformsComplexF64 (niWLANGenerationSession session, int32 reset, int32 numberOfTxChains, int32 individualWaveformSize, float64 t0[], float64 dt[], NIComplexNumber *waveforms, int32* actualNumSamplesInEachWfm, int32* done);

Purpose

Creates WLAN I/Q data for multiple channels and returns the data as an array of complex waveforms. This function returns one frame, including the idle interval, at a time. For multiframe generation, set the reset parameter to `NIWLANG_VAL_FALSE` and run the function in a loop for the specified number of times or until the done parameter is `NIWLANG_VAL_TRUE`.

	Note Use this function if you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM`, `NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM`, or `NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM`.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
reset	int32	Specifies whether to reset the internal states in the created waveform. Set this parameter to `NIWLANG_VAL_TRUE` for

		the first frame of the generation or if you want to reset the pseudonoise (PN) seed.
numberOfTxChains	int32	Specifies the number of Transmit chains. The value of this parameter should be product of the values configured using the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS and NIWLANG_NUMBER_OF_SEGMENTS attributes.
individualWaveformSize	int32	Specifies the size of the waveform per WLAN channel. The size of the waveform is given by the NIWLANG_IQ_WAVEFORM_SIZE attribute.

Output

Name	Type	Description
t0	float64[]	Returns the starting time The size of this array must be at least equal to value of the numberOfTxChains parameter. This value is expressed in seconds.
dt	float64[]	Returns the time interval between baseband I/Q samples. The size array must be at least equal to value of the numberOfTxChains parameter. This value is expressed in seconds.
waveforms	NIComplexNumber*	Returns the WLAN I/Q data. The waveforms are written sequentially in the array. Allocate an array at least as large as numberOfTxChains times individualWaveformSize for this parameter.
actualNumSamplesInEachWfm	int32*	Returns the actual number of samples for each WLAN channel

		waveform. If the array is not large enough to hold all the samples, the function returns an error and this parameter returns the minimum expected size of the output array.
done	int32*	Indicates whether the function has generated all data. If you generate multiple frames, you can include this function in a while loop and use the done parameter as the terminating condition.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

Generation Information

Contains the function to get generator parameters used to generate a waveform.

niWLANG_GetErrorString

int32 __stdcall niWLANG_GetErrorString (niWLANGenerationSession session, int32 errorCode, char errorMessage[], int32 errorMessageLen);

Purpose

Takes the error code returned by niWLAN generation functions and returns the interpretation as a user-readable string.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
errorCode	int32	Specifies the error code that is returned from any of the niWLAN generation functions.
errorMessageLen	int32	Specifies the length of the errorMessage buffer.

Output

Name	Type	Description
errorMessage	char[]	Returns the user-readable message string that corresponds to the error code you specify. The errorMessage buffer must have at least as many elements as are indicated in the errorMessageLen parameter. If you pass NULL to the errorMessage parameter,

		the function returns the actual length of the error message.

Return Value

Name	Type	Description
status	int32	Returns the number of characters written in errorMessage buffer.

niWLANG_GetNumberOfUsersFromRUAllocation

int32 __stdcall niWLANG_GetNumberOfUsersFromRUAllocation(niWLANGenerationSession session, char channelString[], int32* numberOfUsers);

Purpose

Returns the number of users derived from the configured RU allocation in a 802.11ax MU PPDU signal. To use this function, you must set the `NIWLANG_RU_ALLOCATION_MODE` attribute to `NIWLANG_VAL_RU_ALLOCATION_MODE_GROUP` and configure the `NIWLANG_RU_ALLOCATION` attribute.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
numberOfUsers	int32*	Returns the number of users in a 802.11ax MU PPDU signal.

Return Value

Name	Type	Description

status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_GetMappingMatrix

int32 __stdcall niWLANG_GetMappingMatrix (niWLANGenerationSession session, char channelString[], NIComplexNumber* mappingMatrix, int32 numMappingMatrixRows, int32 numMappingMatrixColumns);

Purpose

Returns the matrix used for mapping space-time streams to the transmit channels as defined in section 20.3.11.10.1 of IEEE Standard 802.11n-2009.

The dimensions of the matrix must be N_{TX} * (N_{STS} + N_{ESS}) if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM. The dimensions of the matrix must be N_{TX} * N_{STS} if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL

`_STANDARD_80211AH_MIMO_OFDM`, or `NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM`.

where N_{Tx} is the number of transmit channels

N_{STS} is the number of space-time streams. If you set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM`, N_{STS} determined by the MCS index and the STBC index

N_{ESS} is the number of extension spatial streams

	Note For 802.11n signals, N_STS is determined by the MCS Index and the STBC Index. For 802.11ac signals, N_STS is determined by the NIWLANG_MCS_INDEX and NIWLANG_STBC_INDEX attributes.

For one-to-one mapping (direct mapping), $N_{Tx} = N_{STS}$ and $N_{ESS} = 0$. For one-to-many mapping (spatial expansion), $N_{Tx} \geq (N_{STS} + N_{ESS})$.

Valid values are any matrix of size $N_{Tx} * (N_{STS} + N_{ESS})$.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, then N-STS value is dependent on the NIWLANG_SPATIAL_MAPPING_MODE attribute.

Spatial Mapping Mode Attribute Value	N STS Value
Common	Maximum number of space time streams across Resource Units (RUs)
User Specific	Number of space time streams for the specified user

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Specifies the channel string. You must use the following

		channel string formats when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU. Spatial Mapping Mode Attribute Value / Active Channel String Format: Common — ""(empty string) User Specific — "userx"
mappingMatrix	NIComplexNumber*	Returns the matrix for mapping spatial streams to transmit channels.
numMappingMatrixRows	int32	Specifies the number of mapping matrix rows.
numMappingMatrixColumns	int32	Specifies the number of mapping matrix columns.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning

0	Success
Positive Values	Warnings
Negative Values	Errors

Utility

Contains functions to reset all attributes, obtain the error code returned by WLAN Generation functions, save and load configuration files, and calculate carrier frequencies according to the IEEE 802.11 channel numbering scheme.

niWLANG_LoadConfigurationFromFile

int32 __stdcall niWLANG_LoadConfigurationFromFile (niWLANGenerationSession session, char filePath[], int32 reset);

Purpose

Loads the attributes of a session saved in a file.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
filePath	char[]	Specifies the complete path to the file from which the toolkit loads the configuration.
reset	int32	Specifies whether to reset all the attributes of the session before loading the settings from a file. The default value is `NIWLANG_VAL_TRUE`.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_SaveConfigurationToFile

int32 __stdcall niWLANG_SaveConfigurationToFile (niWLANGenerationSession session, char filePath[], int32 operation);

Purpose

Saves all attributes of the session to a file located at a specified path. Use this file to save the current state of the toolkit.

Parameters

Input

Name	Type	Description

session	niWLANGenerationSession	Specifies the niWLAN generation session.
filePath	char[]	Specifies the complete path to the TDMS file to which the toolkit saves the configuration.
operation	int32	Specifies the operation to perform on the file. The default value is NIWLANG_VAL_FILE_OPERATION_MODE_CREATE_OR_REPLACE. `NIWLANG_VAL_FILE_OPERATION_MODE_OPEN(0)` — Opens an existing file to write the niWLANG settings. `NIWLANG_VAL_FILE_OPERATION_MODE_OPEN_OR_CREATE(1)` — Opens an existing file or creates a new file if the file does not exist. `NIWLANG_VAL_FILE_OPERATION_MODE_CREATE_OR_REPLACE(2)` — Creates a new file or replaces an existing file. `NIWLANG_VAL_FILE_OPERATION_MODE_CREATE(3)` — Creates a new file.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning

condition. Examine the status code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_ChannelNumberToCarrierFrequency

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency (int32 frequencyBand, float64 channelBandwidth, int32 channelNumber, int32 secondaryFactor, float64 channelStartingFactor, float64 *carrierFrequency);

Purpose

Calculates the carrier frequency according to the numbering scheme by converting a set of input parameters, including the channel number, for IEEE 802.11a/b/g/j/n/p, IEEE 802.11ac, IEEE 802.11ah, IEEE 802.11af, and IEEE P802.11ax/D6.0 standards.

Parameters

Input

Name	Type	Description
frequencyBand	int32	Specifies whether to use the 2.4 GHz or the 5 GHz band. The default value is `NIWLANG_VAL`

		`_FREQUENCY_BAND_2p4GHZ`. `NIWLANG_VAL_FREQUENCY_BAND_2p4GHZ` (0) — Specifies a frequency band of 2.4 GHz. `NIWLANG_VAL_FREQUENCY_BAND_5GHZ` (1) — Specifies a frequency band of 5 GHz.
channelBandwidth	float64	Specifies the channel bandwidth Valid values are 5 MHz, 10 MHz, 20 MHz, and 40 MHz. This value is expressed in Hz.
channelNumber	int32	Specifies the offset of the center frequency above the starting frequency of the channel. When you set the channelBandwidth parameter to 40 MHz, the channelNumber parameter is the primary channel number and the corresponding channel center frequency is the primary channel center frequency. This value is expressed in increments of 5 MHz.
secondaryFactor	int32	Specifies whether the secondary channel is above or below the primary channel when you set the channelBandwidth parameter to 40 MHz. The toolkit creates a 40 MHz channel by combining the primary channel and the secondary channel, where the

		channels have a bandwidth of 20 MHz. The secondary channel number is given by the following formula: secondary channel number = primary channel number + (4 * secondary factor) The secondary channel center frequency is given by the following formula: secondary channel center frequency (Hz) = channel starting frequency (Hz) + (secondary channel number * 5 MHz) Valid values are -1 and +1.
channelStartingFactor	float64	Specifies the value used to define the baseline frequency. The channel start frequency is given by the following formula: channel starting frequency (Hz) = channel starting factor * 500 kHz)

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz. This function calculates the carrier frequency using the following equation: Carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 5 MHz).

Return Value

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>status</td><td>int32</td><td>Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Success</td></tr>
<tr><td>Positive Values</td><td>Warnings</td></tr>
<tr><td>Negative Values</td><td>Errors</td></tr>
</table></td></tr>
</table>

niWLANG_ChannelNumberToCarrierFrequency80211ab

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency80211abgjpn (float64 channelStartingFrequency, float64 channelBandwidth, int32 channelNumber, int32 secondaryFactor, float64 *carrierFrequency);

Purpose

Calculates the carrier frequency of 802.11a/b/g/j/p/n channels according to sections 16.4.6, 17.4.6, 18.3.8.4, and 20.3.15 of IEEE Standard 802.11-2012.

Parameters

Input

Name	Type	Description
channelStartingFrequency	float64	Specifies the starting frequency of the frequency band. This value is expressed in Hz.
channelBandwidth	float64	Specifies the channel bandwidth You can choose a 5 MHz, 10 MHz, 20 MHz, or 40 MHz channel. This value is expressed in Hz.
channelNumber	int32	Specifies the offset of the center frequency, in increments of 5 MHz, above the starting frequency of the channel.
secondaryFactor	int32	Specifies whether the secondary channel is above or below the primary channel when you set the channelBandwidth parameter to 40 MHz. The toolkit creates a 40 MHz channel by combining the primary channel and the secondary channel, each with a 20 MHz bandwidth. The value of -1 indicates that the secondary channel is below the primary channel whereas the value of +1 indicates that the secondary channel is above the primary channel. Valid values are -1 and +1.

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz. The function calculates the carrier frequency using the following equation:

		Carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 5 MHz) When you set the channelBandwidth parameter to 40 MHz, the channelNumber parameter is the primary channel number. The function calculates the carrier frequency using the following equation: carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 5 MHz)+ (Secondary factor * 20 MHz)

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Value s	Errors

niWLANG_ChannelNumberToCarrierFrequency80211ac

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency80211ac (float64 channelStartingFrequency, int32 channelNumber, float64 *carrierFrequency);

Purpose

Calculates carrier frequency of 802.11ac channels according to section 22.3.14 of IEEE Standard 802.11ac-2013.

Parameters

Input

Name	Type	Description
channelStartingFrequency	float64	Specifies the start frequency of the frequency band. This value is expressed in Hz.
channelNumber	int32	Specifies the offset of the center frequency above the starting frequency of the channel. This value is expressed in increments of 5 MHz.

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz. The function calculates the carrier frequency using the following equation: Carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 5 MHz)

Return Value

Name	Type	Description

status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

niWLANG_ChannelNumberToCarrierFrequency80211ah

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency80211ah (float64 channelStartingFrequency, int32 channelNumber, float64 *carrierFrequency);

Purpose

Calculates the carrier frequency of 802.11ah channels according to section 22.3.13 of IEEE P802.11ah/D1.3.

Parameters

Input

Name	Type	Description
channelStartingFrequency	float64	Specifies the start frequency of the frequency band. This value is expressed in Hz.

channelNumber	int32	Specifies the offset of the center frequency above the starting frequency of the channel. This value is expressed in increments of 5 MHz.

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz. The function calculates the carrier frequency using the following formula: carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 0.5 MHz)

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values	Errors

niWLANG_ChannelNumberToCarrierFrequency80211af

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency80211af (float64 channelStartingFrequency, float64 channelBandwidth, int32 channelNumber, int32 TVHTMode, float64 *carrierFrequency);

Purpose

Calculates the carrier frequency of 802.11af channels, as defined in section 23.3.14 of IEEE Standard 802.11af-2013.

Parameters

Input

Name	Type	Description
channelStartingFrequency	float64	Specifies the start frequency of the frequency band. The default value is 45 MHz. The channel start frequency is given by the following formula: Channel Starting Frequency (Hz) = Channel Starting Factor * 500 kHz This value is expressed in Hz.
channelBandwidth	float64	Specifies the channel bandwidth used for transmitting the signal. The default value is 6 MHz. This value is expressed in Hz.
channelNumber	int32	Specifies the offset of the center frequency, in increments of the channel bandwidth used for transmitting the signal. This

		value is expressed in Hz. The default value is 1 MHz.
TVHTMode	int32	Specifies the mode of the 802.11af signal transmission. The default value is `NIWLANG_VAL_TVHT_MODE_4`.

`NIWLANG_VAL_TVHT_MODE_1` (0)	Specifies the TVHT mode representing a single Basic Channel Unit (BCU).
`NIWLANG_VAL_TVHT_MODE_2C` (1)	Specifies the TVHT mode representing two contiguous BCUs.
`NIWLANG_VAL_TVHT_MODE_2N` (2)	Specifies the TVHT mode representing two noncontiguous BCUs.
`NIWLANG_VAL_TVHT_MODE_4C` (3)	Specifies the TVHT mode representing four contiguous BCUs.
`NIWLANG_VAL_TVHT_MODE_4N` (4)	Specifies the TVHT mode representing two noncontiguous frequency segments, each of which are composed of two BCUs.

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz.

The function calculates the carrier frequency using the following equation:

carrier frequency (Hz) = channel starting frequency (Hz) + (TVHT_W * channel number + channel center frequency correction (Hz)

where channel center frequency correction (Hz) is used to adjust the carrier frequency in the different TVHT modes. It is 0 for `NIWLANG_VAL_TVHT_MODE_1` and `NIWLANG_VAL_TVHT_MODE_2N`, 0.5 * `TVHT_W` for `NIWLANG_VAL_TVHT_MODE_2C` and `NIWLANG_VAL_TVHT_MODE_4N` and 1.5 * `TVHT_W` for `NIWLANG_VAL_TVHT_MODE_4C`.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_ChannelNumberToCarrierFrequency80211ax

int32 __stdcall niWLANG_ChannelNumberToCarrierFrequency80211ax (float64 channelStartingFrequency, int32 channelNumber, float64 *carrierFrequency);

Purpose

Calculates the carrier frequency of the 802.11ax channels.

Parameters

Input

Name	Type	Description
channelStartingFrequency	float64	Specifies the start frequency of the frequency band. This value is expressed in Hz.
channelNumber	int32	Specifies the offset of the center frequency, in increments of 5 MHz, above the start frequency of the channel. The default value is 1.

Output

Name	Type	Description
carrierFrequency	float64*	Returns the carrier frequency. This value is expressed in Hz. The function calculates the carrier frequency using the following formula:

		carrier frequency (Hz) = channel starting frequency (Hz) + (channel number * 5 MHz)

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_CreateTriggerFrameMSDU

int32 __stdcall niWLANG_CreateTriggerFrameMSDU (niWLANGenerationSession session, char channelString[], int32 *generationDone, int32 triggerFrameMSDUBits[], int32 dataArraySize, int32 *actualDataArraySize);

Purpose

Creates 802.11ax trigger frame MSDU bits according to the configuration you specify. You must set the `NIWLANG_STANDARD` attribute to `NIWLANG_VAL_STANDARD_`

80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU to configure this function. For more information about Trigger frame generation, refer to the Generating the Trigger Frame topic.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
dataArraySize	int32	Specifies the number of elements in the **TriggerFrameMSDUBits** array.

Output

Name	Type	Description
generationDone	int32*	Indicates whether the function has generated all the data.
triggerFrameMSDUBits	int32[]	Returns an array of the encoded bits trace.
actualDataArraySize	int32*	Returns the actual number of elements populated in the **TriggerFrameMSDUBits** array parameter. If the array is not large enough to hold all the samples, the function returns an error and **actualDataArraySize** parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description

status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_CreateAndWriteWaveformsToFile

int32 __stdcall niWLANG_CreateAndWriteWaveformsToFile (niWLANGenerationSession session, char filePath[], int32 operation);

Purpose

Creates waveforms according to the properties configured in an niWLAN generation session and saves the waveforms to a file. In addition to creating the waveform, this function also saves the `NIWLANG_HEADROOM` and `NIWLANG_IQ_RATE` attributes for each waveform. The function reads the `NIWLANG_RF_BLANKING_MARKER_POSITIONS` attribute and saves the value to the file. The `NIWLANG_RF_BLANKING_MARKER_POSITIONS` attribute stored in the file is applicable to all waveforms stored in the file. Additionally, this function saves the NIWLANG_SIGNAL_BANDWIDTH attribute for each waveform, which is computed as:

Signal Bandwidth = 2* {(Channel Bandwidth/2) + |Max Carrier Frequency Offset|}

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
filePath	char[]	Specifies the absolute path to the TDMS file to which the toolkit writes the waveforms.
operation	int32	Specifies the operation to perform on the file. The default value is NIWLANG_VAL_FILE_OPERATION_MODE_CREATE_OR_REPLACE. `NIWLANG_VAL_FILE_OPERATION_MODE_OPEN(0)`: Opens an existing file to write the niWLANG settings. `NIWLANG_VAL_FILE_OPERATION_MODE_OPEN_OR_CREATE(1)`: Opens an existing file or creates a new file if the file does not exist. `NIWLANG_VAL_FILE_OPERATION_MODE_CREATE_OR_REPLACE(2)`: Creates a new file or replaces a file if it exists. `NIWLANG_VAL_FILE_OPERATION_MODE_CREATE(3)`: Creates a new file.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_ReadWaveformFromFile

int32 __stdcall niWLANG_ReadWaveformFromFile (char filePath[], char waveformName[], int64 offset, int64 count, float64 *t0, float64 *dt, NIComplexNumber waveform[], int32 waveformSize, int32 *actualNumWaveformSamples, float64 *IQRate, float64 *headroom, int32 *eof);

Purpose

Reads a waveform from a TDMS file. You can save this file using the NI WLAN Generation Soft Front Panel. This function returns headroom and I/Q rate waveform data that you can subsequently download to an NI RF vector signal generator.

Use the following active channel string formats to query this function.

NIWLANG_STANDARD attribute	Active Channel String Format	Comments
NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM	"" (empty string)	"" for all waveforms
NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM	"channelx"	channel1 for a waveform of channel with index 1
NIWLANA_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANA_VAL_STANDARD_80211AF_MIMO_OFDM, NIWLANA_VAL_STANDARD_80211AX_MIMO_OFDM	"[segmentz/channelx	"segment1/channel0" for a waveform of channel with index 0 of segment with index 1. "segment0/" is optional if the segment index is 1.

Parameters

Input

Name	Type	Description
filePath	char[]	Specifies the absolute path to the TDMS file from which the toolkit saves the waveform.
waveformName	char[]	Specifies the name of the waveform to read from the file. For example, use the channel0 string as a waveform name to read the waveform for channel 0.
offset	int64	Specifies the number of waveform samples at which the function begins reading the I/Q data. The default value is 0. If you set count to 1,000 and offset to 2, the function returns 1,000

		samples, starting from index 2 and ending at index 1,002.
count	int64	Specifies the maximum number of samples of the I/Q complex waveform to read from the file. The default value is –1, which returns all samples. If you set count to 1,000 and offset to 2, the function returns 1,000 samples, starting from index 2 and ending at index 1,002.
waveformSize	int32	Specifies the waveform size in samples.

Output

Name	Type	Description
t0	float64*	Returns the start time. This value is expressed in seconds.
dt	float64*	Returns the time interval between baseband I/Q samples. This value is expressed in seconds.
waveform	NIComplexNumber[]	Returns the baseband time-domain waveform. This parameter must be at least the size of the waveformSize parameter.
actualNumWaveformSamples	int32*	Returns the size of the data array. You can pass NULL to the waveform parameter to obtain the size of the waveform.
IQRate	float64*	Returns the I/Q rate of the waveform. This value is expressed in samples per seconds (S/s).
headroom	float64*	Returns the headroom of waveform. This value is expressed in dB.

eof	int32*	Indicates whether the end of file has been reached with this read.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_ReadBurstStartLocationsFromFile

int32 __stdcall niWLANG_ReadBurstStartLocationsFromFile (char filePath[], char channelString[], int32 burstStartLocations[], int32 dataArraySize, int32 *actualDataArraySize);

Purpose

Reads the burst start locations from a TDMS file. You can save this file using the NI WLAN Generation Soft Front Panel or the niWLANG_CreateandWriteWaveformstoFile function in a programming environment.

Parameters

Input

Name	Type	Description
filePath	char[]	Specifies the absolute path to the TDMS file from which the toolkit reads the waveform.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
dataArraySize	int32	Specifies the number of elements in the **burstStartLocations** array.

Output

Name	Type	Description
burstStartLocations	int32[]	Returns the burst start locations saved in the TDMS file. It is an array of sample positions of the start of the burst, within the waveform.
actualDataArraySize	int32*	Returns the actual number of elements populated in the **burstStartLocations** array parameter. If the array is not large enough to hold all the samples, the function returns an error and **actualDataArraySize** parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_ReadBurstStopLocationsFromFile

int32 __stdcall niWLANG_ReadBurstStopLocationsFromFile (char filePath[], char channelString[], int32 burstStopLocations[], int32 dataArraySize, int32 *actualDataArraySize);

Purpose

Reads the burst stop locations from a TDMS file. You can save this file using the NI WLAN Generation Soft Front Panel or the niWLANG_CreateandWriteWaveformstoFile function in a programming environment.

Parameters

Input

Name	Type	Description
filePath	char[]	Specifies the absolute path to the TDMS file from which the toolkit reads the waveform.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
dataArraySize	int32	Specifies the number of elements in the **burstStartLocations** array.

Output

Name	Type	Description
burstStopLocations	int32[]	Returns the burst stop locations saved in the TDMS file. It is an array of sample positions of the end of the burst, within the waveform.
actualDataArraySize	int32*	Returns the actual number of elements populated in the **burstStartLocations** array parameter. If the array is not large enough to hold all the samples, the function returns an error and **actualDataArraySize** parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status

code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGForceTClkSynchronization

int32 __stdcall niWLANG_RFSGForceTClkSynchronization (niWLANGenerationSession session, ViSession rfsgSessions[], int32 numberOfRFSGSessions, int32 forceSync);

Purpose

Specifies that the TClk synchronization has to be performed as the previous TClk synchronization is invalid due to niRFSG Reset function. This setting will be used by niWLANG RFSG Synchronize TClk function for TClk synchronization.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the

		niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
numberOfRFSGSessions	int32	Passes a reference of the WLAN generation session to the next function. Close the niWLAN generation session reference using the [niWLANG_CloseSession](#) function before the completion of execution to avoid possible memory leak issues.
forceSync	int32	Specifes that a fresh TClk synchronization has to be performed for TClk synchronized generation.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the [niWLANG_GetErrorString](#) function. The general meaning of the status code is as follows: Value / Meaning 0 — Success Positive Values — Warnings

Negative Values	Errors

RFSG Database

Contains functions to configure, store, or retrieve information from the RFSG database.

niWLANG_RFSGCreateAndDownloadWaveform

int32 __stdcall niWLANG_RFSGCreateAndDownloadWaveform (niWLANGenerationSession session, ViSession rfsgSession, char hwChannelString[], char waveformName[]);

Purpose

Creates a single channel waveform, writes it into the RFSG memory, and stores the I/Q rate, the burst start locations, the burst stop locations, the waveform size, and the actual headroom of the waveform in the RFSG database. Use this instance if the `NIWLANG_STANDARD` attribute is set to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, `NIWLANG_VAL_STANDARD_80211BG_DSSS`, or `NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM`.

Inside the `niWLANG_CreateAndDownloadWaveform` function, all the following operations that involve NI-RFSG are performed on the device. The `niWLANG_CreateAndDownloadWaveform` function completes the following operations:

1. Sets the `NIWLANG_MAXIMUM_HARDWARE_IQ_RATE` attribute according to the device model. You must set the value of the `NIWLANG_MAXIMUM_HARDWARE_IQ_RATE` attribute to 1250 MS/s if you are using the PXIe-5840/5841/5841 with PXIe-5655/5830/5831, 250 MS/s if you are using the PXIe-5646, 120 Ms/s if you are using the PXIe-5644/5645, and 200 MS/s if you are using the PXIe-5673/5673E.

2. Reads the NIWLANG_IQ_RATE attribute, and sets the NIRFSG_ATTR_IQ_RATE attribute to the value specified in the NIWLANG_IQ_RATE attribute. The value is stored in the RFSG database for the waveform and device.
3. If the NIWLANG_RF_BLANKING_ENABLED attribute is set to NIWLANG_VAL_TRUE and the device model PXIe-5644, PXIe-5645, PXIe-5646, PXIe-5840, PXIe-5841, or PXIe-5841 with PXIe-5655 is used, sets the NIRFSG_RF_BLANKING_SOURCE attribute to NIRFSG_VAL_MARKER0, if it is not set already.
4. Reads and stores the SIGNAL_BANDWIDTH attribute in the RFSG database for PXIe-5820/5830/5831/5841/5841 with 5655.
5. Reads and stores the NIWLANG_BURST_START_LOCATIONS attribute, NIWLANG_BURST_STOP_LOCATIONS attribute and the waveform size in the RFSG database.
6. Creates the WLAN waveform and downloads it to device.
7. Stores the NIWLANG_ACTUAL_HEADROOM attribute to the RFSG database.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSession	ViSession	Identifies the instrument session. The toolkit obtains this parameter from the niRFSG_init function or the niRFSG_InitWithOptions function.
hwChannelString	char[]	Specifies the RF vector signal generator channel. Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the name used to write the waveform to NI-RFSG device memory and store its attributes to RFSG database. This string is case-insensitive,

		alphanumeric, and does not use reserved words.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value — Meaning 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGCreateAndDownloadMIMOWaveforms

int32 __stdcall niWLANG_RFSGCreateAndDownloadMIMOWaveforms (niWLANGenerationSession session, ViSession rfsgSessions[], char *hwChannelStrings[], int32 numberOfTxChains, char waveformName[]);

Purpose

Creates multiple channel waveforms, writes each waveform into the respective RFSG memory, and stores the I/Q rate, burst start locations, burst stop locations, waveform size and the actual headroom for each channel of the waveform in the respective RFSG database, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMOOFDM.

The function creates a single channel waveform and writes the same waveform to RFSG memory of all devices, and stores the I/Q rate, burst start locations, burst stop locations, the waveform size and the actual headroom of the waveform in all RFSG databases, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM.

Inside the niWLANG_CreateAndDownloadMIMOWaveforms function, all the following operations that involve NI-RFSG are performed on the device. The niWLANG_CreateAndDownloadWaveform function completes the following operations:

1. Sets the NIWLANG_MAXIMUM_HARDWARE_IQ_RATE attribute according to the device model. You must set the value of the NIWLANG_MAXIMUM_HARDWARE_IQ_RATE attribute to 1250 MS/s if you are using the PXIe-5840/5841/5820/5830/5831, 250 MS/s if you are using the PXIe-5646, 120 Ms/s if you are using the PXIe-5644/5645, and 200 MS/s if you are using the PXIe-5673/5673E.
2. Reads the NIWLANG_IQ_RATE attribute, and sets the NIRFSG_ATTR_IQ_RATE attribute to the value specified in the NIWLANG_IQ_RATE attribute. The value is stored in the RFSG database for the waveform and device.
3. Sets the NIRFSG_ATTR_RF_BLANKING_SOURCE attribute to NIRFSG_VAL_MARKER0 (if it is not set already), if the NIWLANG_RF_BLANKING_ENABLED attribute is set to NIWLANG_VAL_TRUE and the device model is the PXIe-5644, PXIe-5645, PXIe-5646, PXIe-5840, or PXIe-5841.
4. Reads and stores the SIGNAL_BANDWIDTH attribute in the RFSG database for PXIe-5820/5830/5831/5841.

5. Reads and stores the NIWLANG_BURST_START_LOCATIONS attribute, NIWLANG_BURST_STOP_LOCATIONS attribute and the waveform size in the RFSG database.
6. Creates the WLAN waveform and downloads it to device.
7. Stores the NIWLANG_ACTUAL_HEADROOM attribute to the RFSG database.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.
rfsgSessions	ViSession[]	Identifies the instrument session. The toolkit obtains this parameter from the niRFSG_init function or the niRFSG_InitWithOptions function.
hwChannelStrings	char*[]	Specifies the RFSG device channel. Set this parameter to NULL.
numberOfTxChains	int32	Specifies the number of Transmit chains. The value of this parameter should be product of the values configured using the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS and NIWLANG_NUMBER_OF_SEGMENTS attributes.
waveformName	char[]	Specifies the name used to store the waveform. This string is case-insensitive, alphanumeric, and does not use reserved words.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code

either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred.
To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGReadAndDownloadWaveformsFromFile

int32 __stdcall int32 __stdcall niWLANG_RFSGReadAndDownloadWaveformsFromFile (ViSession rfsgSessions[], int32 numberOfChannels, char waveformName[], char filePath[]);

Purpose

Reads the waveforms stored in a TDMS file, writes them to the memory of the respective NI RF vector signal generator, and stores the I/Q rate, actual headroom, burst start locations, burst stop locations, waveform size and RF blanking marker positions of the waveforms in the RFSG database.

This function completes the following operations:

- Calls the niWLANG_ReadWaveformFromFile function to read the names of all waveforms present in the TDMS file.
- Calls the niWLANG_ReadRFBlankingMarkerPositionsfromFile function to read RF blanking marker positions.

- Completes the following operations for each waveform in the file and corresponding NI-RFSG device (instrument handle).
  - Calls the niWLANG Read Waveform from File function to read the waveform, along with its I/Q rate and headroom. The niRFSG I/Q Rate (S/s) property is set to the value of I/Q rate read from the file. The I/Q rate and headroom are stored in the RFSG database for the waveform specified in the waveform name parameter and NI-RFSG device.
  - For PXIe-5820/5830/5831/5841/5841 with 5655 devices, reads the Signal Bandwidth from the file and stores in the RFSG database for the waveform specified in the waveform name parameter and NI-RFSG device. If the value is not found in the file, then stores the value equal to 0.8*I/Q rate.
  - If the RF blanking marker positions, or burst start locations and burst stop locations read from the file are not empty arrays and are any of PXIe-5644, PXIe-5645, PXIe-5646, PXIe-5840, PXIe-5841, or PXIe-5841 with PXIe-5655 devices, the niRFSG RF Blanking Source property is set to "marker0", if it is not set. The RF blanking marker positions, burst start locations, burst stop locations, and the waveform size are stored in the RFSG database for the waveform specified in the waveform name parameter and the NI-RFSG device.
  - The function downloads the waveform read from the file to the NI-RFSG device.

Note: If there is a single waveform in the file, these operations are repeated for each NI-RFSG device with the same input waveform.

Parameters

Input

Name	Type	Description
waveformName	char[]	Specifies the name used to store the waveform. This string is case-insensitive, alphanumeric, and does not use reserved words.
rfsgSessions	ViSession[]	Identifies the instrument session. The toolkit obtains this

		parameter from the `niRFSG_init` function or the `niRFSG_InitWithOptions` function.
numberOfChannels	int32	Specifies the number of RFSG sessions.
filePath	char[]	Specifies the absolute path to the TDMS file from which the toolkit reads the waveforms.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGConfigureScript

int32 __stdcall niWLANG_RFSGConfigureScript (ViSession rfsgSession, char channelString[], char script[], float64 powerLevel);

Purpose

Configures the I/Q rate and power level of the waveforms that you specify in the script parameter. If the I/Q rates are the same for all the waveforms, this function sets the NIRFSG_ATTR_IQ_RATE attribute to the I/Q rate in the RFSG database. This function sets the NIRFSG_ATTR_POWER_LEVEL attribute to the sum of the power level that you specify in the powerLevel parameter and the minimum headroom value of all the waveforms.

	Note Call the niWLANG_RFSGCreateAndDownloadWaveform function before calling the niWLANG_RFSGConfigureScript function.

This function completes the following actions:

1. For devices other than PXIe 5820, calls the `niWLANG_RFSGRetrieveMinimumPAPRAllWaveforms` function to retrieve the minimum peak-to-average power ratio (PAPR) across all waveforms present in the script parameter. The `NIRFSG_ATTR_POWER_LEVEL` attribute is set to the sum of the power level parameter and the minimum PAPR.
2. For PXIe 5820/5830/5831/5841/5841 with 5655, calls the niWLANG_RFSGRetrieveSignalBandwidth(Script) function to retrieve the Signal Bandwidth and set niRFSG Signal Bandwidth (Hz) to the value.
3. Calls the `niWLANG_RFSGRetrieveIQRate` function to retrieve the I/Q rate and sets NIRFSG_ATTR_IQ_RATE attribute to the value.
4. Sets the `NIRFSG_ATTR_POWER_LEVEL_TYPE` attribute to `NIRFSG_VAL_PEAK_POWER`.
5. Calls the `niWLANG_RFSGInsertRFBlankingMarkerPositions` function to configure the script with RF blanking marker events.
6. Writes the resultant script to the NI-RFSG device using the `niRFSG_WriteScript` function.
7. Parses the script name from the script parameter and sets the `NIRFSG_ATTR_SELECTED_SCRIPT` attribute to the value.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
script	char[]	Specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name with the `NIRFSG_ATTR_SELECTED_SCRIPT` attribute.
powerLevel	float64	Specifies the average power level. This value is expressed in dBm. The default value is -10 dBm.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

Store

Contains the functions to store parameters for the NI RF vector signal generator.

niWLANG_RFSGStoreIQRate

int32 __stdcall niWLANG_RFSGStoreIQRate (ViSession RFSGSession, char channelString[], char waveformName[], float64 IQRate);

Purpose

Stores the I/Q rate that you specify in the IQRate parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the name of the waveform for which you want to store the I/Q rate.

IQRate	float64	Specifies the I/Q rate to store in the RFSG database. This value is expressed in samples per second (S/s).

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGStorePAPR

int32 __stdcall niWLANG_RFSGStorePAPR (ViSession RFSGSession, char channelString[], char waveformName[], float64 PAPR);

Purpose

Stores the headroom, or peak-to-average power ratio (PAPR), specified in the PAPR parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the waveform for which you want to store the maximum expected PAPR.
PAPR	float64	Specifies the headroom (or PAPR) to store in the RFSG database. This value is expressed in dB.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value | Meaning

0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGStoreRFBlankingMarkerPositions

int32 __stdcall niWLANG_RFSGStoreRFBlankingMarkerPositions (ViSession rfsgSession, char waveformName[], int32 rfBlankingMarkerPositions[], int32 rfBlankingMarkerPositionsArraySize);

Purpose

Stores the RF blanking marker positions that you specify in the rfBlankingMarkerPositions parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the waveform for which you want to store the RF blanking marker positions.
rfBlankingMarkerPositions	int32[]	Specifies the RF blanking marker positions to store. It is an array of sample positions, within the waveform, of marker events that can be used to toggle the state of RF blanking.
rfBlankingMarkerPositionsArraySize	int32	Specifies the number of elements in the

		rfBlankingMarkerPositions array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGStoreWaveformSize

int32 __stdcall niWLANG_RFSGStoreWaveformSize (ViSession rfsgSession, char waveformName[], int32* waveformSize);

Purpose

Stores the waveform size that you specify in the **waveformSize** parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to store the waveform size.

Output

Name	Type	Description
waveformSize	int32*	Specifies the waveform size to store. This value is expressed in samples. The default value is 1.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning

0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGStoreBurstStartLocations

int32 __stdcall niWLANG_RFSGStoreBurstStartLocations (ViSession rfsgSession, char waveformName[], int32 burstStartLocations[], int32 dataArraySize);

Purpose

Stores the burst start locations that you specify in the **burstStartLocations** parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the burst start locations. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.
dataArraySize	int32	Specifies the number of elements in the **burstStartLocations** array.

Output

Name	Type	Description

burstStartLocations	int32[]	Returns the burst start locations stored, for the waveform you specified in the **waveformName** parameter. It is an array of sample positions of the start of the burst, within the waveform. The default value is "" (empty array).

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGStoreBurstStopLocations

int32 __stdcall niWLANG_RFSGStoreBurstStopLocations (ViSession rfsgSession, char waveformName[], int32 burstStopLocations[], int32 dataArraySize);

Purpose

Stores the burst stop locations that you specify in the burst stop locations parameter in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to store the burst stop locations.
burstStopLocations	int32[]	specifies the burst stop locations to store. It is an array of sample positions of the end of the burst, within the waveform. The default value is "" (empty array).
dataArraySize	int32	Specifies the number of elements in the **burstStopLocations** array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the

		function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Success</td></tr><tr><td>Positive Values</td><td>Warnings</td></tr><tr><td>Negative Values</td><td>Errors</td></tr></table>

Retrieve

Contains functions to retrieve parameters for the NI RF vector signal generator.

niWLANG_RFSGRetrieveIQRate

int32 __stdcall niWLANG_RFSGRetrieveIQRate (ViSession rfsgSession, char channelString[], char waveformName[], float64 *IQRate);

Purpose

Returns the I/Q rate stored in the RFSG database. The function uses the waveform name as the key to retrieve the waveform attributes.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions

		functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the I/Q rate. The toolkit uses the waveformName parameter as the key to retrieve the waveform attributes in the RFSG database.

Output

Name	Type	Description
IQRate	float64*	Returns the I/Q rate stored in the RFSG database for the waveform you specified in the waveformName parameter. This value is expressed in samples per second (S/s).

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success

Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGRetrieveMinimumPAPRAllWaveforms

int32 __stdcall niWLANG_RFSGRetrieveMinimumPAPRAllWaveforms (ViSession rfsgSession, char channelString[], char script[], float64 *PAPR);

Purpose

Returns the minimum value of the headroom, or peak-to-average power ratio (PAPR), of all the waveforms in the script specified in the script parameter.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
script	char[]	Specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name with the NIRFSG_ATTR_SELECTED_SCRIPT attribute.

Output

Name	Type	Description
PAPR	float64*	Returns the minimum of all the maximum expected PAPR

		values stored in the RFSG database. This value is expressed in dB.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGRetrieveIQRateAllWaveforms

int32 __stdcall niWLANG_RFSGRetrieveIQRateAllWaveforms (ViSession rfsgSession, char channelString[], char script[], float64 *IQRate);

Purpose

Checks the I/Q rate of all the waveforms in the script that you specify in the script parameter. This function returns the I/Q rate if the I/Q rates are the same for all the waveforms. If the I/Q rates are different, the function returns an error.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
script	char[]	Specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name with the `NIRFSG_ATTR_SELECTED_SCRIPT` attribute.

Output

Name	Type	Description
IQRate	float64*	Returns the I/Q rate, if the I/Q rates are the same for all waveforms specified in the script parameter. If the I/Q rates are different, the function returns an error. This value is expressed in samples per second (S/s).

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the

	function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGRetrievePAPR

int32 __stdcall niWLANG_RFSGRetrievePAPR (ViSession rfsgSession, char channelString[], char waveformName[], float64 *PAPR);

Purpose

Returns the headroom, or peak-to-average power ratio (PAPR), stored in the RFSG database. The function uses the waveform name as the key to retrieve the waveform PAPR.

	Note Use the `niWLANG_RFSGStorePAPR` function to store the PAPR in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions

		functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the PAPR. The toolkit uses the waveformName parameter as the key to retrieve the waveform properties in the RFSG database.

Output

Name	Type	Description
PAPR	float64*	Returns the PAPR stored in the RFSG database, for the waveform you specified in the waveformName parameter. This value is expressed in dB.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success

Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGRetrieveRFBlankingMarkerPositions

int32 __stdcall niWLANG_RFSGRetrieveRFBlankingMarkerPositions (ViSession rfsgSession, char waveformName[], int32 rfBlankingMarkerPositionsArraySize, int32 rfBlankingMarkerPositions[], int32 *actualRFBlankingMarkerPositionsArraySize);

Purpose

Returns the RF blanking marker positions stored in the RFSG database. The function uses the waveform name as the key to retrieve the waveform properties.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the names of the waveforms to clear. If you set this parameter as empty, the function clears all the waveforms and their properties.
rfBlankingMarkerPositionsArraySize	int32	Specifies the number of elements in the **rfBlankingMarkerPositions** array.

Output

Name	Type	Description

rfBlankingMarkerPositions	int32[]	Returns the RF blanking marker positions saved in the TDMS file. It is an array of sample positions, within waveform, of marker events that can be used to toggle the state of RF blanking.
actualRFBlankingMarkerPositionsArraySize	int32*	Specifies the actual number of elements populated in the **rfBlankingMarkerPositions** array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGRetrieveBurstStartLocations

int32 __stdcall niWLANG_RFSGRetrieveBurstStartLocations (ViSession rfsgSession, char waveformName[], int32 burstStartLocations[], int32 dataArraySize, int32* actualDataArraySize);

Purpose

Returns the burst start locations stored in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the burst start locations. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.
dataArraySize	int32	Specifies the number of elements in the **burstStartLocations** array.

Output

Name	Type	Description
burstStartLocations	int32[]	Returns the burst start locations stored, for the waveform you specified in the **waveformName** parameter. It

		is an array of sample positions at the start of the burst, within the waveform.
actualDataArraySize	int32*	Returns the actual number of elements populated in the **burstStartLocations** array parameter. If the array is not large enough to hold all the samples, the function returns an error and **actualDataArraySize** parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niWLANG_RFSGRetrieveBurstStopLocations

int32 __stdcall niWLANG_RFSGRetrieveBurstStopLocations (ViSession rfsgSession, char waveformName[], int32 burstStopLocations[], int32 dataArraySize, int32* actualDataArraySize);

Purpose

Returns the burst stop locations stored in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the burst stop locations. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.
dataArraySize	int32	Specifies the number of elements in the **burstStopLocations** array.

Output

Name	Type	Description
burstStopLocations	int32[]	Returns the burst stop locations stored, for the waveform you specified in the **waveformName** parameter. It

		is an array of sample positions of the end of the burst, within the waveform.
actualDataArraySize	int32*	Returns the actual number of elements populated in the **burstStopLocations** array parameter. If the array is not large enough to hold all the samples, the function returns an error and **actualDataArraySize** parameter returns the minimum expected size of the output array.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value: 0 — Meaning: Success Value: Positive Values — Meaning: Warnings Value: Negative Values — Meaning: Errors

niWLANG_RFSGRetrieveWaveformSize

int32 __stdcall niWLANG_RFSGRetrieveWaveformSize (ViSession rfsgSession, char waveformName[], int32* waveformSize);

Purpose

Returns the waveform size stored in the RFSG database.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
waveformName	char[]	Specifies the name of the waveform for which you want to retrieve the waveform size. The toolkit uses this parameter as the key to retrieve the waveform properties in the RFSG database.

Output

Name	Type	Description
waveformSize	int32*	Returns the waveform size stored, for the waveform you specified in the **waveformName** parameter.

Return Value

Name	Type	Description

status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_ClearDatabase

int32 __stdcall niWLANG_RFSGClearDatabase (ViSession rfsgSession, char channelString[], char waveformName[]);

Purpose

Clears the attributes stored in the RFSG database and clears the waveforms from the NI RF vector signal generator memory.

This function clears the waveforms and the attributes of the waveforms that you specified in the waveformName parameter. If you do not set the waveformName parameter or set it to "" (empty string), this function clears all the waveforms and their attributes.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
waveformName	char[]	Specifies the names of the waveforms to clear. If you set this parameter as empty, the function clears all the waveforms and their attributes.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values	Errors

niWLANG_RFSGConfigurePowerLevel

int32 __stdcall niWLANG_RFSGConfigurePowerLevel (ViSession rfsgSession, char channelString[], char script[], float64 powerLevel);

Purpose

Looks up the waveforms in the script, retrieves the minimum actual headrooms of the waveforms in the script, adds this value to the powerLevel parameter, and sets the result to the `NIRFSG_ATTR_POWER_LEVEL` attribute. Set the `NIRFSG_ATTR_POWER_LEVEL_TYPE` attribute to `NIRFSG_VAL_PEAK_POWER` before calling this function.

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
channelString	char[]	Set this parameter to "" (empty string) or NULL.
script	char[]	Specifies the script that controls waveform generation. NI-RFSG supports multiple scripts that may be selected by name with the NIRFSG_ATTR_SELECTED_SCRIPT attribute.

powerLevel	float64	Specifies the average power level. This value is expressed in dBm.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_RFSGInsertRFBlankingMarkerPositions

int32 __stdcall niWLANG_RFSGInsertRFBlankingMarkerPositions (ViSession rfsgSession, char script[], char scriptOut[], int32 lenOfScriptOut, int32 *actualLenOfScriptOut);

Purpose

Configures the script you specify in the **script** parameter for RF blanking marker events.

For example, assume that the NIRFSG_ATTR_RF_BLANKING_SOURCE attribute is set to "marker0", the value of the NIWLANG_RF_BLANKING_MARKER_POSITIONS attribute is {10000, 12000}, and you configure the **script** parameter as shown in the following example:

```
script scriptName

generate waveformName

end script
```

The **scriptOut** parameter value is as shown in the following example:

```
script scriptName

generate waveformName marker0 (10000, 12000)

end script
```

Parameters

Input

Name	Type	Description
rfsgSessions	ViSession[]	Specifies a reference to an NI-RFSG instrument session. This parameter is obtained from the niRFSG_init or niRFSG_InitWithOptions functions and identifies a particular instrument session.
script	char[]	Specifies the script that controls waveform generation.
lenOfScriptOut	int32	Specifies the size of the **scriptOut** array.

Output

Name	Type	Description
actualLenOfScriptOut	int32*	Returns the number of elements in the **scriptOut** array. If the array is not large enough to hold the output script, the function returns an

		error and this parameter returns the minimum expected size of the output array.
scriptOut	char[]	Returns the modified script which has RF blanking marker events configured.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_ResetSession

int32 __stdcall niWLANG_ResetSession (niWLANGenerationSession session);

Purpose

Resets all the attributes of the session to their default values.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

niWLANG_CloseSession

int32 __stdcall niWLANG_CloseSession (niWLANGenerationSession session);

Purpose

Closes the niWLAN generation session and releases resources associated with that session. Call this function once for each unique named session that you have created.

Parameters

Input

Name	Type	Description
session	niWLANGenerationSession	Specifies the niWLAN generation session.

Return Value

Name	Type	Description
status	int32	Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. Examine the status code from each call to the function to determine if an error occurred. To obtain a text description of the status code, call the niWLANG_GetErrorString function. The general meaning of the status code is as follows: Value / Meaning: 0 — Success Positive Values — Warnings Negative Values — Errors

WLAN Generation Attributes

Contains the niWLAN Generation attributes to access configuration options for WLAN applications using an NI RF vector signal generator.

Note When you query an attribute, the toolkit verifies the WLAN session to ensure that all relevant attributes are set and are valid. If the verification fails, the toolkit returns an error.

NIWLANG_STANDARD

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the IEEE 802.11 standard, which indicates the type of physical layer, for signal generation.

Note: If you do not select a standard, the toolkit returns an error.

Set Function: niWLANG_SetStandard
Get Function: niWLANG_GetStandard

Values:

`NIWLANG_VAL_STANDARD_80211AG_OFDM (0)`	Corresponds to the OFDM mode defined in IEEE Standard 802.11a-1999 and the extended rate physical layer-OFDM (ERP-OFDM) mode defined in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211BG_DSSS (1)`	Corresponds to all the compulsory and optional modes defined in IEEE

	Standard 802.11b-1999 and the ERP-packet binary convolutional coding (ERP-PBCC) mode in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM (2)`	Corresponds to the optional direct sequence spread spectrum-OFDM (DSSS-OFDM) mode defined in IEEE Standard 802.11g-2003.
`NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM (3)`	Corresponds to IEEE Standard 802.11n-2009. To use this option, you must set the compatibilityVersion parameter on the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_020000, NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
`NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM (4)`	Corresponds to IEEE Standard 802.11ac-2013. To use this option, you must set the compatibilityVersion

	parameter on the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
`NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM (5)`	Corresponds to IEEE P802.11ah/D1.3. To use this option, you must set the compatibilityVersion parameter on the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
`NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM (6)`	Corresponds to IEEE Standard 802.11af-2013. To use this option, you must set the compatibilityVersion parameter on the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPA

	TIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.
`NIWLANG_VAL_STANDARD_80211J_OFDM (7)`	Corresponds to the OFDM mode defined in IEEE Standard 802.11j 2004.
`NIWLANG_VAL_STANDARD_80211P_OFDM (8)`	Corresponds to the OFDM mode defined in IEEE Standard 802.11p 2010.
`NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM (9)`	Corresponds to IEEE P802.11ax/D6.0. To use this option, you must set the compatibilityVersion parameter on the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

NIWLANG_CHANNEL_BANDWIDTH

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`

Description:	Specifies the channel width used for transmitting the signal. This value is expressed in Hz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, the channel bandwidth must be equal to 20 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211J_OFDM, the channel bandwidth must be equal to 10 MHz or 20 MHz. NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, the channel bandwidth must be equal to 5 MHz, 10 MHz, or 20 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, the channel bandwidth must be either 20 MHz or 40 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, the channel bandwidth must be either 20 MHz, or 40 MHz, or 80 MHz, or 160 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the channel bandwidth must be either 20 MHz, or 40 MHz, or 80 MHz, or 160 MHz or 320 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the channel bandwidth must be equal to 1MHz, 2 MHz, 4 MHz, 8 MHz, or 16 MHz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the channel bandwidth must be equal to 6

MHz, 7 MHz or 8 MHz.

For OFDM signals, channel bandwidth determines the number of pilot and data subcarriers used. The toolkit ignores the NIWLANG_CHANNEL_BANDWIDTH attribute for other values of the NIWLANG_STANDARD attribute.

Set Function: niWLANG_SetChannelBandwidth
Get Function: niWLANG_GetChannelBandwidth

NIWLANG_CARRIER_FREQUENCY

Data Type: float64

Access: read/write

Functions: `niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`

Description: Specifies the carrier frequency for signal generation. This value is expressed in Hz. If you set the NIWLANG_MULTI_SEGMENT_GENERATION_MODE attribute to NIWLANG_VAL_SINGLE_GENERATOR, you must set this attribute for each segment using the active channel string 'segmenty'.

The default value is 2.412 GHz.

Set Function: niWLANG_SetCarrierFrequency

NIWLANG_OVERSAMPLING_FACTOR

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description:

Specifies the number of times the toolkit increases the Nyquist sample rate to obtain the final sample rate of the signal.

The default value is 4, and the minimum value is 2.

Set Function: niWLANG_SetOverSamplingFactor
Get Function: niWLANG_GetOverSamplingFactor

NIWLANG_IDLE_INTERVAL

Data Type: float64

Access: read/write

Functions:

`niWLANG_SetScalarAttributeF64`
`niWLANG_GetScalarAttributeF64`

Description:

Specifies the interframe spacing for signal generation. This value is expressed in seconds. If you set the NIWLANG_IDLE_INTERVAL_MODE attribute to NIWLANG_VAL_IDLE_INTERVAL_MODE_SPLIT, the toolkit places half of the interframe spacing on either side of the burst. If you set the NIWLANG_IDLE_INTERVAL_MODE attribute to NIWLANG_VAL_IDLE_INTERVAL_MODE_POST_BURST, the toolkit places the interframe spacing at the end of the waveform. The waveform contains zeros for the duration of the interframe spacing.

Tip: For higher values of idle interval, LabVIEW may run out of memory. For a large idle interval, you can create a small idle waveform and generate the waveform multiple times using scripting.

The default value is 100 microseconds. Valid values are 0 to 1, inclusive.

Set Function: niWLANG_SetIdleInterval
Get Function: niWLANG_GetIdleInterval

NIWLANG_IDLE_INTERVAL_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies how the idle interval is placed in the generated waveform. You cannot set this attribute to NIWLANG_VAL_IDLE_INTERVAL_MODE_SPLIT if you set the NIWLANG_RF_BLANKING_ENABLED attribute to NIWLANG_VAL_TRUE. The default value is NIWLANG_VAL_IDLE_INTERVAL_MODE_SPLIT. Set Function: niWLANG_SetIdleIntervalMode Get Function: niWLANG_GetIdleIntervalMode
Values:	

`NIWLANG_VAL_IDLE_INTERVAL_MODE_SPLIT (0)`	Half the idle interval is placed on either side of the burst in the generated waveform.
`NIWLANG_VAL_IDLE_INTERVAL_MODE_POST_BURST (1)`	Entire idle interval is placed at the end of the burst in the generated waveform.

NIWLANG_NUMBER_OF_FRAMES

Data Type:	int32
Access:	read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the number of frames to generate. Each iteration of the niWLANG_CreateWaveformComplexF64 or niWLANG_CreateMIMOWaveformsComplexF64 function generates only one frame along with the idle interval that you specify using the NIWLANG_IDLE_INTERVAL attribute.

If you set the NIWLANG_NUMBER_OF_FRAMES attribute to a value greater than 1, call the niWLANG_CreateWaveformComplexF64 function or the NIWLANG_CreateMIMOWaveformsComplexF64 function in a loop and concatenate the output values from all iterations. If you encounter memory usage issues, download a single frame to the NI RF vector signal generator memory on each iteration.

To generate the required number of frames, call the niWLANG_CreateWaveformComplexF64 function or the NIWLANG_CreateMIMOWaveformsComplexF64 function for number of times equal to NIWLANG_NUMBER_OF_FRAMES attribute. To stop the loop, use the done parameter of the niWLANG_CreateWaveformComplexF64 function or the NIWLANG_CreateMIMOWaveformsComplexF64 function. The toolkit sets the value of the done parameter to 1 for the last frame.

You can use the niRFSG_AllocateArbWaveform function to preallocate arb memory and then download the waveform frame-by-frame.

The default value is 1. Valid values are 1 to 1,000 (inclusive).

Set Function: niWLANG_SetNumberOfFrames
Get Function: niWLANG_GetNumberOfFrames

NIWLANG_FRAME_DURATION

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the duration of a generated frame excluding idle interval. This property is applicable only if you set NIWLANG_PAYLOAD_AUTO_DATA_LENGTH attribute to NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_FRAME_DURATION. When the value of NIWLANG_FRAME_DURATION results in fractional NIWLANG_NUMBER_OF_DATA_SYMBOLS, and if it results in generated frame duration less than 5.484 milliseconds, the number of data symbols is rounded to the next highest integer, otherwise, the number of symbols is rounded to the next lowest integer. The default value is 1 millisecond. Valid values are 100 microseconds to 5.484 milliseconds. Set Function: niWLANG_SetFrameDuration Get Function: niWLANG_GetFrameDuration

NIWLANG_AUTO_HEADROOM_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the toolkit calculates the headroom or uses a user-defined value. For

multiframe generation, the toolkit uses the headroom calculated on the first frame to scale the waveform. NI recommends that you do not set this attribute to NIWLANG_VAL_TRUE for multiframe generation because variation of the peak-to-average power ratio (PAPR) across frames may lead to excessive clipping. To avoid excessive clipping, set this attribute to NIWLANG_VAL_FALSE and use the default values for the NIWLANG_HEADROOM attribute.

The default value is NIWLANG_VAL_TRUE.

Note: Use this attribute only if you set the NIWLANG_COMPATIBILITY_VERSION attribute to NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

Set Function:
niWLANG_SetAutoHeadroomEnabled
Get Function:
niWLANG_GetAutoHeadroomEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Specifies that the toolkit uses the headroom that you specify in the NIWLANG_HEADROOM attribute.
`NIWLANG_VAL_TRUE (1)`	Specifies that the toolkit calculates the headroom automatically.

NIWLANG_HEADROOM

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the headroom per transmit channel. This value is expressed in dB. This value represents the maximum peak-to-average power ratio (PAPR) allowed in the generated signal. The toolkit clips any portion of the signal that exceeds the peak power corresponding to this value. The toolkit ignores this attribute if you set the NIWLANG_AUTO_HEADROOM_ENABLED attribute to NIWLANG_VAL_TRUE. If you set the NIWLANG_AUTO_HEADROOM_ENABLED attribute to NIWLANG_VAL_FALSE and you do not specify the NIWLANG_HEADROOM attribute, the toolkit uses default values based on the NIWLANG_STANDARD attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the headroom applied on the signal with respect to the average power computed is based on the NIWLANG_AVERAGE_POWER_REFERENCE attribute. Note: If you specify a value that is more than the actual PAPR of the signal, there is loss of dynamic range of the digital-to-analog converter (DAC). If you specify a value that is less than the actual PAPR of the signal, the toolkit clips the generated signal. Note: In toolkit version NIWLANG_VAL_COMPATIBILITY_VERSION_020000, the NIWLANG_HEADROOM attribute was

called NIWLANG_MAX_EXPECTED_PAPR. To get the same behavior of this attribute as in NIWLANG_VAL_COMPATIBILITY_VERSION_020000, you must set the NIWLANG_AUTO_HEADROOM_ENABLED attribute to NIWLANG_VAL_FALSE in NIWLANG_VAL_COMPATIBILITY_VERSION_030000, NIWLANG_VAL_COMPATIBILITY_VERSION_040000, or NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string as the active channel string to configure this attribute.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use 'channelx' as the active channel string to configure this attribute.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDMor NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use '[segmenty/]channelx' as the active channel string to configure this attribute. 'segment0/' is optional if the NIWLANG_NUMBER_OF_SEGMENTS attribute is set to 1.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use '[segmenty/]channelx' as the active channel string to configure this attribute. '[segment0/]' is optional if the NIWLANG_NUMBER_OF_SEGMENTS attribute is set to 1.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use 'channelx' as the active channel string when the NIWLANG_MULTI_SEGMENT_GENERATION_MODE attribute is set to NIWLANG_VAL_SINGLE_GENERATOR.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS, the default value is 5. Otherwise, the default value is 12.

Refer to the Configuring Active Channels (LabWindows/CVI) topic for more information about configuring active channels.

Set Function: niWLANG_SetHeadroom
Get Function: niWLANG_GetHeadroom

NIWLANG_FULLSCALE_BACKOFF

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the additional scaling factor applied to the waveform when you set the NIWLANG_AUTO_HEADROOM_ENABLED attribute to NIWLANG_VAL_TRUE. This value is expressed in dB. The toolkit ignores this attribute, if you set NIWLANG_AUTO_HEADROOM_ENABLED attribute to NIWLANG_VAL_FALSE. The default value is 2.

Set Function: niWLANG_SetFullscaleBackoff
Get Function: niWLANG_GetFullscaleBackoff

NIWLANG_AVERAGE_POWER_REFERENCE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the portions of the packet used to compute the average power value to apply waveform power scaling and additive white Gaussian noise (AWGN). This attribute is valid when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU. The default value is NIWLANG_VAL_AVERAGE_POWER_REFERENCE_NON_BOOSTED_FIELDS. Set Function: niWLANG_SetAveragePowerReference Get Function: niWLANG_GetAveragePowerReference
Values:	

`NIWLANG_VAL_AVERAGE_POWER_REFERENCE_NON_BOOSTED_FIELDS (0)`	Specifies that the toolkit uses the packet fields where no power boosting is applied.
`NIWLANG_VAL_AVERAGE_POWER_REFERENCE_POWER_BOOSTED_FIELDS (1)`	Specifies that the toolkit uses the packet fields where power boosting is applied.

NIWLANG_VAL_AVERAGE_POWER_REFERENCE_ENTIRE_PACKET (2)	Specifies that the toolkit uses the complete packet for computing the average power value.

NIWLANG_MAXIMUM_HARDWARE_IQ_RATE

Data Type: float64

Access: read/write

Functions: niWLANG_SetScalarAttributeF64
niWLANG_GetScalarAttributeF64

Description: Specifies the maximum I/Q rate that the NI vector signal generator supports.

This attribute is set according to the device model in the niWLANG_RFSGCreateandDownloadWaveform function.

If you are using PXIe-5840/PXIe-5841/PXIe-5841 with PXIe-5655/PXIe-5820/PXIe-5830/PXIe-5831, the valid value is 1250 MS/s.

If you are using the PXIe-5646, the valid value is 250 MS/s.

If you are using the PXIe-5644/PXIe-5645, the valid value is 120 MS/s.

If you are using the PXIe-5673/5673E, the valid value is 200 MS/s.

Set Function:
niWLANG_SetMaximumHardwareIQRate
Get Function:
niWLANG_GetMaximumHardwareIQRate

NIWLANG_RF_BLANKING_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable RF blanking. Set this attribute to NIWLANG_VAL_TRUE, if you want to attenuate the RF OUT signal during the idle interval. This behavior prevents any DC leakage from the local oscillator of the signal generator from appearing at the RF OUT signal. RF blanking attenuates the RF OUT signal of signal generators quickly. For more details about RF blanking, refer to the NIWLANG_BURST_START_LOCATIONS attribute, the NIWLANG_BURST_STOP_LOCATIONS attribute, the niWLANG_RFSGCreateAndDownloadWaveform function, and the niWLANG_RFSGConfigureScript function. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetRFBlankingEnabled Get Function: niWLANG_GetRFBlankingEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Specifies that the toolkit does not enable RF blanking.
`NIWLANG_VAL_TRUE (1)`	Specifies that the toolkit enables RF blanking. If you select this option, the toolkit completes the following actions: The

	toolkit returns the NIWLANG_RF_BLANKING_MARKER_POSITIONS attribute. The toolkit queries the burst start locations and the burst stop locations to get marker positions that can be used to toggle the state of RF blanking. These marker positions are generated such that RF blanking is ON during the idle interval. The niWLANG_RFSGCreateAndDownloadWaveform function sets the niRFSG RF Blanking Source attribute to 'marker0' , if it is not set already. The niWLANG_RFSGConfigureScript function modifies the input from the script parameter to specify marker events.

NIWLANG_LO_SHARING_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable configuration for sharing of local oscillator (LO) signal for multiple NI RF vector signal generators and NI

	RF vector signal transceivers, or both. This attribute is queried as part of the niWLANG_RFSGConfigureMultipleDeviceSynchronization function. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetLOSharingEnabled Get Function: niWLANG_GetLOSharingEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables LO sharing. `NIWLANG_VAL_TRUE (1)` Enables LO sharing.

NIWLANG_LO_FREQUENCY_OFFSET_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies how the LO frequency offset is derived to configure frequency on the NI RF vector signal generators and the NI synthesizers in the niWLANG_RFSGConfigureFrequecySingleLO and niWLANG_RFSGConfigureFrequecyMultipleLO functions. The toolkit ignores this attribute if you do not use PXIe-5840, PXIe-5841, PXIe-5841 with PXIe-5655, PXIe-5646, PXIe-5830, or PXIe-5831 devices. The default value is NIWLANG_VAL_LO_FREQUENCY_OFFSET_MODE_AUTO. Set Function: niWLANG_SetLOFrequencyOffsetMode Get Function: niWLANG_GetLOFrequencyOffsetMode

Values:

NIWLANG_VAL_LO_FREQUENCY_OFFSET_MODE_AUTO (0)	Specifies that the LO frequency offset value is computed for optimal EVM performance.
NIWLANG_VAL_LO_FREQUENCY_OFFSET_MODE_USER_DEFINED (1)	Sets the LO frequency offset value to the value you specified in the NIWLANG_LO_FREQUENCY_OFFSET attribute.
NIWLANG_VAL_LO_FREQUENCY_OFFSET_MODE_DISABLED (2)	Specifies that the LO frequency offset is not set by the toolkit.

NIWLANG_LO_FREQUENCY_OFFSET

Data Type: float64

Access: read/write

Functions: `niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`

Description: Specifies the LO frequency offset to be used when you set the NIWLANG_LO_FREQUENCY_OFFSET_MODE attribute to NIWLANG_VAL_LO_FREQUENCY_OFFSET_MODE_USER_DEFINED.

The default value is 0 Hz.

Set Function: niWLANG_SetLOFrequencyOffset
Get Function: niWLANG_GetLOFrequencyOffset

NIWLANG_IQ_RATE

Data Type: float64

Access: read only

Functions: `niWLANG_GetScalarAttributeF64`

Description: Returns the recommended sample rate for the current signal configuration. This value is expressed in samples per second. The dt parameter of the created waveform is the inverse of the recommended sample rate.

If you set the usewaveformdtforIQrate parameter of the niRFSG_Write_Arb_Waveform function to False, wire the NIWLANG_IQ_RATE attribute to the NIRFSG_IQ_RATE attribute. This value is expressed in samples per second. Get Function: niWLANG_GetIQRate

NIWLANG_ACTUAL_HEADROOM

Data Type: float64

Access: read only

Functions: `niWLANG_GetScalarAttributeF64`

Description: Returns the actual headroom that the toolkit applies to the waveform. This value is expressed in dB. Use an active channel string to query this attribute for a transmit channel.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM you must use an empty active channel string to query this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OF

DM, you must use an 'channelx' active channel string to query this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDMor NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use '[segmenty/]channelx' as the active channel string to query this attribute. '[segment0/]' is optional if the NIWLANG_NUMBER_OF_SEGMENTS attribute is set to 1.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use '[segmenty/]channelx' as the active channel string to query this attribute. '[segment0/]' is optional if the NIWLANG_NUMBER_OF_SEGMENTS attribute is set to 1.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use 'channelx' as the active channel string when the NIWLANG_MULTI_SEGMENT_GENERATION_MODE attribute is set to NIWLANG_VAL_SINGLE_GENERATOR.

Refer to the Configuring Active Channels (LabWindows/CVI) help topic for more information about configuring active channels.

Get Function: niWLANG_GetActualHeadroom

NIWLANG_RF_BLANKING_MARKER_POSITIONS

Data Type:	int32 []
Access:	read only
Functions:	`niWLANG_GetVectorAttributeI32`
Description:	Returns the array of sample positions of marker events, which are used to toggle the state of RF

blanking, within the waveform. The marker positions are such that RF blanking is enabled during the idle interval. This attribute is applicable only if the NIWLANG_RF_BLANKING_ENABLED attribute is set to NIWLANG_VAL_TRUE. This attribute is read by the niWLANG_RFSGCreateandDownloadWaveform function to store RF blanking marker positions in the RFSG database.

Get Function: niWLANG_GetRFBlankingMarkerPositions

NIWLANG_BURST_START_LOCATIONS

Data Type:	int32 []
Access:	read/write
Functions:	`niWLANG_SetVectorAttributeI32` `niWLANG_GetVectorAttributeI32`
Description:	Returns the array of sample positions of start of the burst, within the waveform. Set Function: niWLANG_SetBurstStartLocations Get Function: niWLANG_GetBurstStartLocations

NIWLANG_BURST_STOP_LOCATIONS

Data Type:	int32 []
Access:	read/write
Functions:	`niWLANG_SetVectorAttributeI32` `niWLANG_GetVectorAttributeI32`
Description:	Returns the array of sample positions of end of the burst, within the waveform. Set Function: niWLANG_SetBurstStopLocations Get Function: niWLANG_GetBurstStopLocations

NIWLANG_SIGNAL_BANDWIDTH

Data Type:	float64
Access:	read only
Functions:	`niWLANG_GetScalarAttributeF64`
Description:	Returns the Signal Bandwidth value that needs to be configured on the NI RFSG session for the PXIe-5820, PXIe-5830, PXIe-5831, PXIe-5841, or PXIe-5841 with PXIe-5655 devices. The toolkit computes this value using the following equation. Signal Bandwidth (Hz) = 2 * {(Channel Bandwidth/2) + |Maximum Carrier Frequency Offset|}. Get Function: niWLANG_GetSignalBandwidth

NIWLANG_AMPDU_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether all medium access control (MAC) protocol data units (MPDUs) are transmitted as aggregate-MPDU (A-MPDU). Note: This attribute is applicable only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OF

DM,
NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, or
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_NON_HT_MODULATION_MODE attribute to NIWLANG_VAL_NON_HT_MODULATION_MODE_OFF.

The default value is NIWLANG_VAL_FALSE, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM.

The default value is NIWLANG_VAL_TRUE, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM.

Set Function: niWLANG_SetAMPDUEnabled
Get Function: niWLANG_GetAMPDUEnabled

Values:

NIWLANG_VAL_FALSE (0)	Disables the attribute.
NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_PAYLOAD_AUTO_NUMBER_OF_MPDUS

Data Type: int32

Access: read/write

Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to compute the number of MPDUs in an AMPDU of 802.11ax Trigger-Based PPDU using the following frame parameters: NIWLANG_L_SIG_LENGTH Length attribute, NIWLANG_PRE_FEC_PADDING_FACTOR attribute, and NIWLANG_LDPC_EXTRA_SYMBOL_SEGMENT attribute. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetAutoNumberOfMPDUs Get Function: niWLANG_GetAutoNumberOfMPDUs
Values:	

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_PAYLOAD_MAC_FRAME_TYPE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the type of frame for the MPDU. You can set this attribute to NIWLANG_TRIGGER_FRAME, only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OF

DM, or
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211B/G_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, use an empty string active channel string format to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_FDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use 'mpdux' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, use 'mpdux' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, or use 'userx/mpduy' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE and the NIWLANG_PPDU_TYPE attribute is set to

NIWLANG_VAL_PPDU_TYPE_MU_PPDU to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, or use 'mpdux' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_EXTENDED_RANGE_SU_PPDU to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, 'mpdux' as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, or use 'userx/mpduy' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_TRIGGER_BASED_PPDU to configure this attribute.

The default value is NIWLANG_VAL_PAYLOAD_FRAME_TYPE_GENER

	AL_FRAME. Set Function: niWLANG_SetMACFrameType Get Function: niWLANG_GetMACFrameType
Values:	`NIWLANG_VAL_PAYLOAD_MAC_FRAME_TYPE_GENERAL_FRAME (0)` Specifies that the MAC frame type is General. `NIWLANG_VAL_PAYLOAD_MAC_FRAME_TYPE_DATA_FRAME (1)` Specifies that the MAC frame type is Data. `NIWLANG_VAL_PAYLOAD_MAC_FRAME_TYPE_TRIGGER_FRAME (2)` Specifies that the MAC frame type is Trigger Frame.

NIWLANG_PAYLOAD_NUMBER_OF_MPDUS

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the number of medium access control (MAC) protocol data units (MPDUs) to combine into one aggregate-MPDU (A-MPDU). If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OF

DM, use an empty string as the active channel string format, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, or use 'userx' as the active channel string format if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or use 'userx' as the active channel string format if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU to configure this attribute.

Note: The toolkit ignores this attribute if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE or if you set the NIWLANG_STANDARD attribute to a value other than NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM; or if you set the NIWLANG_NON_HT_MODULATION_MODE attribute to NIWLANG_VAL_NON_HT_MODULATION_MODE_ON, or if you set the

NIWLANG_AUTO_NUMBER_OF_MPDUs attribute to NIWLANG_VAL_TRUE.

The default value is 1.

Set Function: niWLANG_SetNumberOfMPDUs
Get Function: niWLANG_GetNumberOfMPDUs

NIWLANG_PAYLOAD_DATA_LENGTH

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the length of the payload, excluding the length of the medium access control (MAC) header and frame check sequence (FCS). This value is expressed in bytes. The payload length encoded into the physical layer (PHY) header is the sum of the payload data length and the length of the MAC header and FCS.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, use an empty string active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string

format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute is to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if the you set NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE and use the niWLANG_CreateTriggerFrameMSDU to generate trigger frame.

The default values are as follows:

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, the default value is 1,024, and the limits on MPDU length are 0 to 4,095 (inclusive) and the limits on PSDU length are 0 to 65,535 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, the default value is 4,096, and the limits on PSDU length are 0 to 65,535 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OF

DM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, the default value is 4,096 and the limits on MPDU length are 0 to 16,383 (inclusive) and the limits on PSDU length are 0 to 46,92,480 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, the default value is 4,096, and the limits on PSDU length are 0 to 46,92,480 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, the default value is 256 and the limits on MPDU length are 0 to 16,383 (inclusive) and the limits on PSDU length are 0 to 797,159 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, the default value is 256, and the limits on PSDU length are 0 to 511 (inclusive).

If you set the NIWLANG_STANDARD attribute to any other value, the default value is 1,024 and valid values are 0 to 4,095 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDMand the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, the default value is 4096, the limits on MPDU length are 0 to 16,383 (inclusive) and the limits on PSDU length are 0 to 1,065,600 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMO_OF

DMand the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, the default value is 4096 and the limits on PSDU length are 0 to 1,065,600 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, the default value is 4096, the limits on MPDU length are 0 to 16,383 (inclusive) and the limits on PSDU length are 0 to 65,00,631 (inclusive).

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, the default value is 4096 and the limits on PSDU length are 0 to 65,00,631 (inclusive). If you set the NIWLANG_AUTO_DATA_LENGTH attribute to NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_L_SIG_LENGTH or NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_FRAME_DURATION, the toolkit ignores this attribute.

Set Function: niWLANG_SetPayloadDataLength
Get Function: niWLANG_GetPayloadDataLength

NIWLANG_AUTO_PAYLOAD_DATA_LENGTH_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to use the value specified in NIWLANG_PAYLOAD_DATA_LENGTH attribute or automatically compute the data length of the MPDUs in an AMPDU. The default value is

NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_DISABLED. Set Function: niWLANG_SetAutoPayloadDataLengthMode Get Function: niWLANG_GetAutoPayloadDataLengthMode

Values:

NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_DISABLED (0)	Specifies that the toolkit uses the value specified in NIWLANG_PAYLOAD_DATA_LENGTH attribute.
NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_L_SIG_LENGTH (1)	Specifies that the toolkit computes the data length of the MPDUs in an AMPDU of 802.11ax Trigger-Based PPDU using the following trigger frame parameters; NIWLANG_L_SIG_LENGTH, NIWLANG_PRE_FEC_PADDING_FACTOR, NIWLANG_PE_DISAMBIGUITY, and NIWLANG_LDPC_EXTRA_SYMBOL_SEGMENT attribute.
NIWLANG_VAL_AUTO_PAYLOAD_DATA_LENGTH_MODE_FRAME_DURATION (2)	Specifies that the toolkit computes the data length of the MPDUs in an AMPDU using the value specified in NIWLANG_FRAME_DURATION attribute.

NIWLANG_PAYLOAD_DATA_TYPE

Data Type: int32

Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the type of payload for waveform generation. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute: An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE. 'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel

	string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. The default value is NIWLANG_VAL_PN_SEQUENCE. Set Function: niWLANG_SetPayloadDataType Get Function: niWLANG_GetPayloadDataType
Values:	

`NIWLANG_VAL_USER_DEFINED (0)`	Specifies that the toolkit uses the sequence of bits that you specify using the NIWLANG_PAYLOAD_USER_DEFINED_BITS attribute.
`NIWLANG_VAL_PN_SEQUENCE (1)`	Specifies that the toolkit generates the bits using a pseudonoise (PN) sequence with the PN seed and order that you specify using the NIWLANG_PAYLOAD_PN_SEED attribute and NIWLANG_PAYLOAD_PN_ORDER attribute.

NIWLANG_PAYLOAD_PN_ORDER

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description:	Specifies the order (length of memory) of the pseudorandom bit sequence (PRBS) generator. The generated sequence is repeated (2^PN order) - 1 bits. If you set the NIWLANG_PAYLOAD_DATA_TYPE attribute to NIWLANG_VAL_USER_DEFINED, the toolkit ignores the NIWLANG_PAYLOAD_PN_ORDER attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute: An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE. 'mpdux' as the active channel string format if

you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE

	_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. The default value is 9. Valid values are 5 to 31, inclusive. Set Function: niWLANG_SetPayloadPNOrder Get Function: niWLANG_GetPayloadPNOrder

NIWLANG_PAYLOAD_PN_SEED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the initialization seed used for the pseudorandom bit sequence (PRBS) generator. If you set the NIWLANG_PAYLOAD_DATA_TYPE attribute to NIWLANG_VAL_USER_DEFINED, the toolkit ignores NIWLANG_PAYLOAD_PN_SEED attribute. If you set the NIWLANG_NUMBER_OF_FRAMES attribute to a value greater than 1 and the reset parameter of the niWLANG_CreateWaveformComplexF64 function or the NIWLANG_CreateMIMOWaveformsComplexF64 function to NIWLANG_VAL_FALSE, the toolkit uses the PRBS generator state at the end of the payload in frame n as the seed for frame n + 1. If

you set the reset parameter of the niWLANG_CreateWaveformComplexF64 function or the NIWLANG_CreateMIMOWaveformsComplexF64 function to NIWLANG_VAL_TRUE, all frames use the value of the NIWLANG_PAYLOAD_PN_SEED attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to

NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0xD6BF7DF2.

Set Function: niWLANG_SetPayloadPNSeed
Get Function: niWLANG_GetPayloadPNSeed

NIWLANG_PAYLOAD_USER_DEFINED_BITS

Data Type:	int32 []
Access:	read/write
Functions:	`niWLANG_SetVectorAttributeI32` `niWLANG_GetVectorAttributeI32`
Description:	Specifies a user-defined bit pattern as an array of zeros and ones. If the array length is greater than the configured payload length, the toolkit uses a subset of the required length from the beginning of the array for waveform generation. If the array length is less than the configured payload length, the toolkit repeats the user-defined bit pattern until the required length is achieved. If you set the NIWLANG_PAYLOAD_DATA_TYPE attribute to NIWLANG_VAL_PN_SEQUENCE, the toolkit ignores the NIWLANG_PAYLOAD_USER_DEFINED_BITS attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or

NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, use an empty string active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, use the following active channel string formats to configure this attribute:
An empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute is to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if the you set NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to

NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default is an empty array. Valid values include arrays of zeros and ones.

Set Function:
niWLANG_SetPayloadUserDefinedBits
Get Function:
niWLANG_GetPayloadUserDefinedBits

NIWLANG_MAC_FRAME_FORMAT

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the medium access control (MAC) frame is long or short. The default value is NIWLANG_VAL_MAC_FRAME_FORMAT_LONG. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use the following active channel string formats to configure this attribute. An empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE. Set Function: niWLANG_SetMACFrameFormat Get Function: niWLANG_GetMACFrameFormat
Values:	

`NIWLANG_VAL_MAC_FRAME_FORMAT_LONG (0)`	Specifies that the MAC frame format is long. The long format follows the general frame structure as defined in section 8.2.3 of IEEE Standard 802.11-2012.
`NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT (1)`	Specifies that the MAC frame format is short. The short format follows the short frame

structure as defined in section 8.8 of IEEE Standard P802.11ah/D1.3.

NIWLANG_MAC_HEADER_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable the medium access control (MAC) header, as defined in section 7.1.2 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013, IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string

format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is NIWLANG_VAL_TRUE.

Set Function: niWLANG_SetMACHeaderEnabled
Get Function: niWLANG_GetMACHeaderEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_MAC_FRAME_CONTROL

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the two-byte frame control field of the medium access control (MAC) header as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) at the rightmost position.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to

NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default values are as follows:

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_GENERAL_FRAME, the default value is 0x0.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_DATA_FRAME, the default value is 0x8.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the default value is 0x24.

Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

Set Function: niWLANG_SetMACFrameControl
Get Function: niWLANG_GetMACFrameControl

NIWLANG_MAC_DURATION_OR_ID

Data Type:	int32
Access:	read/write

Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the two-byte duration ID field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013, IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit (LSB) in the rightmost position. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, use an empty string active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, use the following active channel string formats to configure this attribute: Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE. Use the 'mpdux' active channel string format if you the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute is to NIWLANG_VAL_TRUE. Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if the you set NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_

PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Note: The toolkit ignores this attribute if you set the NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT.

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

Note: The toolkit ignores this attribute, if you set the NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT.

Set Function: niWLANG_SetMACDurationOrID
Get Function: niWLANG_GetMACDurationOrID

NIWLANG_MAC_ADDRESS1_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable the Address1 field of the medium access control (MAC) header. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to

NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to

NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is NIWLANG_VAL_TRUE.

Set Function:
niWLANG_SetMACAddress1Enabled
Get Function:
niWLANG_GetMACAddress1Enabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.

`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_MAC_ADDRESS1_LENGTH

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the length of Address1 field when the NIWLANG_MAC_FRAME_FORMAT attribute is set to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use the following active channel string formats to configure this attribute.
An empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE.

The default value is 2. Valid values are 2 and 6.

Set Function: niWLANG_SetMACAddress1Length
Get Function: niWLANG_GetMACAddress1Length

NIWLANG_MAC_ADDRESS1

Data Type: int64

Access: read/write

Functions: `niWLANG_SetScalarAttributeI64` `niWLANG_GetScalarAttributeI64`

Description:

Specifies the Address1 field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. The length of this field is defined by the NIWLANG_MAC_ADDRESS1 Length attribute if you set the NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT; otherwise, the length is six bytes. If this field is a MAC address, it is represented with the least significant byte in the leftmost position and each byte is represented with the least significant bit in the rightmost position.

For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0 x 123456789ABC.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to

NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

Set Function: niWLANG_SetMACAddress1
Get Function: niWLANG_GetMACAddress1

NIWLANG_MAC_ADDRESS2_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable the Address2 field of the medium access control (MAC) header.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM,

NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFD

M, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is NIWLANG_VAL_TRUE.

Set Function:
niWLANG_SetMACAddress2Enabled

Get Function: niWLANG_GetMACAddress2Enabled

Values:

NIWLANG_VAL_FALSE (0)	Disables the attribute.
NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_MAC_ADDRESS2_LENGTH

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the length of Address2 field if you set the NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use the following active channel string formats to configure this attribute.
An empty string active channel string format if you set the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE.

The default value is 2. Valid values are 2 and 6.

Set Function: niWLANG_SetMACAddress2Length
Get Function: niWLANG_GetMACAddress2Length

NIWLANG_MAC_ADDRESS2

Data Type: int64

Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI64` `niWLANG_GetScalarAttributeI64`
Description:	Specifies the Address2 field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. The length of this field is defined by the NIWLANG_MAC_ADDRESS2_LENGTH attribute if you set the NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT; otherwise, the length is six bytes. If this field is a MAC address, it is represented with the least significant byte in the leftmost position and each byte is represented with the least significant bit in the rightmost position. For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0 x 123456789ABC. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the

NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to

NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

Set Function: niWLANG_SetMACAddress2
Get Function: niWLANG_GetMACAddress2

NIWLANG_MAC_ADDRESS3_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description:

Specifies whether to enable the Address3 field of the medium access control (MAC) header.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE

attribute to
NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the
NIWLANG_AMPDU_ENABLED attribute to
NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to
NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the
NIWLANG_AMPDU_ENABLED attribute to
NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the
NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the
NIWLANG_AMPDU_ENABLED attribute to
NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU or
NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the
NIWLANG_AMPDU_ENABLED attribute to
NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU or
NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the
NIWLANG_AMPDU_ENABLED attribute to
NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the
NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_MU_PPDU or
NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_

	PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. The default value is NIWLANG_VAL_TRUE. If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute. Set Function: niWLANG_SetMACAddress3Enabled Get Function: niWLANG_GetMACAddress3Enabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_MAC_ADDRESS3

Data Type:	int64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI64` `niWLANG_GetScalarAttributeI64`
Description:	Specifies the six-byte Address3 field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position. For example, the medium access control (MAC) address 12-34-56-78-9A-BC is represented by the number 0x123456789ABC.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACAddress3
Get Function: niWLANG_GetMACAddress3

NIWLANG_MAC_SEQUENCE_CONTROL_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable the Sequence Control field of the medium access control (MAC) header.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the

NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to

NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is NIWLANG_VAL_TRUE.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACSequenceControlEnabled
Get Function: niWLANG_GetMACSequenceControlEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.

NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_MAC_SEQUENCE_CONTROL

Data Type: int32

Access: read/write

Functions: niWLANG_SetScalarAttributeI32
niWLANG_GetScalarAttributeI32

Description: Specifies the two-byte Sequence Control field as defined in section 7.1.3 of IEEE Standard 802.11-2007, IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0.

This field is represented with the least significant bit in the rightmost position.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE

attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant 2 bytes.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACSequenceControl
Get Function: niWLANG_GetMACSequenceControl

NIWLANG_MAC_ADDRESS4_ENABLED

Data Type:	int32
Access:	read/write

Functions:

`niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:

Specifies whether to the enable Address4 field of the medium access control (MAC) header.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to

	NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. The default value is NIWLANG_VAL_TRUE. If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute. Set Function: niWLANG_SetMACAddress4Enabled Get Function: niWLANG_GetMACAddress4Enabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_MAC_ADDRESS4

Data Type:	int64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI64` `niWLANG_GetScalarAttributeI64`
Description:	Specifies the six-byte Address4 field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant byte in the leftmost position, and each byte is represented with the least significant bit in the rightmost position. For example, the medium access control (MAC)

address 12-34-56-78-9A-BC is represented by the number 0 x 123456789ABC.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE

attribute to
NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to
NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_

PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFFFFFF. For values outside this range, the toolkit uses the least significant six bytes.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACAddress4
Get Function: niWLANG_GetMACAddress4

NIWLANG_MAC_QOS_CONTROL_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable the QoS Control field of the medium access control (MAC) header. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or

NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to

NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is NIWLANG_VAL_FALSE. If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function:
niWLANG_SetMACQOSControlEnabled
Get Function:
niWLANG_GetMACQOSControlEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.

NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_MAC_QOS_CONTROL

Data Type: int32

Access: read/write

Functions: niWLANG_SetScalarAttributeI32
niWLANG_GetScalarAttributeI32

Description: Specifies the two-byte QoS Control field as defined in section 7.1.3 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit in the rightmost position.

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string

formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

Note: If you set NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT or the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_MAC_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores the NIWLANG_MAC_QOS_CONTROL attribute.

The default value is 0x0. Valid values are 0x0 to 0xFFFF. For values outside this range, the toolkit uses the least significant two bytes.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACQOSControl
Get Function: niWLANG_GetMACQOSControl

NIWLANG_MAC_HT_CONTROL_ENABLED

Data Type: int32

Access: read/write

Functions:
`niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:
Specifies whether to enable the HT Control field of the medium access control (MAC) header.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:

An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.

	Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE. Note: If you set NIWLANG_MAC_FRAME_FORMAT attribute to NIWLANG_VAL_MAC_FRAME_FORMAT_SHORT or the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_TRIGGER_FRAME, the toolkit ignores the NIWLANG_MAC_HT_CONTROL_ENABLED attribute. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetMACHTControlEnabled Get Function: niWLANG_GetMACHTControlEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_MAC_HT_CONTROL

Data Type:	int32
Access:	read/write

Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the four-byte HT Control field as defined in section 7.1.3 of IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0. This field is represented with the least significant bit in the rightmost position. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute: An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.

'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the

NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

The default value is 0x0. Valid values are 0x0 to 0xFFFFFFFF. For values outside this range, the toolkit uses the least significant four bytes.

If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACHTControl
Get Function: niWLANG_GetMACHTControl

NIWLANG_MAC_SEQUENCE_NUMBER_INCREMENT_EN

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies whether to increment the sequence number in a sequence of frames.

The default value is NIWLANG_VAL_FALSE. If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute.

Set Function: niWLANG_SetMACSequenceNumberIncrementE

	nabled Get Function: niWLANG_GetMACSequenceNumberIncrementEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_MAC_SEQUENCE_NUMBER_INCREMENT_IN

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the number of frames after which the toolkit increments the sequence number by 1. The starting number is the value represented by the sequence number sub-field of the Sequence Control field. The sequence number is wrapped to 0 after reaching the value 4,095 or (2^12-1). The toolkit ignores this attribute if you set the NIWLANG_MAC_SEQUENCE_NUMBER_INCREMENT_ENABLED attribute to NIWLANG_VAL_FALSE and the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME. The default value is 1. Valid values are 0 to 4,095, inclusive. Set Function: niWLANG_SetMACSequenceNumberIncrementInterval Get Function: niWLANG_GetMACSequenceNumberIncrementInterval

NIWLANG_MAC_FRAGMENT_NUMBER_INCREMENT_EN

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to increment fragment number in a sequence of frames. The starting number is the value represented by the fragment number sub-field of the NIWLANG_MAC_SEQUENCE_CONTROL attribute. If you set the NIWLANG_MAC_FRAGMENT_NUMBER_INCREMENT_ENABLED attribute to NIWLANG_VAL_TRUE, the fragment number increments by 1 for every successive frame having the same sequence number. The fragment number wraps to the starting number when the sequence number increments. The fragment number wraps to 0 after reaching the value 15. The default value is NIWLANG_VAL_FALSE. If you set the NIWLANG_PAYLOAD_MAC_FRAME_TYPE attribute to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME, the toolkit ignores this attribute. Set Function: niWLANG_SetMACFragmentNumberIncrementEnabled Get Function: niWLANG_GetMACFragmentNumberIncrementEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.

`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_MPDU_LENGTH

Data Type:	int32
Access:	read only
Functions:	`niWLANG_GetScalarAttributeI32`
Description:	Returns the length of the medium access control (MAC) protocol data unit (MPDU). This value is expressed in bytes. An MPDU comprises of a MAC header, a frame body, and a frame check sequence (FCS). The NIWLANG_MPDU_LENGTH attribute is the sum of the length of MAC header, the value of the NIWLANG_PAYLOAD_DATA_LENGTH attribute, and the length of FCS, which is four bytes. If you disable the NIWLANG_MAC_HEADER_ENABLED and NIWLANG_MAC_FCS_ENABLED attributes, the lengths of MAC header and FCS are zero. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to query this attribute. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active

channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to query this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to query this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to query this attribute:

Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

Get Function: niWLANG_GetMPDULength

NIWLANG_MAC_FCS_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable the medium access control (MAC) frame check sequence (FCS), as defined in section 7.1.2 of IEEE Standard 802.11-2007 and IEEE Standard 802.11n-2009, section 8.2.4 of IEEE Standard 802.11ac-2013 and IEEE P802.11ah/D1.3, and section 9.2.4 of IEEE P802.11ax/D6.0.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, you must use an empty string to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, you must use an empty string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE, to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
'mpdux' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to

NIWLANG_VAL_TRUE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE, or use the 'mpdux' active channel string if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use the following active channel string formats to configure this attribute:
Use an empty string active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_FALSE.
Use the 'mpdux' active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.
Use the 'userx/mpduy' as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU and the NIWLANG_AMPDU_ENABLED attribute to NIWLANG_VAL_TRUE.

Set Function: niWLANG_SetMACFCSEnabled
Get Function: niWLANG_GetMACFCSEnabled

Values:

NIWLANG_VAL_FALSE (0)	Disables the attribute.
NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_TRIGGER_FRAME_AID12

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the AID12 field in the trigger frame.

The valid values are 1 to 2007, which are used for indication of RUs used by Trigger-based PPDU.

The values 0 and 2045 are used for Random Access RU information specification, the value 2046 is used for unassigned RU location indication, and the value 4095 is reserved for trigger frame padding indication.

When you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, and the NIWLANG_PPDU_TYPE attribute is NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, use 'userx' as active channel string formats to configure this attribute.

The default value is 1.

Set Function: niWLANG_SetAID12
Get Function: niWLANG_GetAID12

NIWLANG_TRIGGER_FRAME_AP_TX_POWER

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the AP Tx Power field of the Trigger frame. The power values -20 dBm to 40 dBm are mapped to the field values 0 to 60 respectively.

The default value is 0. The valid values are 0 to 63, inclusive.

Set Function: niWLANG_SetAPTXPower
Get Function: niWLANG_GetAPTXPower

NIWLANG_TRIGGER_FRAME_CS_REQUIRED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the CS required sub-field in the 802.11ax Trigger Frame.

The default value is 0. The valid values are 0 and 1.

Set Function: niWLANG_SetCSRequired
Get Function: niWLANG_GetCSRequired

NIWLANG_TRIGGER_FRAME_TARGET_RSSI

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the UL-Target RSSI field of the Trigger frame. The power values -110 dBm to -20 dBm are mapped to the field values 0 to 90, respectively. To specify maximum transmit power for the assigned MCS, you must set a value of 127.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use 'userx' as the active channel string format to configure this attribute if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

The default value is 78, which is the corresponding power value -32 dBm. The valid values are 0 to 127,inclusive.

Set Function: niWLANG_SetTargetRSSI
Get Function: niWLANG_GetTargetRSSI

NIWLANG_TRIGGER_FRAME_MAC_PADDING_DURATIO

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the padding duration when the NIWLANG_FRAME_TYPE attribute is set to NIWLANG_VAL_PAYLOAD_FRAME_TYPE_TRIGGER_FRAME.

This attribute is valid, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM,

NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM use an empty string active channel string format to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE, or use 'mpdux' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE to configure this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use an empty string as the active channel string format, if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_FALSE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or use 'mpdux' as the active channel string format if the NIWLANG_AMPDU_ENABLED attribute is set to NIWLANG_VAL_TRUE and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU to configure this attribute.

The default value is NIWLANG_VAL_MAXIMUM_PADDING_DURATION_0US. Set Function:

	niWLANG_SetMaximumMACPaddingDuration Get Function: niWLANG_GetMaximumMACPaddingDuration
Values:	<table><tr><td>NIWLANG_VAL_PADDING_DURATION_0US (0)</td><td>Specifies that there is no padding.</td></tr><tr><td>NIWLANG_VAL_PADDING_DURATION_8US (1)</td><td>Specifies that the padding duration is 8us.</td></tr><tr><td>NIWLANG_VAL_PADDING_DURATION_16US (2)</td><td>Specifies that the padding duration is 16us.</td></tr></table>

NIWLANG_NUMBER_OF_DATA_SYMBOLS

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Returns the number of symbols in the data portion of the generated WLAN frame. Symbol refers to the chip if the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211BG_DSSS, and the symbol refers to the OFDM symbol for other values of the NIWLANG_STANDARD attribute. Set Function: niWLANG_SetNumberOfDataSymbols Get Function: niWLANG_GetNumberOfDataSymbols

NIWLANG_SUBCARRIER_MASK

Data Type:	float64 []

Access: read/write

Functions: `niWLANG_SetVectorAttributeF64`
`niWLANG_GetVectorAttributeF64`

Description: Specifies the sequence of attenuation values on each subcarrier in the signal and payload symbols if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. You must specify a 64-element array. The first element of the array corresponds to subcarrier index-32, and the 64th element corresponds to subcarrier index 31, as defined in section 17.3.2.5 in IEEE Standard 802.11a-1999.

If an element has a value of 0, this indicates that the corresponding subcarrier is absent in the generated signal. If an element has a value of 1, this indicates that the corresponding subcarrier is present in the generated signal.

The default value is a 64-element array of all ones. Valid values are 64-element arrays of zeros and ones.

Set Function: niWLANG_SetSubcarrierMask
Get Function: niWLANG_GetSubcarrierMask

NIWLANG_LOCKED_CLOCK_BIT_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:	Specifies whether to enable the Locked Clock Bit flag for the direct sequence spread spectrum (DSSS) header, as defined in sections 18.2.3.4 and 18.2.3.11 of IEEE Standard 802.11b-1999, and section 19.3.2.1 of IEEE Standard 802.11g-2003. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetLockedClockBitEnabled Get Function: niWLANG_GetLockedClockBitEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Sets the locked clocks bit to 0.
`NIWLANG_VAL_TRUE (1)`	Sets the locked clocks bit to 1.

NIWLANG_HEADER_ENCODER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable convolutional encoding of the OFDM SIGNAL field, as defined in section 17.3.5.5 of IEEE Standard 802.11a-1999. Note: Configure the NIWLANG_HEADER_ENCODER_ENABLED attribute only when you set the

	NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. The default value is NIWLANG_VAL_TRUE. Set Function: niWLANG_SetHeaderEncoderEnabled Get Function: niWLANG_GetHeaderEncoderEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Disables convolutional encoding and repeats each input bit twice.
`NIWLANG_VAL_TRUE (1)`	Enables convolutional encoding.

NIWLANG_HEADER_INTERLEAVER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable interleaving for the OFDM SIGNAL field, as defined in section 17.3.5.6 of IEEE Standard 802.11a-1999. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or

	NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. The default value is NIWLANG_VAL_TRUE. Set Function: niWLANG_SetHeaderInterleaverEnabled Get Function: niWLANG_GetHeaderInterleaverEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_PAYLOAD_SCRAMBLER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable scrambling of the payload for OFDM packets and the entire burst for direct sequence spread spectrum (DSSS) packets, as defined in section 17.3.5.4 of IEEE Standard 802.11a-1999 and section 18.2.4 of IEEE Standard 802.11b-1999. Note: If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OF

	DM, the toolkit ignores the NIWLANG_PAYLOAD_SCRAMBLER_ENABLED attribute and always enables the payload scrambler. The default value is NIWLANG_VAL_TRUE. Set Function: niWLANG_SetPayloadScramblerEnabled Get Function: niWLANG_GetPayloadScramblerEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_PAYLOAD_SCRAMBLER_SEED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the initial state of the scrambler seed. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, you must use an empty string to configure

this attribute.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string syntax to configure this attribute.

An empty string as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU.
'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use the following active channel string syntax to configure this attribute.

An empty string as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU.
'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

For direct sequence spread spectrum (DSSS) packets, the default value follows the requirements defined in sections 18.2.3.1 and section 18.2.3.8 of IEEE Standard 802.11b-1999.

For OFDM and DSSS-OFDM packets, the default value is 93.

Set Function:
niWLANG_SetPayloadScramblerSeed

Get Function:
niWLANG_GetPayloadScramblerSeed

NIWLANG_PAYLOAD_ENCODER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable convolutional encoding of the OFDM payload, as defined in section 17.3.5.5 of IEEE Standard 802.11a-1999. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. The default value is NIWLANG_VAL_TRUE. Set Function: niWLANG_SetPayloadEncoderEnabled Get Function: niWLANG_GetPayloadEncoderEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_FEC_CODING_TYPE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the type of forward error correction (FEC) coding to use if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, as defined in section 20.3.11.3 of IEEE Standard 802.11n-2009. If the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, you must use an empty string as the active channel string format to configure this attribute. If the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string syntax to configure this attribute. An empty string as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU. 'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use the following active channel string syntax to configure this attribute.

An empty string as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU.
'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

The default value is NIWLANG_VAL_FEC_CODING_TYPE_LDPC, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, or NIWLANG_VAL_FEC_CODING_TYPE_BCC otherwise.

Set Function: niWLANG_SetFECCodingType
Get Function: niWLANG_GetFECCodingType

Values:

`NIWLANG_VAL_FEC_CODING_TYPE_BCC (0)`	Specifies that the FEC coding type is binary convolutional code (BCC).
`NIWLANG_VAL_FEC_CODING_TYPE_LDPC (1)`	Specifies that the FEC coding type is low density parity check (LDPC).

NIWLANG_PAYLOAD_INTERLEAVER_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable interleaving for the OFDM payload, as defined in section 17.3.5.6 of

IEEE Standard 802.11a-1999.

Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM.

The default value is NIWLANG_VAL_TRUE.

Set Function: niWLANG_SetPayloadInterleaverEnabled
Get Function: niWLANG_GetPayloadInterleaverEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_PPDU_TYPE

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the type of physical layer convergence procedure (PLCP) protocol data unit (PPDU), if the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM.

The default value is

	NIWLANG_VAL_PPDU_TYPE_SU_PPDU. Set Function: niWLANG_SetPPDUType Get Function: niWLANG_GetPPDUType
Values:	

`NIWLANG_VAL_PPDU_TYPE_SU_PPDU (0)`	Specifies that the toolkit generates a single user (SU) PPDU.
`NIWLANG_VAL_PPDU_TYPE_MU_PPDU (1)`	Specifies that the toolkit generates a multi-user (MU) PPDU.
`NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU (2)`	Specifies that the toolkit generates an extended range single user (ER SU) PPDU.
`NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (3)`	Specifies that the toolkit generates a trigger-based (TB) PPDU.

NIWLANG_RU_ALLOCATION_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies how to configure the multi-user allocation in a 802.11ax MU PPDU signal. The default value is NIWLANG_VAL_RU_ALLOCATION_MODE_INDIVIDUAL. Set Function: niWLANG_SetRUAllocationMode Get Function: niWLANG_GetRUAllocationMode
Values:	

`NIWLANG_VAL_RU_ALLOCATION_MODE_INDIVIDUAL (0)`	Users are configured using the NIWLANG_RU_SIZE and the

	NIWLANG_RU_OFFSET attributes.
`NIWLANG_VAL_RU_ALLOCATION_MODE_GROUP (1)`	Users are configured as per the common field of the HE-SIG-B using the NIWLANG_RU_ALLOCATION attribute.

NIWLANG_RU_SIZE

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the size of resource unit (RU) in terms of the number of subcarriers for the 802.11ax signal. You must configure this attribute you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

When you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, this attribute supports only NIWLANG_VAL_RU_SIZE_242 and NIWLANG_VAL_RU_SIZE_106 as valid values.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use 'userx' as the active channel string format, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, or use an empty string as the string format if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE

_SU_PPDU to configure this attribute.

The default value is NIWLANG_VAL_RU_SIZE_26.

Set Function: niWLANG_SetRUSize
Get Function: niWLANG_GetRUSize

Values:

`NIWLANG_VAL_RU_SIZE_26 (26)`	Specifies that the RU size is 26 subcarriers.
`NIWLANG_VAL_RU_SIZE_52 (52)`	Specifies that the RU size is 52 subcarriers.
`NIWLANG_VAL_RU_SIZE_106 (106)`	Specifies that the RU size is 106 subcarriers.
`NIWLANG_VAL_RU_SIZE_242 (242)`	Specifies that the RU size is 242 subcarriers.
`NIWLANG_VAL_RU_SIZE_484 (484)`	Specifies that the RU size is 484 subcarriers.
`NIWLANG_VAL_RU_SIZE_996 (996)`	Specifies that the RU size is 996 subcarriers.
`NIWLANG_VAL_RU_SIZE_2x996 (1992)`	Specifies that the RU size is 1992 subcarriers.

NIWLANG_RU_OFFSET

Data Type: int32

Access: read/write

Functions:
`niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the location of the resource unit (RU), in terms of the index of 26-tone RU, assuming the entire bandwidth is composed of 26-tone RUs in the 802.11ax signal.

Refer to the Configuring RU Offset topic in NI WLAN Generation Toolkit Help for more information.

This attribute is valid only if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

The valid values are as follows:

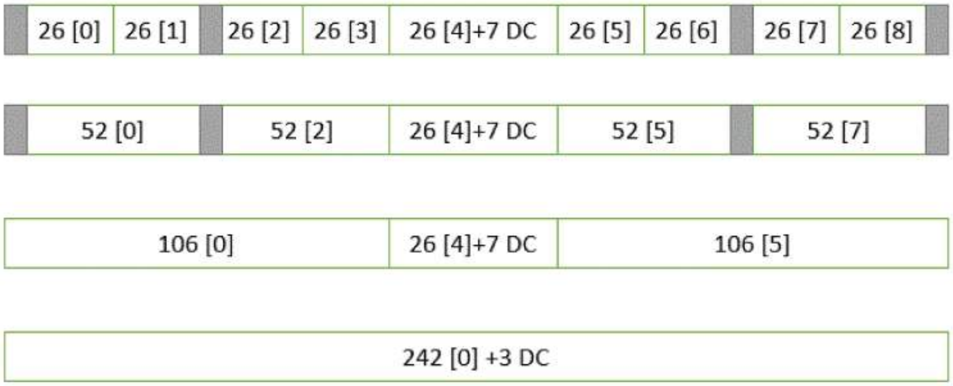
If the NIWLANG_CHANNEL_BANDWIDTH is 20 MHz and the NIWLANG_NUMBER_OF_SEGMENTS is 1, the valid values are 0 to 8.
If the NIWLANG_CHANNEL_BANDWIDTH is 40 MHz and the NIWLANG_NUMBER_OF_SEGMENTS is 1, the valid values are 0 to 17.
If the NIWLANG_CHANNEL_BANDWIDTH is 80 MHz and the NIWLANG_NUMBER_OF_SEGMENTS is 1, the valid values are 0 to 36.
If the NIWLANG_CHANNEL_BANDWIDTH is 160 MHz and the NIWLANG_NUMBER_OF_SEGMENTS is 1, the valid values are 0 to 73.
If the NIWLANG_CHANNEL_BANDWIDTH is 80 MHz and the NIWLANG_NUMBER_OF_SEGMENTS is 2, the valid values are 0 to 73.

The default value is 0.

Set Function: niWLANG_SetRUOffset
Get Function: niWLANG_GetRUOffset

Configuring RU Offset

The following figure is an example that illustrates that to specify 106-tone RU second from left you must configure the NIWLANG_RU_OFFSET attribute to 5. The value in the bracket indicates the RU offset.



In the preceding image,

- The shaded area indicates null subcarriers
- The square bracket value indicates the RU offset

NIWLANG_RU_ALLOCATION

Data Type: int32 []

Access: read/write

Functions: `niWLANG_SetVectorAttributeI32`
`niWLANG_GetVectorAttributeI32`

Description: Specifies the common field of the HE-SIG-B to be used for multi-user allocation in a 802.11ax MU PPDU signal.

Each element of the array represents the RU allocation sub-field of the common field corresponding to each 20 MHz sub-channel in the channel bandwidth.

The NIWLANG_RU_SIZE and the NIWLANG_RU_OFFSET attributes of the configured users are derived based on the value of this attribute. Query the number of users using the niWLANG_GetNumberOfUsersFromRUAllocation function to configure the user related attributes. The users are indexed in the increasing order of frequency.

When the NIWLANG_CHANNEL_BANDWIDTH attribute is set to 80 MHz or 160 MHz, the 26 Resource Units (RU) at the center are enabled by default. These users can be disabled by using the NIWLANG_USER_ENABLED attribute.

The toolkit ignores this attribute if you set the NIWLANG_RU_ALLOCATION_MODE attribute to NIWLANG_VAL_RU_ALLOCATION_MODE_INDIVIDUAL.

The default value in the array is 0d128.

NIWLANG_USER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable the user in a 802.11ax MU PPDU signal.

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_POWER_BOOST_FACTOR

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`

Description:

Specifies the factor, per resource unit (RU) by which the amplitude of the HE modulated fields in 802.11ax signals are scaled, when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU. The value of this attribute must be the same across all users within an RU. An RU is defined by the NIWLANG_RU_SIZE and NIWLANG_RU_OFFSET attributes.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_OFDM, you muse use 'userx' as the active channel string format to configure this attribute.

The default value is 1. The valid values are 0.1 to 10, inclusive. To convert this value to dB, use the following formula:

Power Boost Factor (dB) = 20 * log (Power Boost Factor)

Set Function: niWLANG_SetPowerBoostFactor
Get Function: niWLANG_GetPowerBoostFactor

NIWLANG_RELATIVE_POWER

Data Type: float64

Access: read/write

Functions:

`niWLANG_SetScalarAttributeF64`
`niWLANG_GetScalarAttributeF64`

Description:

Specifies the per user power scaling value of the 802.11ax signal when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU. This value is expressed in dB. The power scaling value is with reference to the user with index 0. The toolkit ignores the attribute value, if it is specified for the user with index 0.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_OFDM, you must use 'userx' as the active channel string format to configure this attribute.

The default value is 0. The valid values are -100 to 100, inclusive.

Set Function: niWLANG_SetRelativePower
Get Function: niWLANG_GetRelativePower

NIWLANG_NON_HT_MODULATION_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the format of the incoming OFDM signal is non-high throughput (HT). This attribute is valid only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM. A non-HT format signal has only L-LTF, L-STF, and L-SIG symbols in the preamble, which are similar to 802.11n, 802.11ac, or 802.11af signals with same bandwidth. The payload is modulated in the same manner as an 802.11a signal with bandwidth of 20 MHz. The payload is repeated with appropriate tone rotation to fill the channel bandwidth. The default value is

	NIWLANG_VAL_NON_HT_MODULATION_MODE_OFF. Set Function: niWLANG_SetNonHTModulationMode Get Function: niWLANG_GetNonHTModulationMode
Values:	NIWLANG_VAL_NON_HT_MODULATION_MODE_OFF (0) — Specifies that the signal is not of non-HT format. NIWLANG_VAL_NON_HT_MODULATION_MODE_ON (1) — Specifies that the signal is of non-HT format.

NIWLANG_OFDM_DATA_RATE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the data rate for the OFDM payload, as defined in section 17.3.2.2 of IEEE Standard 802.11-2007. This value is expressed in Mbps. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, or if you set the NIWLANG_NON_HT_MODULATION_MODE attribute to NIWLANG_VAL_NON_HT_MODULATION_MODE_ON. The default value is NIWLANG_VAL_OFDM_DATA_RATE_6.

Set Function: niWLANG_SetOFDMDataRate
Get Function: niWLANG_GetOFDMDataRate

Values:

`NIWLANG_VAL_OFDM_DATA_RATE_6 (0)`	Specifies a data rate of 1.5 Mbps, 3 Mbps, and 6 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
`NIWLANG_VAL_OFDM_DATA_RATE_9 (1)`	Specifies a data rate of 2.25 Mbps, 4.5 Mbps, and 9 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
`NIWLANG_VAL_OFDM_DATA_RATE_12 (2)`	Specifies a data rate of 3 Mbps, 6 Mbps, and 12 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
`NIWLANG_VAL_OFDM_DATA_RATE_18 (3)`	Specifies a data rate of 4.5 Mbps, 9 Mbps, and 18 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
`NIWLANG_VAL_OFDM_DATA_RATE_24 (4)`	Specifies a data rate of 6 Mbps, 12 Mbps, and 24 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
`NIWLANG_VAL_OFDM_DATA_RATE_36 (5)`	Specifies a data rate of 9 Mbps, 18 Mbps, and 36 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.

NIWLANG_VAL_OFDM_DATA_RATE_48 (6)	Specifies a data rate of 12 Mbps, 24 Mbps, and 48 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.
NIWLANG_VAL_OFDM_DATA_RATE_54 (7)	Specifies a data rate of 13.5 Mbps, 27 Mbps, and 54 Mbps for respective channel bandwidths of 5 MHz, 10 MHz, and 20 MHz.

NIWLANG_ACTUAL_OFDM_DATA_RATE

Data Type: float64

Access: read only

Functions: `niWLANG_GetScalarAttributeF64`

Description: Returns the OFDM data rate depending upon the values of the NIWLANG_CHANNEL_BANDWIDTH and NIWLANG_OFDM_DATA_RATE attributes, only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, or if you set the NIWLANG_NON_HT_MODULATION_MODE attribute to NIWLANG_VAL_NON_HT_MODULATION_MODE_ON. This value is expressed in Mbps. For more information about the OFDM data rate, refer to section 17.2.3.3 of IEEE Standard 802.11-2007.

	Get Function: niWLANG_GetActualOFDMDataRate

NIWLANG_DSSS_DATA_RATE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the data rate for the direct sequence spread spectrum (DSSS) payload, as defined in IEEE Standard 802.11b-1999 and the extended rate physical layer-packet binary convolutional coding (ERP -PBCC) mode, as defined in IEEE Standard 802.11g-2003. This value is expressed in Mbps. Note: Configure this attribute only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS. The default value is NIWLANG_VAL_DSSS_DATA_RATE_1. Set Function: niWLANG_SetDSSSDataRate Get Function: niWLANG_GetDSSSDataRate
Values:	

`NIWLANG_VAL_DSSS_DATA_RATE_1 (0)`	Specifies a data rate of 1 Mbps, as defined in sections 18.4.6.3 and 18.4.6.4 of IEEE Standard 802.11b-1999.
`NIWLANG_VAL_DSSS_DATA_RATE_2 (1)`	Specifies a data rate of 2 Mbps, as defined in sections 18.4.6.3 and 18.4.6.4 of IEEE Standard 802.11b-1999.

NIWLANG_VAL_DSSS_DATA_RATE_5p5_CCK (2)	Specifies a data rate of 5.5 Mbps complementary code keying (CCK), as defined in section 18.4.6.5 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_5p5_PBCC (3)	Specifies a data rate of 5.5 Mbps PBCC, as defined in section 18.4.6.6 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_11_CCK (4)	Specifies a data rate of 11 Mbps CCK, as defined in section 18.4.6.5 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_11_PBCC (5)	Specifies a data rate of 11 Mbps PBCC, as defined in section 18.4.6.6 of IEEE Standard 802.11b-1999.
NIWLANG_VAL_DSSS_DATA_RATE_22 (6)	Specifies a data rate of 22 Mbps, as defined in section 19.3.3.2 of IEEE Standard 802.11g-2003.
NIWLANG_VAL_DSSS_DATA_RATE_33 (7)	Specifies a data rate of 33 Mbps, as defined in section 19.3.3.2 of IEEE Standard 802.11g-2003.

NIWLANG_DSSS_PREAMBLE_TYPE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to use a long or short preamble for direct sequence spread spectrum (DSSS) and DSSS-OFDM packets, as defined in IEEE Standard 802.11b-1999. Note: Configure this attribute only if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM. The default value is NIWLANG_VAL_PREAMBLE_TYPE_LONG_PREAMBLE. Set Function: niWLANG_SetDSSSPreambleType Get Function: niWLANG_GetDSSSPreambleType
Values:	

`NIWLANG_VAL_PREAMBLE_TYPE_SHORT_PREAMBLE (0)`	Uses a short preamble as defined in section 18.2.2.2 of IEEE Standard 802.11b-1999.
`NIWLANG_VAL_PREAMBLE_TYPE_LONG_PREAMBLE (1)`	Uses a long preamble as defined in section 18.2.2.1 of IEEE Standard 802.11b-1999.

NIWLANG_STA_ID

Data Type:	int32
Access:	read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies 11 LSBs of the association identifier (AID) in a 802.11ax signal when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU.

When the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU, the valid values are 0 to 2047.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use 'userx' as the active channel string format to configure this attribute, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU.

The default value is 0.

Set Function: niWLANG_SetSTAID
Get Function: niWLANG_GetSTAID

NIWLANG_MCS_INDEX

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the modulation and coding scheme (MCS) index. The MCS index is a compact representation that determines the modulation scheme, coding rate, and number of spatial streams as specified in section 20.3.5 of IEEE Standard 802.11n-2009, section 22.5 of IEEE Standard 802.11ac-2013, section 24.5 of IEEE P802.11ah/D1.3, section 23.5 of IEEE Standard 802.11af-201, and section 27.5 of IEEE P802.11ax/D6.0.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM the active channel string syntax is an empty string.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, you must use the following active channel string formats to configure this attribute:
An empty string as the active channel string format if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU.
'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, you must use the following active channel string syntax to configure this attribute.

An empty string as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU.
'userx' as the active channel string syntax if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

The default value is 0.

For

NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, valid values are 0 to 11, inclusive.

For NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, valid values are 0 to 13, inclusive.

For NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, valid values are 0 to 32, inclusive.

For NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, valid values are 0 to 10, inclusive.

For NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, valid values are 0 to 9, inclusive.

Set Function: niWLANG_SetMCSIndex
Get Function: niWLANG_GetMCSIndex

NIWLANG_DCM_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the dual carrier modulation (DCM) is applied to the data part of the 802.11ax signals or not. The property can be set to NIWLANG_VAL_TRUE only when MCS index is 0, 1, 3, or 4 and number of spatial streams is 1 or 2. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use an empty string as the string format, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or

	NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU and 'userx' as the active channel string format, if the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU to configure this attribute. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetDCMEnabled Get Function: niWLANG_GetDCMEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_HE_SIG_B_MCS_INDEX

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the value of the modulation and coding scheme (MCS) index of the HE-SIG-B field of 802.11ax signal when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU. The default value is 0. The valid values are 0 to 5, inclusive. Set Function: niWLANG_SetHESIGBMCSIndex Get Function: niWLANG_GetHESIGBMCSIndex

NIWLANG_HE_SIG_B_DCM_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the dual carrier modulation (DCM) is applied on the HE-SIG-B field of 802.11ax signals or not. The property can be set to NIWLANG_VAL_TRUE only when HE-SIG-B MCS index is 0, 1, 3 or 4. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetHESIGBDCMEnabled Get Function: niWLANG_GetHESIGBDCMEnabled
Values:	`NIWLANG_VAL_FALSE (0)` — Disables the attribute. `NIWLANG_VAL_TRUE (1)` — Enables the attribute.

NIWLANG_80211N_PLCP_FRAME_FORMAT

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the format of the physical layer convergence protocol (PLCP) frame structure. The frame structure determines the arrangement of preambles, header (SIGNAL field), and payload in a frame as defined in section 20.3.2 of IEEE Standard 802.11n-2009. The default value is NIWLANG_VAL_80211N_PLCP_FRAME_FORMAT_MIXED.

	Set Function: niWLANG_Set80211nPLCPFrameFormat Get Function: niWLANG_Get80211nPLCPFrameFormat
Values:	

NIWLANG_VAL_80211N_PLCP_FRAME_FORMAT_MIXED (0)	Specifies that the PLCP frame structure consists of non-HT preamble and header (SIGnal field) followed by high throughput (HT) header, preambles, and payload as specified in IEEE Standard 802.11n-2009.
NIWLANG_VAL_80211N_PLCP_FRAME_FORMAT_GREENFIELD (1)	Specifies that the PLCP frame structure does not support non-HT, and starts with HT preamble, followed by HT SIGnal field and payload.

NIWLANG_80211AH_PREAMBLE_TYPE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the preamble type of packet if you set the NIWLANG_STANDARD attribute to 80211AH MIMOOFDM as defined in section 24.3.8.2 of IEEE Standard P802.11ah/D1.3. The default value is NIWLANG_VAL_80211AH_PREAMBLE_TYPE_SHORT.

	Set Function: niWLANG_Set80211ahPreambleType Get Function: niWLANG_Get80211ahPreambleType
Values:	<table><tr><td>NIWLANG_VAL_80211AH_PREAMBLE_TYPE_SHORT (0)</td><td>Specifies that the preamble type is S1G_Short.</td></tr><tr><td>NIWLANG_VAL_80211AH_PREAMBLE_TYPE_LONG (1)</td><td>Specifies that the preamble type is S1G_Long.</td></tr></table>

NIWLANG_TRANSMISSION_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the value of the uplink indication field of the S1G-SIG field when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM. This attribute also specifies whether the packet is uplink or downlink when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM. Refer to section 24.3.8.2.1.4 of IEEE Standard P802.11ah/D1.3 for more information about the 80211ah Uplink Indication. The default value is NIWLANG_VAL_TRANSMISSION_MODE_DL. Set Function: niWLANG_SetTransmissionMode Get Function: niWLANG_GetTransmissionMode

Values:

NIWLANG_VAL_TRANSMISSION_MODE_DL (0)	Specifies that the transmission mode is downlink (DL).
NIWLANG_VAL_TRANSMISSION_MODE_UL (1)	Specifies that the transmission mode is uplink (UL).

NIWLANG_PREAMBLE_PUNCTURING_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies whether to enable the preamble puncturing (channel puncturing) on 802.11ax MU PPDU signals. Preamble puncturing is valid only when you set the NIWLANG_CHANNEL_BANDWIDTH attribute to 80 MHz or 160 MHz.

The default value is NIWLANG_VAL_FALSE.

Set Function:
niWLANG_SetPreamblePuncturingEnabled
Get Function:
niWLANG_GetPreamblePuncturingEnabled

Values:

NIWLANG_VAL_FALSE (0)	Disables the attribute.
NIWLANG_VAL_TRUE (1)	Enables the attribute.

NIWLANG_PRIMARY_20MHZ_CHANNEL_INDEX

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the index of the primary 20 MHz sub-channel in the channel bandwidth. This attribute along with the puncturing information defines the mode of puncturing.

The default value is 0. For 80 MHz, valid values are 0 to 3, inclusive. For (80 + 80) MHz and 160 MHz, valid values are 0 to 7, inclusive.

Set Function:
niWLANG_SetPrimary20MHzChannelIndex
Get Function:
niWLANG_GetPrimary20MHzChannelIndex

NIWLANG_PREAMBLE_PUNCTURING_MASK

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the 20 MHz sub-channels to be punctured in the 802.11ax MU PPDU signal when preamble puncturing is enabled. The mask value specified here is a binary mask represented as an integer, where bit '0' represents the punctured sub-channel.

In the binary value, the least significant bit represents the 20 MHz sub-channel lower in frequency and the most significant bit represents the 20 MHz sub-channel higher in frequency. For (80 + 80) MHz case, LSB represents the lowest sub-channel in first segment. For 80M case, the toolkit considers the least significant 4 bits as the mask value. For (80 + 80)MHz and 160 MHz cases, the toolkit considers the least significant 8 bits.

The default value is 255 (1111 1111) which represents full preamble bandwidth.

Set Function: niWLANG_SetPreamblePuncturingMask
Get Function: niWLANG_GetPreamblePuncturingMask

NIWLANG_BSS_COLOR

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the identifier of the BSS (Basic Service Set) from which the 802.11ax PPDU is transmitted. The default value is 63. The valid values are 0 to 63, inclusive. Set Function: niWLANG_SetBSSColor Get Function: niWLANG_GetBSSColor

NIWLANG_NOT_SOUNDING_BIT

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the value of the Not Sounding field of the HT-SIG if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM as defined in section 20.3.9.4.3 of the IEEE Standard 802.11n-2009.

The default value is 1. Valid values are 0 or 1.

Set Function: niWLANG_SetNotSoundingBit
Get Function: niWLANG_GetNotSoundingBit

NIWLANG_GUARD_INTERVAL_TYPE

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the type of guard interval (cyclic prefix) in an OFDM symbol.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR is 0.8 microseconds, and the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT is 0.4 microseconds.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR is 8 microseconds, and the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT is 4 microseconds.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, and the bandwidth to 6 MHz or 7 MHz, NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR is 6 microseconds, and the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT is 3 microseconds.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AF_MIMOOFD

	M, and the bandwidth to 8 MHz, the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR is 4.5 microseconds, and the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT is 2.25 microseconds. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR is 3.2 microseconds, the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT is 1.6 microseconds, and the NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_SIXTEEN is 0.8 microseconds. The default value is NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR. Set Function: niWLANG_SetGuardIntervalType Get Function: niWLANG_SetGuardIntervalType
Values:	

`NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_FOUR (0)`	Specifies that guard interval length is equal to one-fourth of IDFT/DFT period.
`NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_EIGHT (1)`	Specifies that guard interval length is equal to one-eighth of IDFT/DFT period.
`NIWLANG_VAL_GUARD_INTERVAL_TYPE_ONE_BY_SIXTEEN (2)`	Specifies that guard interval length is equal to one-sixteenth of the IDFT/DFT period.

NIWLANG_MIDAMBLE_PERIODICITY

Data Type:	int32
Access:	read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the interval, in number of data symbols, after which the midambles are inserted in the data field of the 802.11ax signals. You must set this attribute to NIWLANG_VAL_MIDAMBLE_PERIODICITY_NONE when the value of the NIWLANG_NUMBER_OF_SPACE_TIME_STREAMS attribute is greater than 4.

The default value is NIWLANG_VAL_MIDAMBLE_PERIODICITY_NONE.

Set Function: niWLANG_SetOFDMMidamblePeriodicity
Get Function: niWLANG_GetOFDMMidamblePeriodicity

Values:

`NIWLANG_VAL_MIDAMBLE_PERIODICITY_NONE (0)`	Specifies that the midamble is not inserted.
`NIWLANG_VAL_MIDAMBLE_PERIODICITY_TEN_SYMBOLS (1)`	Specifies that the midamble is inserted after every 10 data symbols.
`NIWLANG_VAL_MIDAMBLE_PERIODICITY_TWENTY_SYMBOLS (2)`	Specifies that the midamble is inserted after every 20 data symbols.

NIWLANG_L_SIG_LENGTH

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the UL-LENGTH field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of UL-LENGTH is derived from the payload settings.

Set Function: niWLANG_SetOFDMLSIGLength
Get Function: niWLANG_GetOFDMLSIGLength

NIWLANG_PRE_FEC_PADDING_FACTOR

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the value of the pre-FEC padding factor sub-field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of the pre-FEC padding factor is derived from the payload settings.

Set Function:
niWLANG_SetOFDMPreFECPaddingFactor
Get Function:
niWLANG_GetOFDMPreFECPaddingFactor

NIWLANG_PE_DISAMBIGUITY

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:

Specifies the value of the PE disambiguity sub-field in the trigger frame that is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of the PE Disambiguity is derived from the payload settings.

Set Function:
niWLANG_SetOFDMPEDisambiguity
Get Function:
niWLANG_GetOFDMPEDisambiguity

NIWLANG_LDPC_EXTRA_SYMBOL_SEGMENT

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:

Specifies the value of the LDPC extra symbol segment field in the trigger frame which is used for 802.11ax Trigger-Based PPDU generation.

The default value is -1, which indicates that the value of the LDPC extra symbol segment field is derived from the payload settings.

Set Function:
niWLANG_SetOFDMLDPCExtraSymbolsUsed
Get Function:
niWLANG_GetOFDMLDPCExtraSymbolsUsed

NIWLANG_NOMINAL_PACKET_PADDING

Data Type: int32

Access: read/write

Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the nominal packet padding value used for determining the packet extension duration when you set the NIWLANG_STANDARD attribute to 80211AX MIMOOFDM. When you set the NIWLANG_PPDU_TYPE attribute to MU PPDU or Trigger-Based PPDU, this attribute value corresponds to the maximum nominal packet padding across all users. When you set the NIWLANG_PPDU_TYPE attribute to Trigger-Based PPDU and none of the NIWLANG_L_SIG_LENGTH attribute, NIWLANG_PRE_FEC_PADDING_FACTOR attribute, NIWLANG_PE_DISAMBIGUITY attribute, NIWLANG_LDPC_EXTRA_SYMBOL_SEGMENT attribute are set to -1, the toolkit ignores this attribute. The default value is Auto. Set Function: niWLANG_SetOFDMNominalPacketPadding Get Function: niWLANG_GetOFDMNominalPacketPadding
Values:	

`NIWLANG_VAL_NOMINAL_PACKET_PADDING_AUTO (-1)`	Specifies that the nominal packet padding is any of the following: 0us, if all the resource units are of size less than 242 and DCM is disabled. 0us, if all the resource units are of size less than 106 and DCM is enabled. 16us, otherwise.

NIWLANG_VAL_NOMINAL_PACKET_PADDING_0US (1)	Specifies that the nominal packet padding is 0us.
NIWLANG_VAL_NOMINAL_PACKET_PADDING_8US (2)	Specifies that the nominal packet padding is 8us.
NIWLANG_VAL_NOMINAL_PACKET_PADDING_16US (3)	Specifies that the nominal packet padding is 16us.

NIWLANG_PACKET_EXTENSION_DURATION

Data Type:	float64
Access:	read only
Functions:	`niWLANG_GetScalarAttributeF64`
Description:	Returns the duration of packet extension in the waveform when the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AX_MIMOOFDM. This value is expressed in seconds. Get Function: niWLANG_GetOFDMPacketExtensionDuration

NIWLANG_NUMBER_OF_SEGMENTS

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the number of frequency segments for 802.11ac or 802.11ax signals. For 80 MHz + 80 MHz transmission of 802.11ac or 802.11ax signals, set this attribute to 2 and the

NIWLANG_CHANNEL_BANDWIDTH attribute to 80 MHz.

The default value is 1.

Set Function: niWLANG_SetNumberOfSegments
Get Function: niWLANG_GetNumberOfSegments

NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the number of transmit channels for multiple input multiple output (MIMO) signals, as defined in section 20.3.3 of IEEE Standard 802.11n-2009.

If you set the NIWLANG_MAPPING_MATRIX_TYPE attribute to NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT, the number of transmit channels must be equal to the number of space-time streams. If you set the NIWLANG_MAPPING_MATRIX_TYPE attribute to values other than NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the NIWLANG_NUMBER_OF_EXTENSION_SPATIAL_STREAMS attribute.

The toolkit defines the number of space-time streams using the NIWLANG_MCS_INDEX attribute and NIWLANG_STBC_INDEX attribute, as specified in IEEE Standard 802.11n-2009.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, and the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS attribute to greater than one, the same waveform is generated on multiple RFSG devices inside the niWLANG_RFSGCreateAndDownloadMIMOWaveforms function.

The default value is 1. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the valid values are 1 to 4, inclusive. If you set the NIWLANG_STANDARD attribute to any other value, the valid values are 1 to 8, inclusive.

Set Function:
niWLANG_SetNumberOfTransmitChannels
Get Function:
niWLANG_GetNumberOfTransmitChannels

NIWLANG_NUMBER_OF_SPACE_TIME_STREAMS

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the number of space-time streams into which the data is divided.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU , use an empty string as the active channel string format to set this attribute.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU, use 'userx' as the active channel string syntax to set this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU or NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, use an empty string as the active channel string format to set this attribute.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, use 'userx' as the active channel string format to set this attribute.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, the active string syntax is an empty string.

The default value is 1.

The allowed values for this attribute are as follows. All values are inclusive.

If you set the NIWLANG_STANDARD attribute to

NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, the values are 1 to 4. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, the valid values are 1 to 8.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU, the valid values are 1 to 4.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, the valid values are 1 to 8.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, the valid values are 1 to 4.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, the valid values are 1 to 2.

Set Function:
niWLANG_SetNumberOfSpaceTimeStreams
Get Function:
niWLANG_GetNumberOfSpaceTimeStreams

NIWLANG_SPACE_TIME_STREAM_OFFSET

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the per user space time stream offset which is used for 802.11ax Trigger-Based PPDU generation.

The default value is 0.

You must use the following active channel string formats to set this property. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU, use 'userx' as the active channel string format to set this attribute.

Set Function:
niWLANG_SetSpaceTimeStreamOffset
Get Function:
niWLANG_GetSpaceTimeStreamOffset

NIWLANG_MU_MIMO_LTF_MODE_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies whether the HE-LTF sequence corresponding to each space time stream is masked by distinct orthogonal code, if the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

	The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetMUMIMOLTFModeEnabled Get Function: niWLANG_GetMUMIMOLTFModeEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_NUMBER_OF_USERS

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the number of users in a multi-user (MU) physical layer convergence procedure (PLCP) protocol data unit (PPDU). The default value is 1. You can set this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PAYLOAD_PPDU_TYPE_MU_PPDU, or when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PAYLOAD_PPDU_TYPE_MU_PPDU or NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

Get Function: niWLANG_GetNumberOfUsers
Set Function: niWLANG_SetNumberOfUsers

NIWLANG_MULTI_SEGMENT_GENERATION_MODE

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies whether to use a single generator or two generators for each channel of a multi-segment (80+80) MHz 802.11ac signal. This attribute is applicable when you set the NIWLANG_NUMBER_OF_SEGMENTS attribute to 2 and the channel bandwidth is 80 MHz. When you set this attribute to NIWLANG_VAL_SINGLE_GENERATOR, you have to specify the value of the NIWLANG_CARRIER_FREQUENCY attribute for both the segments.

The default value is NIWLANG_VAL_MULTIPLE_GENERATORS.

Set Function: niWLANG_SetMultiSegmentGenerationMode
Get Function: niWLANG_GetMultiSegmentGenerationMode

Values:

`NIWLANG_VAL_SINGLE_GENERATOR (0)`	Specifies that one vector signal generator is used to generate the (80+80) MHz 802.11ac signal.
`NIWLANG_VAL_MULTIPLE_GENERATORS (1)`	Specifies that two vector signal generators are used to generate the (80+80) MHz 802.11ac signal.

NIWLANG_SPATIAL_MAPPING_MODE

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether the spatial mapping is created from a single global matrix or per user. This attribute is applicable, only when you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU. The default value is NIWLANG_VAL_SPATIAL_MAPPING_MODE_COMMON. Set Function: niWLANG_SetSpatialMappingMode Get Function: niWLANG_GetSpatialMappingMode
Values:	

`NIWLANG_VAL_SPATIAL_MAPPING_MODE_COMMON (0)`	Specifies that a single global matrix is used for spatial mapping of all users.
`NIWLANG_VAL_SPATIAL_MAPPING_MODE_USER_SPECIFIC (1)`	Specifies that per user spatial mapping is done.

NIWLANG_MAPPING_MATRIX_TYPE

Data Type:	int32
Access:	read/write

Functions:

`niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description:

Specifies the mapping matrix type for mapping space-time streams to transmit channels as defined in section 20.3.11.10.1 of IEEE Standard 802.11n-2009.

Note: If you set the NIWLANG_MAPPING_MATRIX_TYPE attribute to values other than NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT, the number of transmit channels must be greater than or equal to the sum of the number of space-time streams and the NIWLANG_NUMBER_OF_EXTENSION_SPATIAL_STREAMS attribute if the standard is set to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM.

The default value is NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT. If the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute is set to NIWLANG_VAL_STANDARD_TRIGGER_BASED_PPDU then N-STS value is dependent on the NIWLANG_SPATIAL_MAPPING_MODE attribute. If you set the NIWLANG_SPATIAL_MAPPING_MODE attribute to NIWLANG_VAL_COMMON, the N STS value is the maximum number of space time streams across Resource Units (RUs). If you set the NIWLANG_SPATIAL_MAPPING_MODE attribute to NIWLANG_VAL_USER_SPECIFIC, the N STS value is the number of space time streams for the specified user. You must use the following active channel string formats when NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute is set to

NIWLANG_VAL_STANDARD_TRIGGER_BASED_PPDU. If you set the NIWLANG_SPATIAL_MAPPING_MODE attribute to NIWLANG_VAL_COMMON, the active channel string format is an empty string. If you set the NIWLANG_SPATIAL_MAPPING_MODE attribute to NIWLANG_VAL_USER_SPECIFIC, the active channel string format is 'userx'.

Set Function: niWLANG_SetMappingMatrixType
Get Function: niWLANG_GetMappingMatrixType

Values:

NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT (0)	Refer to the Mapping Matrix Type Values topic.
NIWLANG_VAL_MAPPING_MATRIX_TYPE_HADAMARD (1)	Refer to the Mapping Matrix Type Values topic.
NIWLANG_VAL_MAPPING_MATRIX_TYPE_FOURIER (2)	Refer to the Mapping Matrix Type Values topic.
NIWLANG_VAL_MAPPING_MATRIX_TYPE_USER_DEFINED (3)	Refer to the Mapping Matrix Type Values topic.

Mapping Matrix Type Values

The following table describes the values for the NIWLANG_MAPPING_MATRIX_TYPE attribute.

Value	Description
NIWLANG_VAL_MAPPING_MATRIX_TYPE_DIRECT (0)	Specifies direct mapping for space-time streams. The direct mapping matrix is given by $\mathbf{N_{Tx}}$ = $\mathbf{N_{STS}}$. Direct mapping matrix is derived by taking the subset ($\mathbf{N_{STS}}$ * $\mathbf{N_{STS}}$) of the 8 × 8 identity matrix. where $\mathbf{N_{Tx}}$ is the number of transmit channels s

	pecified by the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS attribute. **N_{STS}** is the number of space-time streams. If the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM**, the value of **N_{STS}** is determined by the MCS index and STBC index. The following matrix is an example of a 4 × 4 identity matrix. $\begin{bmatrix} 1 0 0 0 \\ 0 1 0 0 \\ 0 0 1 0 \\ 0 0 0 1 \end{bmatrix}$
NIWLANG_VAL_MAPPING_MATRIX_TYPE_HADAMARD (1)	Specifies Hadamard mapping for space-time streams and the extension spatial streams. The Hadamard mapping matrix is of size **N_{Tx}** * (**N_{STS}** + **N_{ESS}**), where **N_{Tx}** ≥ (**N_{STS}** + **N_{ESS}**) if the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM**. The Hadamard mapping matrix is of size **N_{Tx}** * (**N_{STS}**), where **N_{Tx}** ≥ N_{STS} if the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM**, **NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM** or **NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM**. The Hadamard mapping matrix is derived by taking the subset of the 8 × 8 Hadamard matrix. where **N_{Tx}** is the number of transmit channels specified by the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS attribute. **N_{STS}** is the number of space-time streams. I

	f the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM**, the value of $\mathbf{N_{STS}}$ is determined by the MCS index and STBC index $\mathbf{N_{ESS}}$ is the number of extension spatial streams The following matrix is an example of a 4 × 4 Hadamard matrix. $\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$
NIWLANG_VAL_MAPPING_MATRIX_TYPE_FOURIER (2)	Specifies Fourier mapping for space-time streams and the extension spatial streams. The Fourier mapping matrix is given by the following equation: $\{a_{nk}\} = \frac{1}{\sqrt{N}} e^{\left[\frac{j2\pi kn}{N}\right]}$ where **n**= 0 … (**N** - 1) **N** = $\mathbf{N_{Tx}}$, where $\mathbf{N_{Tx}}$ is the number of transmit channels specified by the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS attribute. **k** = 0…(($\mathbf{N_{STS}}$ + $\mathbf{N_{ESS}}$) -1) if the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM**, or **k** = 0…(**NSTS** - 1) if the NIWLANG_STANDARD attribute is set to **NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM**, **NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM** or **NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM**, $\mathbf{N_{STS}}$ is the number of space-time streams. If the

	`NIWLANG_STANDARD` attribute is set to **NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM**, the value of N_{ESS} is determined by the MCS index and STBC index. N_{ESS} is the number of extension spatial streams The following matrices are examples of 2 × 2, 3 × 3, and 4 × 4 matrices. $\frac{1}{\sqrt{2}}\begin{bmatrix}1+i0 & 1+i0\\1+i0 & 1+i0\end{bmatrix}$ $\frac{1}{\sqrt{3}}\begin{bmatrix}1+i0 & 1+i0 & 1+i0\\1+i0 & -0.5+i0.9 & -0.5-i0.9\\1+i0 & -0.5-i0.9 & -0.5+i0.9\end{bmatrix}$ $\frac{1}{2}\begin{bmatrix}1+i0 & 1+i0 & 1+i0 & 1+i0\\1+i0 & 0+i1 & -1+i0 & 0-i1\\1+i0 & -1+i0 & 1+i0 & -1+i0\\1+i0 & 0-i1 & -1+i0 & 0+i1\end{bmatrix}$
`NIWLANG_VAL_MAPPING_MATRIX_TYPE_USER_DEFINED (3)`	Specifies user-defined mapping for space-time streams and extension spatial steams. You can set this matrix using the `niWLANG_SetMappingMatrix` function.

NIWLANG_STBC_INDEX

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the difference between the number of space-time streams and the number of spatial streams, as defined in section 20.3.9.4.3 of IEEE Standard 802.11n-2009. The toolkit derives the number of spatial streams from the specified

	value of the NIWLANG_MCS_INDEX attribute. Different space-time coding schemes are defined in section 20.3.11.8.1 of IEEE Standard 802.11n-2009. The default value is 0. Valid Values are 0 to 2, inclusive. Set Function: niWLANG_SetSTBCIndex Get Function: niWLANG_GetSTBCIndex

NIWLANG_STBC_ALL_STREAMS_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether space-time block coding (STBC) is performed at the transmitter when the NIWLANG_STANDARD attribute is set to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM. Whenever STBC is performed, the number of space-time streams is equal to two times the number of spatial streams. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetSTBCAllStreamsEnabled Get Function: niWLANG_GetSTBCAllStreamsEnabled
Values:	

`NIWLANG_VAL_FALSE (0)`	Specifies that the STBC is not performed at the transmitter.

NIWLANG_VAL_TRUE (1)	Specifies that the STBC is performed at the transmitter.

NIWLANG_NUMBER_OF_EXTENSION_SPATIAL_STREAM

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the number of extension spatial streams (N_ESS) as defined in section 20.3.9.4.6 of IEEE Standard 802.11n-2009.

The value of N_ESS follows the definition shown in the following equation:
N_STS + N_ESS is less than or equal to N_Tx.

where N_STS is the number of space-time streams and is determined by the MCS index and STBC index. N_Tx is the number of transmit channels specified by the NIWLANG_NUMBER_OF_TRANSMIT_CHANNELS attribute.

Note: N_TX must be greater than or equal to N_STS.

The default value is 0. Valid values are 0 to 3, inclusive.

Set Function:
niWLANG_SetNumberOfExtensionSpatialStreams
Get Function:
niWLANG_GetNumberOfExtensionSpatialStreams

NIWLANG_CARRIER_FREQUENCY_OFFSET

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the offset in the center frequency of the complex baseband signal from the Carrier Frequency. This value is expressed in Hz. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string format is 'segmentx'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM the active channel string format is an empty string. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or

NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid value is -(1/2) * IQ Rate to (1/2) * IQ Rate, inclusive, where IQ Rate is the value of the NIWLANG_IQ_RATE attribute.

Set Function:
niWLANG_SetCarrierFrequencyOffset
Get Function:
niWLANG_GetCarrierFrequencyOffset

NIWLANG_SAMPLE_CLOCK_OFFSET

Data Type: float64

Access: read/write

Functions:
`niWLANG_SetScalarAttributeF64`
`niWLANG_GetScalarAttributeF64`

Description:
Specifies the offset in the Sample Clock frequency from the sample frequency defined by the following equation:

Sampling Frequency = max (maximum hardware I/Q Rate, oversampling factor * Nyquist Sampling Rate)

where
maximum hardware I/Q is the value of the NIWLANG_MAXIMUM_HARDWARE_IQ_RATE attribute
oversampling factor is the value of the NIWLANG_OVERSAMPLING_FACTOR attribute.

This value is expressed in parts per million (ppm).

For large offset values, with large waveform sizes, clock cycle slips may occur. Clock cycle slips can cause the final waveform size to be different from the expected size, given the ideal burst length and the interframe spacing.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string format is 'segmentx'

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM the active channel string format is an empty string.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty' if you set the

NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid values are -1000 to 1000, inclusive.

Set Function: niWLANG_SetSampleClockOffset
Get Function: niWLANG_GetSampleClockOffset

NIWLANG_ALL_IQ_IMPAIRMENTS_ENABLED

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies whether to apply I/Q impairments such as I DC offset, Q DC offset, quadrature skew, and I/Q gain imbalance to the waveform.

The default value is NIWLANG_VAL_TRUE.

Set Function: niWLANG_SetAllIQImpairmentsEnabled
Get Function: niWLANG_GetAllIQImpairmentsEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_IQ_GAIN_IMBALANCE

Data Type: float64

Access: read/write

Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the ratio of the mean amplitude of the in-phase (I) signal to the mean amplitude of the quadrature-phase (Q) signal. This value is expressed in dB. Note: If you set this attribute to a large value, there may be loss of dynamic range at the digital-to-analog converter (DAC). I/Q gain imbalance follows the definition shown in the following equation: I' = I - (gamma) * sin(phi) * Q + I0 Q' = (gamma) * cos(phi) * Q + Q0 where gamma = 10^(I/Q gain imbalance/20) phi is the quadrature skew I is the in-phase component of the waveform before applying impairments Q is the quadrature component of the waveform before applying impairments I' is the in-phase component of the waveform after applying impairments Q' is the quadrature component of the waveform before applying impairments I0 is the (RMS value of I) * (I DC Offset)/100 Q0 is the (RMS value of Q) * (Q DC Offset)/100 If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS or to NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM the active channel string syntax is an empty string. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFD

M or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid values are -6 to +6, inclusive.

Set Function: niWLANG_SetIQGainImbalance
Get Function: niWLANG_GetIQGainImbalance

NIWLANG_I_DC_OFFSET

Data Type:	float64
Access:	read/write

Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the value of the DC offset in the in-phase (I) channel as a percentage of the RMS magnitude of the unaltered I channel. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM the active channel string syntax is an empty string. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the

NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid values are -100 to +100, inclusive.

Set Function: niWLANG_SetIDCOffset
Get Function: niWLANG_GetIDCOffset

NIWLANG_Q_DC_OFFSET

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the value of the DC offset in the quadrature-phase (Q) channel as percentage of the RMS magnitude of the unaltered Q channel. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM the active channel string syntax is an empty string. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or

NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid values are -100 to +100, inclusive.

Set Function: niWLANG_QDCOffset
Get Function: niWLANG_QDCOffset

NIWLANG_QUADRATURE_SKEW

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the deviation in angle from 90 degrees between the in-phase (I) and quadrature-phase (Q) signals. This value is expressed in degrees. Refer to the Quadrature Skew help topic for more information about quadrature skew.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM the active channel string syntax is an empty string.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

The default value is 0. Valid values are -30 to +30, inclusive.

Set Function: niWLANG_SetQuadratureSkew
Get Function: niWLANG_GetQuadratureSkew

Quadrature Skew

Quadrature skew follows the definition shown in the following equation:

$$I' = I - y * \sin(\Psi) * Q + I_0$$

$$Q' = y * \cos(\Psi) * Q + Q_0$$

where

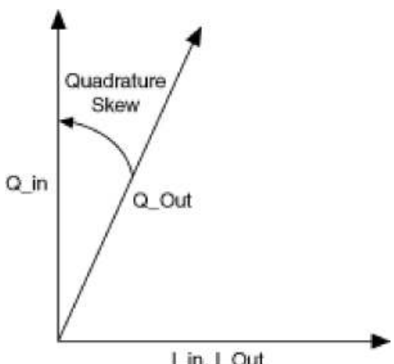
$$y = 10^{\left(\frac{IQ\ gain\ imbalance}{20}\right)}$$

$$\Psi = quadrature\ skew$$

$$I_0 = (I_{\mathrm{RMS}} * I_{\mathrm{DC\ offset}})/100$$

$$Q_0 = (Q_{\mathrm{RMS}} * Q_{\mathrm{DC\ offset}})/100$$

The following figure is a graphical representation of quadrature skew.



NIWLANG_TIMING_SKEW

Data Type: float64

Access: read/write

Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the difference between the sampling instants of I and Q streams. This value is expressed in seconds. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM the active channel string syntax is an empty string. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the

NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

Refer to the Configuring Active Channels (LabWindows/CVI) help topic for more information about configuring active channels.

The default value is 0. Valid values are -1 microsecond to 1 microsecond, inclusive.

Set Function: niWLANG_SetTimingSkew
Get Function: niWLANG_GetTimingSkew

NIWLANG_TIME_DELAY

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the time delay for each user within an 802.11ax Trigger-Based signal. This value is expressed in seconds. You must set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM and the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU. Use this attribute to introduce relative time delays between multiple users within an 802.11ax Trigger-Based signal. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, use 'userx' as the active channel string format, if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU.

The default value is 0.

Set Function: niWLANG_SetTimeDelay
Get Function: niWLANG_GetTimeDelay

NIWLANG_AWGN_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to introduce additive white Gaussian noise (AWGN) to the baseband waveform. The toolkit uses the value specified in the NIWLANG_CARRIER_TO_NOISE_RATIO attribute to add the AWGN. The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetAWGNEnabled Get Function: niWLANG_GetAWGNEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_CARRIER_TO_NOISE_RATIO

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the carrier-to-noise ratio (CNR)of the waveform generated. This value is expressed in dB. Noise bandwidth is equal to the value of the NIWLANG_IQ_RATE attribute. The toolkit

ignores the NIWLANG_CARRIER_TO_NOISE_RATIO attribute if you set the NIWLANG_AWGN_ENABLED attribute to NIWLANG_VAL_FALSE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211BG_DSSS, or NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, the active channel string syntax is an empty string.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, the active channel string syntax is 'channelx'.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM or NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, the active channel string syntax is 'segmentx/channely'.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is 'segmentx/channely' if you set the NIWLANG_PPDU_TYPE attribute to NIWLANG_VAL_PPDU_TYPE_SU_PPDU, NIWLANG_VAL_PPDU_TYPE_EXTENDED_RANGE_SU_PPDU, or NIWLANG_VAL_PPDU_TYPE_MU_PPDU.
If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the active channel string format is '[userx/]segmenty/channelz' if you set the NIWLANG_PPDU_TYPE attribute to

NIWLANG_VAL_PPDU_TYPE_TRIGGER_BASED_PPDU (userx is optional if you want to apply to all the users).

Refer to the Configuring Active Channels (LabWindows/CVI) help topic for more information about configuring active channels.

The default value is 50. Valid values are -100 to 100, inclusive.

Set Function: niWLANG_SetCarrierToNoiseRatio
Get Function: niWLANG_GetCarrierToNoiseRatio

NIWLANG_PULSE_SHAPING_FILTER_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to apply pulse-shaping filter to the generated signal. Note: The toolkit ignores this attribute and enables the pulse shaping filter, if you set the NIWLANG_COMPATIBILITY_VERSION attribute to NIWLANG_VAL_COMPATIBILITY_VERSION_010000. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS, the default value is NIWLANG_VAL_TRUE. If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFD

M,
NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM,
NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM,
NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM,
NIWLANG_VAL_STANDARD_80211AF_MIMO_OFDM, or
NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM, the default value is NIWLANG_VAL_FALSE.

Set Function:
niWLANG_SetPulseShapingFilterEnabled
Get Function:
niWLANG_GetPulseShapingFilterEnabled

Values:

`NIWLANG_VAL_FALSE (0)`	Disables the attribute.
`NIWLANG_VAL_TRUE (1)`	Enables the attribute.

NIWLANG_PULSE_SHAPING_FILTER_TYPE

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the pulse-shaping filter type to use to ensure that the signal spectrum meets the spectral mask criteria as defined in section 17.3.9.2 of IEEE Standard 802.11a-1999, section 18.4.7.3 of IEEE Standard 802.11b-1999, and section 20.3.21.1 of IEEE Standard 802.11n-2009.

The default value is NIWLANG_VAL_FILTER_RECTANGULAR if you set the NIWLANG_STANDARD attribute to

NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMO_OFDM.

The default value is NIWLANG_VAL_FILTER_ROOT_RAISED_COSINE if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS.

The default value is NIWLANG_VAL_FILTER_RAISED_COSINE if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211P_DSSS.

Set Function:
niWLANG_SetPulseShapingFilterType
Get Function:
niWLANG_GetPulseShapingFilterType

Values:

`NIWLANG_VAL_FILTER_RECTANGULAR (0)`	Refer to the Pulse Shaping Filter Type Values help topic in the NI WLAN Generation Toolkit Help.
`NIWLANG_VAL_FILTER_RAISED_COSINE (1)`	Refer to the Pulse Shaping Filter Type Values help topic in the

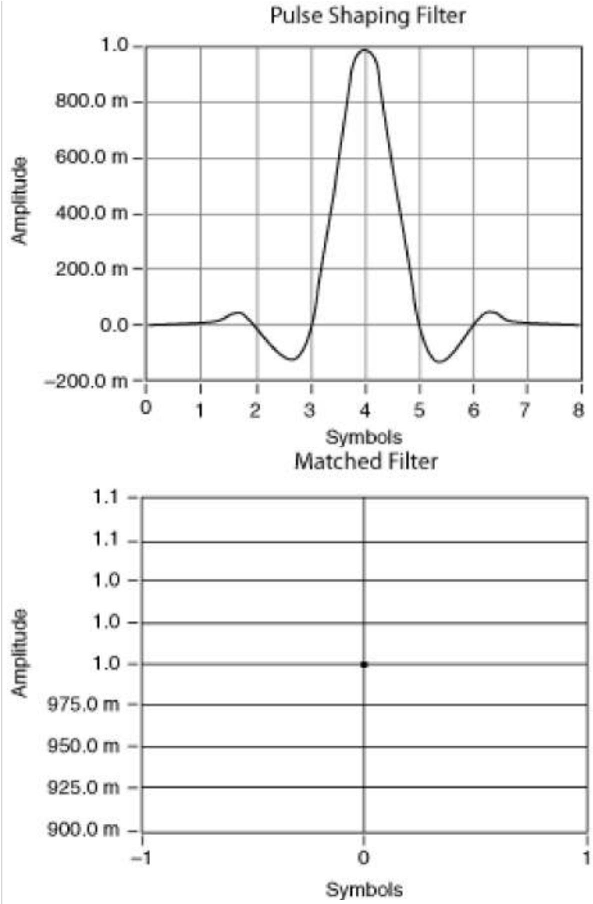
	NI WLAN Generation Toolkit Help.
NIWLANG_VAL_FILTER_ROOT_RAISED_COSINE (2)	Refer to the Pulse Shaping Filter Type Values help topic in the NI WLAN Generation Toolkit Help.
NIWLANG_VAL_FILTER_GAUSSIAN (3)	Refer to the Pulse Shaping Filter Type Values help topic in the NI WLAN Generation Toolkit Help.

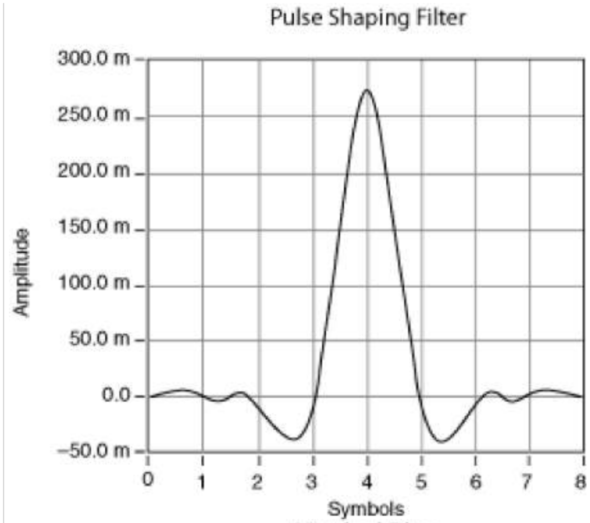
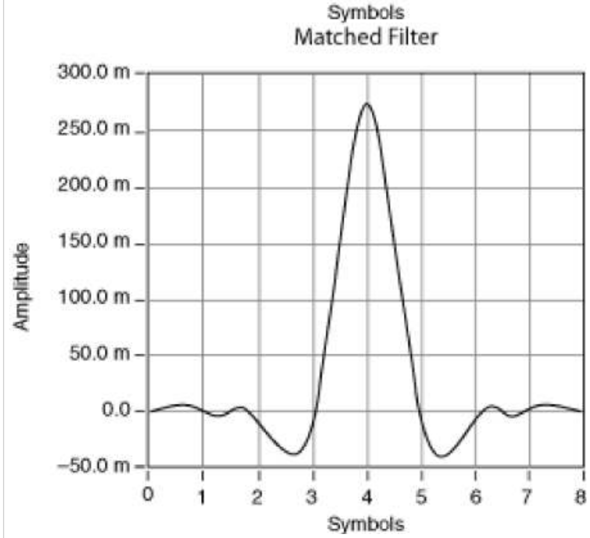
Pulse Shaping Filter Type Values

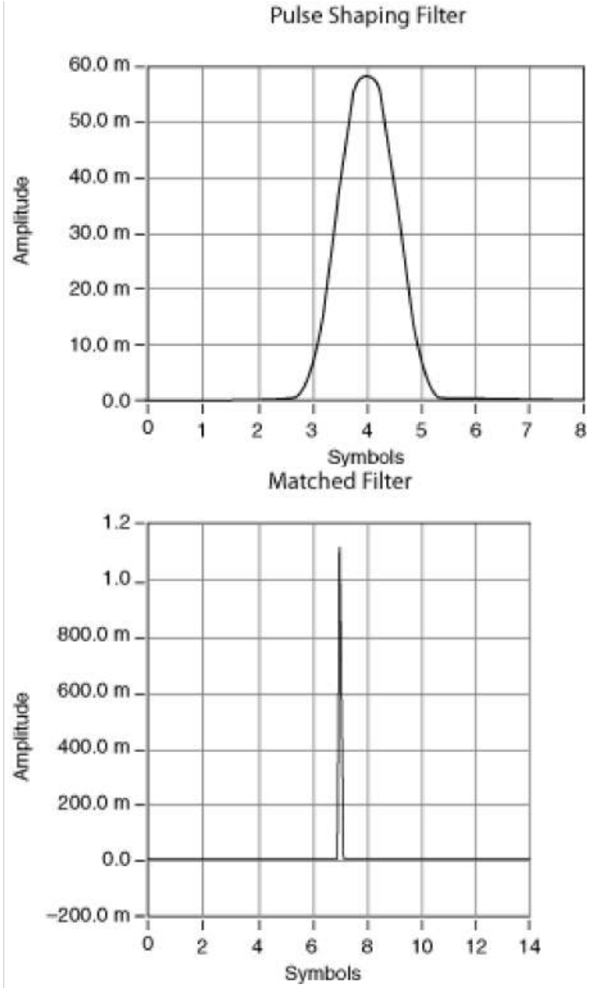
The following table describes the values for the NIWLANG_PULSE_SHAPING_FILTER_TYPE attribute.

The default value is NIWLANG_VAL_FILTER_ROOT_RAISED_COSINE if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_80211BG_DSSS. For other values of the NIWLANG_STANDARD attribute the default value is NIWLANG_VAL_FILTER_RECTANGULAR.

Value	Description
NIWLANG_VAL_FILTER_RECTANGULAR (0)	Generates a rectangular filter that is always one symbol long.
NIWLANG_VAL_FILTER_RAISED_COSINE (1)	Generates a filter with a frequency-domain response in the transition band that has the shape of a raised cosine. The filter is defined by a roll-off factor configured using the NIWLANG_PULSE_SHAPING_FILTER_PARAMETER attribute.

	
NIWLANG_VAL_FILTER_ROOT_RAISED_COSINE (2)	Generates a filter with a frequency-domain response in the transition band that has the shape of a square root of a raised cosine. The filter is defined by a roll-off factor configured using the NIWLANG_PULSE_SHAPING_FILTER_PARAMETER attribute.

	 
`NIWLANG_VAL_FILTER_GAUSSIAN (3)`	Generates a filter with a frequency-domain response and time-domain responses that are Gaussian. The filter is defined by the product of 3 dB bandwidth and symbol time, configured using the `NIWLANG_PULSE_SHAPING_FILTER_PARAMETER` attribute.



NIWLANG_PULSE_SHAPING_FILTER_PARAMETER

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the value of the rolloff factor (alpha) if you set the NIWLANG_PULSE_SHAPING_FILTER_TYPE attribute to NIWLANG_VAL_FILTER_RAISED_COSINE or NIWLANG_VAL_FILTER_ROOT_RAISED_COSINE. If you set the

NIWLANG_PULSE_SHAPING_FILTER_TYPE attribute to NIWLANG_VAL_FILTER_GAUSSIAN, you can calculate the NIWLANG_PULSE_SHAPING_FILTER_PARAMETER attribute by multiplying B and T, where B is the 3 dB bandwidth and T is the symbol period for a Gaussian filter.

If you set the NIWLANG_PULSE_SHAPING_FILTER_TYPE attribute to NIWLANG_VAL_FILTER_RECTANGULAR, the toolkit ignores this attribute.

The default value is 0.5. Valid values are 0.1 to 0.95, inclusive.

Set Function: niWLANG_SetPulseShapingFilterParameter
Get Function: niWLANG_GetPulseShapingFilterParameter

NIWLANG_PULSE_SHAPING_FILTER_LENGTH

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`

Description: Specifies the length of the pulse-shaping filter. This value is expressed in symbols. The length affects the frequency response of the filter. The toolkit ignores this attribute when the NIWLANG_PULSE_SHAPING_FILTER_LENGTH attribute is set to NIWLANG_VAL_FALSE.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211P_OFDM, the default value is 100. In all other instances, The default value is 8.

Set Function:
niWLANG_SetPulseShapingFilterLength
Get Function:
niWLANG_GetPulseShapingFilterLength

NIWLANG_WINDOWING_METHOD

Data Type: int32

Access: read/write

Functions: `niWLANG_SetScalarAttributeI32`
`niWLANG_GetScalarAttributeI32`

Description: Specifies the method of applying window to the baseband signal, if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211AG_OFDM, NIWLANG_VAL_STANDARD_80211J_OFDM, NIWLANG_VAL_STANDARD_80211P_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211G_DSSS_OFDM, NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AH_MIMO_OFDM, NIWLANG_VAL_STANDARD_80211AF_MIMOOFDM, or NIWLANG_VAL_STANDARD_80211AX_MIMOOFDM. This attribute is ignored if you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211BG_DSSS.

Refer to the Windowing topic for more information about windowing for OFDM signals.

The default value is

	NIWLANG_VAL_WIN_METHOD_CENTERED_AT_SYMBOL_BOUNDARY. Set Function: niWLANG_SetWindowingMethod Get Function: niWLANG_GetWindowingMethod
Values:	

`NIWLANG_VAL_WIN_METHOD_CENTERED_AT_SYMBOL_BOUNDARY (0)`	Specifies that the window is applied with its center at the boundary between two OFDM symbols.
`NIWLANG_VAL_WIN_METHOD_STARTING_AT_SYMBOL_BOUNDARY (1)`	Specifies that the window is applied with its starting position at the boundary between two OFDM symbols.

NIWLANG_DSSS_WINDOW_LENGTH

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the window length for direct spread spectrum signals. This value is expressed in seconds. If you do not want windowing, set this attribute to 0. This attribute provides power ramp-up and ramp-down for the entire burst. Refer to the Windowing help topic for more information about windowing for DSSS signals. The default value is 2 microseconds. Set Function: niWLANG_SetDSSSWindowLength Get Function: niWLANG_GetDSSSWindowLength

NIWLANG_OFDM_WINDOW_LENGTH

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies the window length for OFDM signals at the sampling rate equal to the channel bandwidth. This value is expressed in samples. For example, if the window length is 2, the channel bandwidth is 20 MHz and the oversampling factor is 4, then the samples over which windowing is applied is 8. This attribute provides a smooth, spurious free transition from the end of one OFDM symbol to the cyclic prefix of the next symbol. If you do not want windowing, set this attribute to 0. Refer to the Windowing help topic for more information about windowing for OFDM signals. The default value is 2. Set Function: niWLANG_SetOFDMWindowLength Get Function: niWLANG_GetOFDMWindowLength

NIWLANG_SWAP_I_AND_Q_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to swap the data in the I and Q streams.

	The default value is NIWLANG_VAL_FALSE. Set Function: niWLANG_SetSwapIAndQEnabled Get Function: niWLANG_GetSwapIAndQEnabled
Values:	<table><tr><td>`NIWLANG_VAL_FALSE (0)`</td><td>Disables the attribute.</td></tr><tr><td>`NIWLANG_VAL_TRUE (1)`</td><td>Enables the attribute.</td></tr></table>

NIWLANG_COMPATIBILITY_VERSION

Data Type:	int32
Access:	read only
Functions:	`niWLANG_GetScalarAttributeI32`
Description:	Indicates the compatibilityVersion parameter of the niWLANG_OpenSession function. Get Function: niWLANG_GetToolkitCompatibilityVersion
Values:	<table><tr><td>`NIWLANG_VAL_COMPATIBILITY_VERSION_010000 (10000)`</td><td>Indicates that the compatibilityVersion parameter of the niWLANG_OpenSession function is set to NIWLANG_VAL_COMPATIBILITY_VERSION_010000.</td></tr><tr><td>`NIWLANG_VAL_COMPATIBILITY_VERSION_020000 (20000)`</td><td>Indicates that the compatibilityVersion parameter of the niWLANG_OpenSession function is set to NIWLANG_VAL_COMPATIBILITY_VERSION_020000.</td></tr></table>

NIWLANG_VAL_COMPATIBILITY_VERSION_030000 (30000)	Indicates that the compatibilityVersion parameter of the niWLANG_OpenSession function is set to NIWLANG_VAL_COMPATIBILITY_VERSION_030000.
NIWLANG_VAL_COMPATIBILITY_VERSION_040000 (40000)	Indicates that the compatibilityVersion parameter of the niWLANG_OpenSession function is set to NIWLANG_VAL_COMPATIBILITY_VERSION_040000.
NIWLANG_VAL_COMPATIBILITY_VERSION_050000 (50000)	Indicates that the compatibilityVersion parameter of the niWLANG_OpenSession function is set to NIWLANG_VAL_COMPATIBILITY_VERSION_050000.

NIWLANG_IQ_WAVEFORM_SIZE

Data Type:	int32
Access:	read only
Functions:	`niWLANG_GetScalarAttributeI32`
Description:	Returns the size of the generated I/Q waveform. This value is expressed in samples. Get Function: niWLANG_GetIQWaveformSize

NIWLANG_SAMPLE_CLOCK_RATE_FACTOR

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the factor by which the Sample Clock rate is multiplied to generate a signal that is compressed in the frequency domain and expanded in the time domain. For example, a 40 MHz 802.11n signal can be compressed in the frequency domain to 20 MHz, if the Sample Clock rate is reduced to half. In this case, you must set this attribute to 0.5 to generate the signal. The default value is 1. Valid values are 0.001 to 1, inclusive. Set Function: niWLANG_SetSampleClockRateFactor Get Function: niWLANG_GetSampleClockRateFactor

NIWLANG_OFDM_LEGACY_SCALING_ENABLED

Data Type:	int32
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeI32` `niWLANG_GetScalarAttributeI32`
Description:	Specifies whether to enable standard-defined transmit chain scaling of the legacy part of the high throughput (HT) and very high throughput (VHT) frames. Note: Configure this attribute only when you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM or

	NIWLANG_VAL_STANDARD_80211AC_MIMO_OFDM. Note: The toolkit ignores the NIWLANG_OFDM_LEGACY_SCALING_ENABLED attribute for all other standards. Note: This attribute is available only when you set the toolkit compatibility version parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_020000 or NIWLANG_VAL_COMPATIBILITY_VERSION_030000. The default value is NIWLANG_VAL_TRUE. Set Function: niWLANG_SetOFDMLegacyScalingEnabled Get Function: niWLANG_GetOFDMLegacyScalingEnabled
Values:	`NIWLANG_VAL_FALSE (0)` Disables the attribute. `NIWLANG_VAL_TRUE (1)` Enables the attribute.

NIWLANG_POWER_LEVEL

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the average power level of the active portion of the burst for signal generation. This value is expressed in dBm. The active portion of the burst is the WLAN packet excluding the interframe spacing.

Note: This attribute is available only if you set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_010000 or NIWLANG_VAL_COMPATIBILITY_VERSION_020000.

If you set the NIWLANG_STANDARD attribute to NIWLANG_VAL_STANDARD_80211N_MIMO_OFDM, use an active channel string to configure this attribute. Refer to the Configuring Active Channels (LabWindows/CVI) help topic for more information about configuring active channels.

The default value is -10.

Set Function: niWLANG_SetPowerLevel

NIWLANG_WAVEFORM_SCALING_FACTOR

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the scaling factor for the waveform, as a percentage of the maximum sample magnitude, to reduce the overshoot associated with the digital-to-analog converter (DAC) interpolation filter and other finite impulse response (FIR) filters in the NI RF vector signal generators. Note: This attribute is available only if you set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_010000.

The default value is 99. Valid values are 1 to 100, inclusive. You can reduce the value to avoid clipping, but you may lose dynamic range.

NIWLANG_PEAK_TO_AVERAGE_POWER_RATIO

Data Type:	float64
Access:	read only
Functions:	`niWLANG_GetScalarAttributeF64`
Description:	Returns the peak-to-average power ratio (PAPR)of the output complex waveform. This value is expressed in dB. Note: This attribute is available only if you set the compatibilityVersion parameter of the niWLANG_OpenSession function to NIWLANG_VAL_COMPATIBILITY_VERSION_010000.

NIWLANG_GUARD_INTERVAL

Data Type:	float64
Access:	read/write
Functions:	`niWLANG_SetScalarAttributeF64` `niWLANG_GetScalarAttributeF64`
Description:	Specifies the length of the cyclic prefix (CP)of an OFDM symbol as specified in section 20.1.1 of IEEE Standard 802.11n-2009 and section 22.3.6 of IEEE Standard 802.11ac-2013. This value is expressed in nanoseconds. The guard interval can be 800 nanoseconds (long) or 400 nanoseconds (short). Note: The toolkit does not support short guard

interval if the modulation and coding scheme (MCS) index is 0-7 and the physical layer convergence protocol (PLCP) frame format is Greenfield format.

Note: This attribute is available only when you set the toolkit compatibilityVersion parameter of the niWLANG_OpenSession function to 2.0.0 or 3.0.0.

The default value is 800 ns. Valid values are 400 nanoseconds and 800 nanoseconds.

Set Function: niWLANG_SetOFDMGuardInterval
Get Function: niWLANG_GetOFDMGuardInterval

WLAN Calibration Utility

This help file contains information about using the WLAN Calibration Utility to perform self-calibration, local oscillator alignment and noise floor measurement on NI vector signal transceivers (VSTs), PXIe-5646 and 5840, during WLAN measurements.

Getting Started

To launch the WLAN Calibration Utility (WCU), go to **Start»All Programs»National Instruments»WLAN Calibration Utility**.

NI recommends closing the WLAN Generation Soft Front Panel and the WLAN Analysis Soft Front Panel while running WLAN Calibration Utility.

Calibration of NI VST Devices

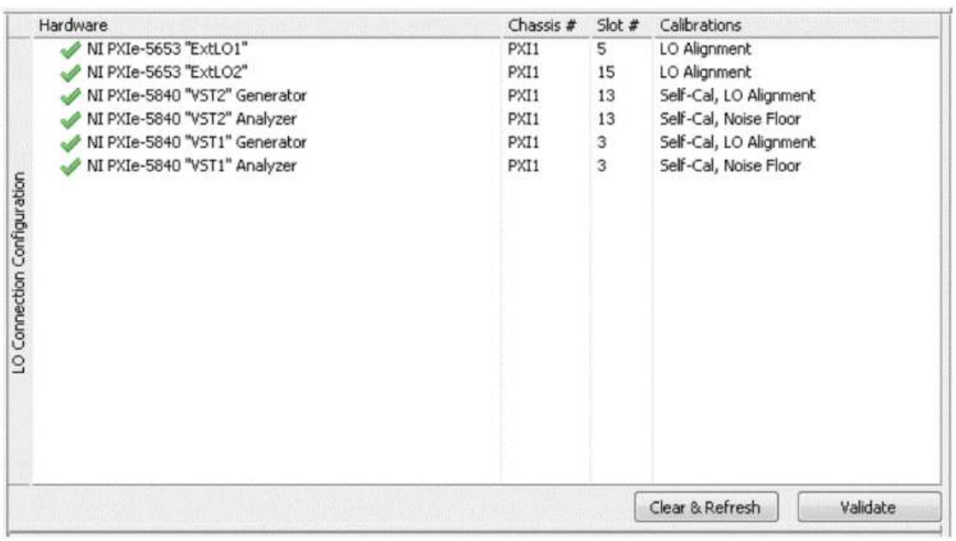
Refer to the Calibration of NI Vector Signal Transceiver Devices for WLAN Measurements topic for information about,

- Kinds of calibrations required to perform WLAN measurements and why they are important.
- How to setup hardware for different calibrations.

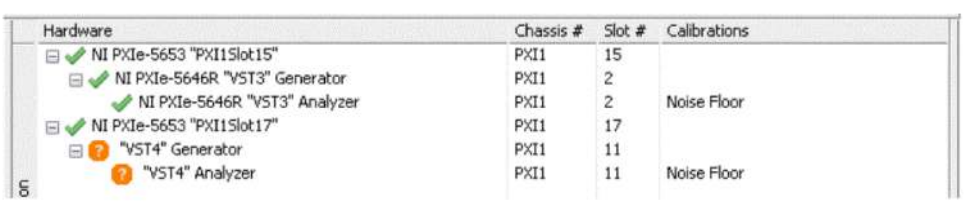
- The order in which the calibrations need to be performed and the minimum requirements.

LO Connection Configuration

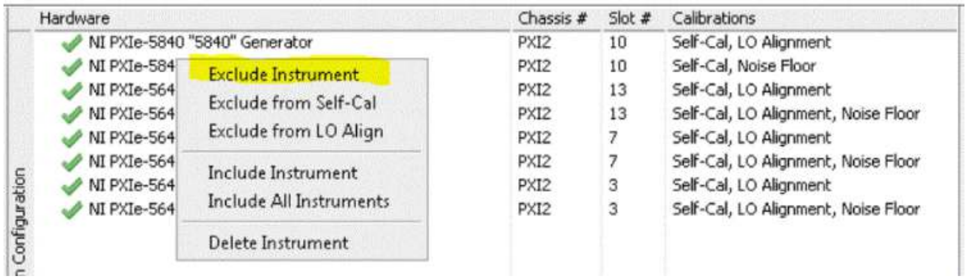
The LO Connection Configuration window specifies the LO connection configurations for the hardware listed in the tree. Click **Clear & Refresh** to view/ refresh the LO Connection Configuration tree that displays the hardware specific to WLAN measurements that are available in the PXI chassis, such as NI Vector Signal Generators, NI Vector Signal Analyzers, and the NI External LO Devices (PXIe-5653).



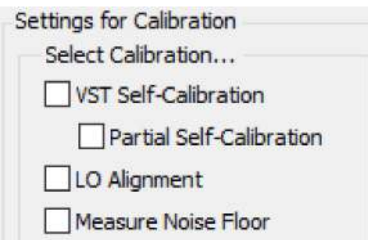
You can drag and drop the selected hardware to define the LO Connection Topology. For more illustrations, refer to the **Specifying Connections in a LO Connection Configuration Widow** section in the Configuring LO Connection in WLAN Analysis Soft Front Panel topic. Devices that are validated against NI-MAX show a green check mark, while the devices not validated against NI-MAX during the validation procedure show an orange question mark.



Click **Validate** to validate the hardware shown in the LO Connection Configuration tree against the hardware listed in NI-MAX. For more information about validating hardware refer to the Validate Hardware section. You can identify a hardware module in your physical setup in the **Chassis#** and **Slot#** columns. Right-click to include or exclude calibrations for the instrument. You can delete an instrument from the tree as well.



Selecting Calibrations



The different types of calibrations are,

- ### VST Self Calibration

  Performs standard device self-calibration.

- ### LO Alignment

  Performs mixer level calibration with the an external LO source.

- ### Measure Noise Floor

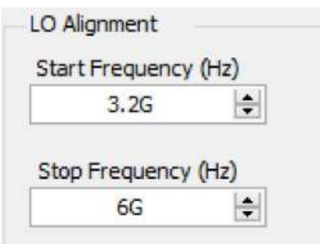
  Measures the noise floor of the instrument.

Refer to the Calibration of NI Vector Signal Transceiver Devices for WLAN Measurements topic for more information on WLAN calibration.

LO Alignment

LO Alignment configures the frequency range to align to the external LO source or device(PXIe-5653), and is applicable only for the devices that take in an external LO signal.

Configure the start and stop frequencies of the range to be considered for LO Alignment as shown in the following image. The default **Start Frequency** is 3.2 GHz, which is the lowest possible frequency supported in WCU. The default **Stop Frequency** is 6 GHz, which is the maximum frequency supported in WCU.



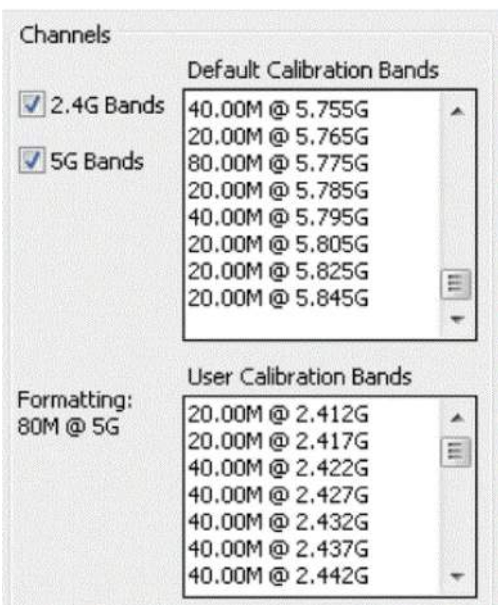
Noise Floor Measurement

Use this section to configure the minimum and maximum reference levels in order to perform the Noise Floor Measurement for all reference levels with the specified step size. These reference level values correspond to the signal received at the RF IN port.

- Minimum reference level for noise floor measurement must be greater than or equal to -50 dBm.
- Currently, the only value allowed for the reference level step size is 0.5 dB.

Channels for Noise Floor Measurement

Use this section to configure channels for which you want to perform noise floor measurement. A channel is the combination of channel bandwidth and center frequency as shown in the following image, where 40.00M is the channel bandwidth in MHz and 5.755G is the center frequency in GHz.



The **Default Calibration Bands** list displays the default list of channels in each band that you select. You can select **2.4G Bands** to specify channels in the 2.4 GHz band and **5G Bands** to specify channels in the 5G Bands.

Use the **User Calibration Bands** control to specify any additional channels using the same string format as in the **Default Calibration Bands**.

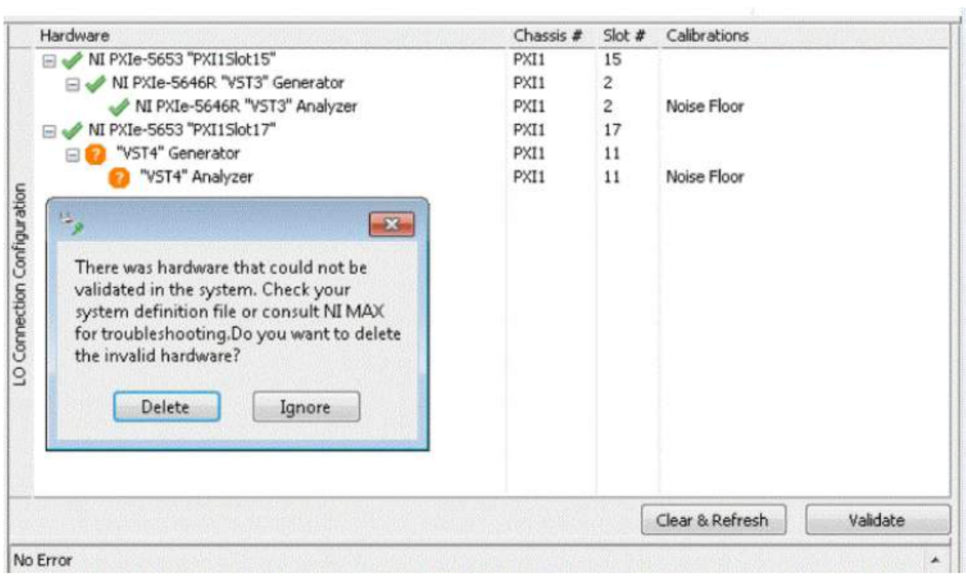
Validating Hardware

You can validate the current hardware listed in the LO Connection Configuration Tree against the hardware listed in NI-MAX to

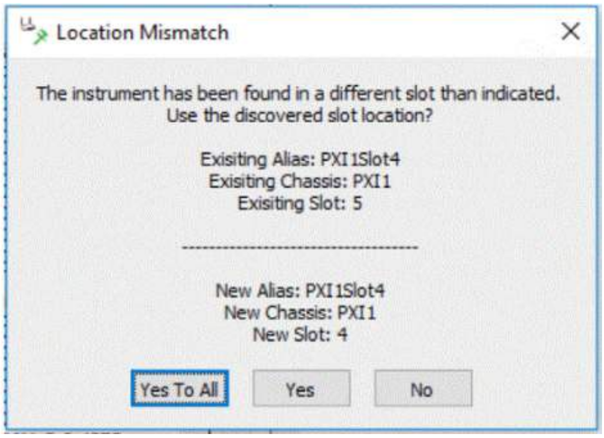
- Load the default configuration on launch
- Load the WCU configuration
- Click **Validate**
- Click **Run**

During validation, the hardware that is listed in the LO connection Configuration Tree that is not available in NI-MAX will show an orange question mark. A pop-up window with an option of either ignoring or deleting the invalid hardware is provided as show in the the following image.

Click Delete to remove the invalid hardware from LO connection configuration.

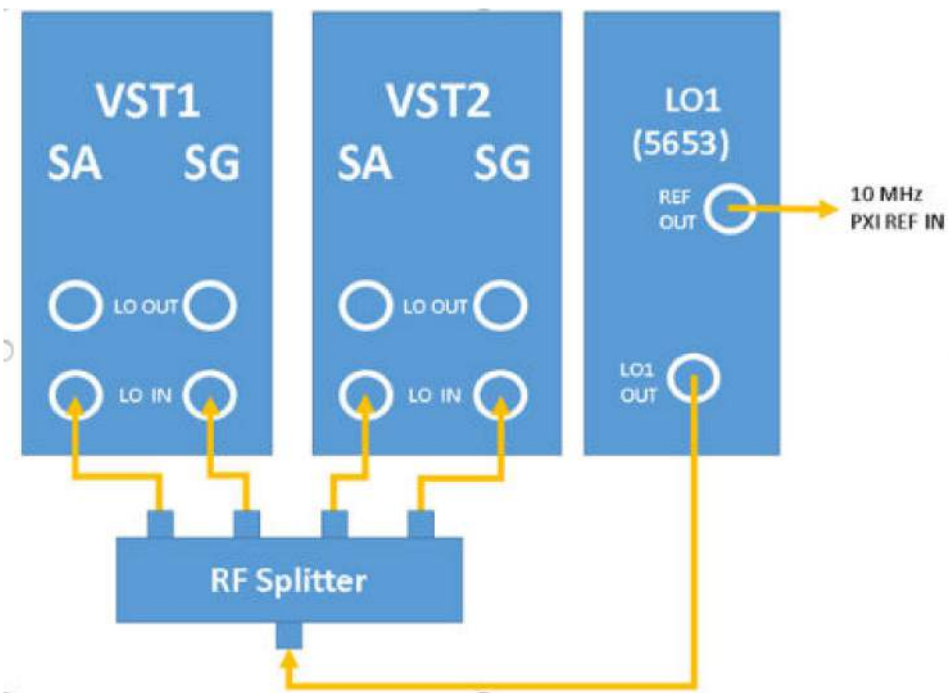


The validation procedure will also validate the location of your hardware in the system. If the location is changed, you will get a pop-up message to update the chassis and slot location.

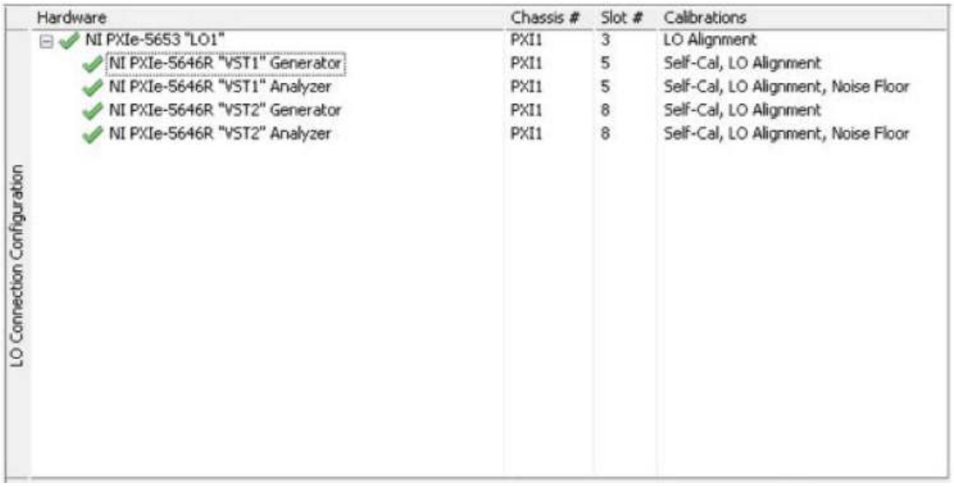


Specifying Connections in LO Connection Configuration Window

The following image illustrates a 2x2 MIMO system with external LO shared between SAs and SGs.



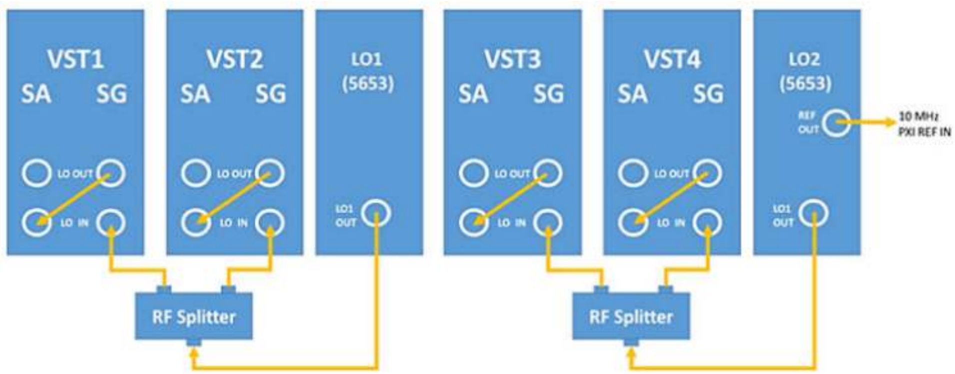
The WCU Connection Configuration Window for a 2x2 MIMO System with External LO shared between SAs and SGs is as shown in the following image.



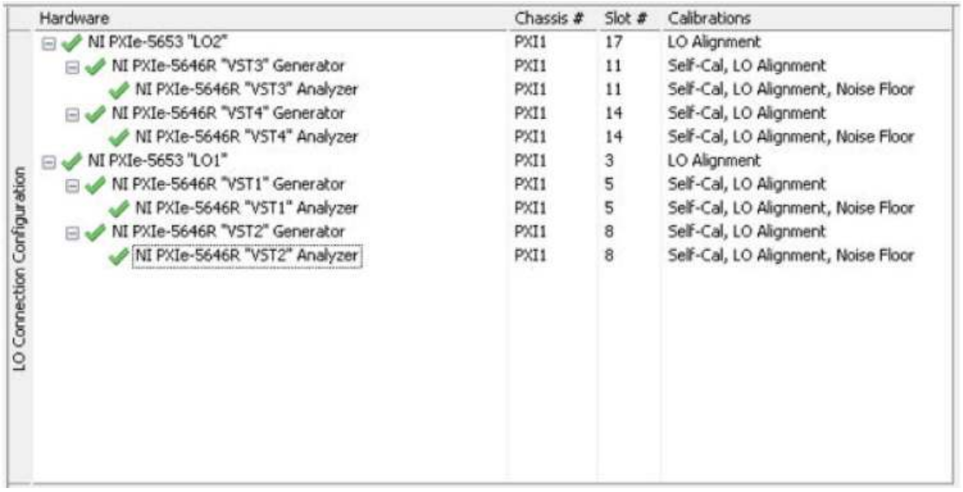
LO Connection Configuration

Hardware	Chassis #	Slot #	Calibrations
NI PXIe-5653 "LO1"	PXI1	3	LO Alignment
NI PXIe-5646R "VST1" Generator	PXI1	5	Self-Cal, LO Alignment
NI PXIe-5646R "VST1" Analyzer	PXI1	5	Self-Cal, LO Alignment, Noise Floor
NI PXIe-5646R "VST2" Generator	PXI1	8	Self-Cal, LO Alignment
NI PXIe-5646R "VST2" Analyzer	PXI1	8	Self-Cal, LO Alignment, Noise Floor

The following image illustrates a 4x4 MIMO/Multi-segment system with two external LOs shared between SGs. The SG LOs are exported to SAs.



The WCU Connection Configuration Window for a 4x4 MIMO/Multi Segment System with two External LOs shared between SGs is shown in the following image. The SG LOs are exported to SAs.

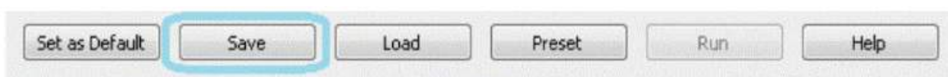


Set Default Configuration

Set as Default sets the current configuration as the default configuration. This default configuration will be used on subsequent launches of the WLAN Calibration Utility. This configuration will also be used to automatically load the LO connection topology in the WLAN Generation Soft Front Panel and the WLAN Analysis Soft Front Panel.

Save a WCU Configuration

Click **Save** to save the configuration into an XML file.

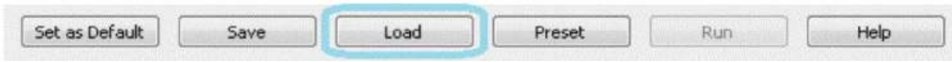


Load WCU Configuration

Use **Load** to load previously saved XML files in the WLAN Calibration Utility.

Note You will find example XML documents at this location, C:\Program Files\National Instruments\WLAN Toolkit\Calibration Utility\Examples directory.

Once you load the configuration file, this utility will automatically run a validation procedure. For more information, refer to **Validate Hardware** section.

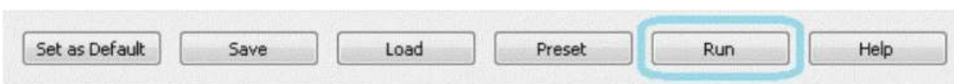


Preset

Preset returns WCU to the default state.

Run

Run starts the calibration procedure. If the hardware listed in the connection configuration are not validated yet, clicking **Run** will first validate the listed hardware with NI-MAX. For more information about validation, refer to the **Validate Hardware** section. Calibration procedure will start only after hardware has been successfully validated.



Help

Help opens up a dialog/pop-up window that displays basic information about WCU.