IVI
Interchangeable
Virtual
Instruments

IVI-4.14: IviUpconverter Class Specification

October 14, 2016 Edition
Revision 2.0

Important Information

The IVI-4.14: IviUpconverter Class Specification is authored by the IVI Foundation member companies. For a vendor membership roster list, please visit the IVI Foundation web site at www.ivifoundation.org.

The IVI Foundation wants to receive your comments on this specification. You can contact the Foundation through the web site at www.ivifoundation.org.

Warranty

The IVI Foundation and its member companies make no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The IVI Foundation and its member companies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Trademarks

Product and company names listed are trademarks or trade names of their respective companies.

No investigation has been made of common-law trademark rights in any work.

Table of Contents

IviUpconverter Class Specification

IviUpconverter Revision History

This section is an overview of the revision history of the IviUpconverter specification.

Table 1-1-1 IviUpconverter Class Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 1.0	October 21, 2009	Version 1.0 of Specification Approved
Revision 2.0	June 9, 2010	Incorporated IVI.NET.
Revision 2.0	August 25, 2011	Editorial IVI.NET change. Change references to process-wide locking to AppDomain-wide locking. Add an overload to the Create factory method that takes locking related parameters.
Revision 2.0	September 24, 2015	Editorial Change – Clarified the use of one-based index for C, and zero-based index for .NET for repeated capabilities in sections 4.3.4, 4.3.5, and 10.3.1.
Revision 2.0	October 14, 2016	Editorial Change – Modified header text for table 32.2 to indicate that the messages do not apply to .NET exceptions..

API Versions

Architecture	Drivers that comply with version 2.0 comply with all of the versions below
C	1.0, 2.0
COM	1.0, 2.0
.NET	2.0

Drivers that comply with this version of the specification also comply with earlier, compatible, versions of the specification as shown in the table above. The driver may benefit by advertising that it supports all the API versions listed in the table above.

1.Overview of the IviUpconverter Specification

1.1.Introduction

This specification defines the IVI class for frequency upconverters. The IviUpconverter class is designed to support the typical upconverter as common extended functionality found in more complex instruments. This section summarizes the *IviUpconverter Class Specification* and contains general information that the reader might need in order to understand, interpret, and implement aspects of this specification. These aspects include the following:

- IviUpconverter class overview
- The definitions of terms and acronyms
- References

1.2.IviUpconverter Class Overview

This specification defines the IVI class for upconverters. The IviUpconverter class is designed to support devices that convert IF input frequency signals to RF output frequencysignals..

The IviUpconverter class is divided into a base capability group and several extension groups. The base capability group is used to configure frequency, basic input and output filtering, and gain/attenuation stages. The IviUpconverter base capability group is described in Section 4, *IviUpconverterBase Capability Group*.

Extension groups are provided for more advanced capabilities. Several extension groups are provided to support different styles of sweeping the RF output. Other extension groups provide more precise control of signal routing through the various upconverter stages.

1.3.References

Several other documents and specifications are related to this specification. These other related documents are as follows:

- IVI-3.1: Driver Architecture Specification
- IVI-3.2: Inherent Capabilities Specification
- IVI-3.3: Standard Cross Class Capabilities Specification
- IVI-3.18: IVI.NET Utility Classes and Interfaces Specification
- IVI- 5.0: Glossary

1.4.Definitions of Terms and Acronyms

This section defines terms and acronyms that are specific to the IviUpconverter class.

Local Oscillator (LO)	An oscillator mixed with the IF input signal to produce the RF output.
Radio Frequency (RF)	Refers to the frequency output by the upconverter
Intermediate Frequency (IF)	Refers to the signal passed to the upconverter that is translated to the requested RF output signal.
I/Q Signals	A pair of quadrature input signals used for modulating the RF carrier output.
ALC	Automatic Level Control, Used for controlling the Upconverter output signal on a constant power.
AM	Amplitude Modulation

FM	Frequency Modulation
PM	Phase Modulation

Refer to *IVI–5.0: Glossary* for a description of more terms used in this specification.

2.IviUpconverter Class Capabilities

2.1.Introduction

The IviUpconverter specification divides generic upconverter capabilities into a base capability group and multiple extension capability groups. Each capability group is discussed in a separate section. This section defines names for each capability group and gives an overview of the information presented for each capability group.

2.2.IviUpconverter Group Names

The capability group names for the IviUpconverter class are defined in the following table. The Group Name is used to represent a particular capability group and is returned as one of the possible group names from the Class Group Capabilities attribute.

Table 2-1. IviUpconverter Group Names

Group Name	Description
`IviUpconverterBase`	Base Capabilities of the IviUpconverter specification. This group includes the ability to set the IF input frequency, RF output frequency, input attenuation, output gain, and other control parameters.
`IviUpconverterOutputGain`	Extension: IviUpconverter with the ability to specify the output gain.
`IviUpconverterOutputPowerLevel`	Extension: IviUpconverter with the ability to specify the output power level as an absolute value.
`IviUpconverterCalibration`	Extension: IviUpconcerter with the ability to perform self-calibration.
`IviUpconverterOutputReadyTrigger`	Extension: IviUpconverter with the ability to send a trigger signal when the RF output has settled to a point that is suitable for processing by a downstream component.
`IviUpconverterModulateAM`	Extension: IviUpconverter with the ability to apply amplitude modulation to an output signal.
`IviUpconverterModulateFM`	Extension: IviUpconverter with the ability to apply frequency modulation to an output signal.
`IviUpconverterModulatePM`	Extension: IviUpconverter with the ability to apply phase modulation to an output signal.
`IviUpconverterAnalogModulationSource`	Extension: IviUpconverter with at least one modulation source.
`IviUpconverterModulatePulse`	Extension: IviUpconverter with the ability to apply pulse modulation to an output signal.
`IviUpconverterBypass`	Extension: IviUpconverter with the ability to have the IF input completely bypass the upconverter.
`IviUpconverterSweep`	Extension: IviUpconverter with the ability to sweep the output frequency, power, or gain.
`IviUpconverterFrequencySweep`	Extension: IviUpconverter with the ability to sweep the output frequency.

Table 2-1. IviUpconverter Group Names

Group Name	Description
IviUpconverterPowerSweep	Extension: IviUpconverter with the ability to sweep the output power.
IviUpconverterGainSweep	Extension: IviUpconverter with the ability to sweep the upconverter gain.
IviUpconverterFrequencyStep	Extension: IviUpconverter with the ability to sweep the output frequency in discrete steps.
IviUpconverterPowerStep	Extension: IviUpconverter with the ability to sweep the output power in discrete steps.
IviUpconverterGainStep	Extension: IviUpconverter with the ability to sweep the upconverter gain in discrete steps.
IviUpconverterList	Extension: IviUpconverter with the ability to sweep the output frequency, power, or gain by a list of values.
IviUpconverterALC	Extension: IviUpconverter with the ability to use an automatic level control.
IviUpconverterAttenuatorHold	Extension: IviUpconverter with the ability to hold the RF output attenuators.
IviUpconverterReferenceOscillator	Extension: IviUpconverter with the ability to use an external reference frequency.
IviUpconverterSoftwareTrigger	Extension: IviUpconverter with the ability to trigger off of a software signal.
IviUpconverterModulateIQ	Extension: IviUpconverter with the ability to apply vector (IQ) modulation to an output signal.
IviUpconverterIQImpairment	Extension: IviUpconverter with the ability to apply impairment to vector (IQ) modulation.

2.3.Repeated Capability Names

The IviUpconverter Class Specification defines two repeated capabilities. Refer to the sections of IVI-3.1: *Driver Architecture Specification* that deal with repeated capabilities. They are Section 2.7, *Repeated Capabilities*, Section 4.1.9, *Repeated Capabilities*, Section 4.2.5, *Repeated Capabilities*, Section 4.3.9, *Repeated Capabilities*, and Section 5.9, *Repeated Capability Identifiers and Selectors*.

- IFInput
- RFOutput
- AnalogModulationSource

2.3.1.IFInput

In the configuration store, the repeated capability name for the IFInput capability shall be exactly one of "IFInput" or "IviUpconverterIFInput". Drivers that implement multiple repeated capabilities with the name "IFInput" shall use the latter form to disambiguate the name.

All IF Input related attributes and functions operate exclusively on the active IF Input, controlled by setting the Active IF Input attribute prior to using any IF Input related attributes and functions.

2.3.2.RFOutput

In the configuration store, the repeated capability name for the RFOutput capability shall be exactly one of "RFOutput" or "IviUpconverterRFOutput". Drivers that implement multiple repeated capabilities with the name "RFOutput" shall use the latter form to disambiguate the name.

All RF Output related attributes and functions operate exclusively on the active RF Output, controlled by setting the Active RF Output attribute prior to using any RF Output related attributes and functions.

2.3.3.AnalogModulationSource

In the configuration store, the repeated capability name for the AnalogModulationSource capability shall be exactly one of "AnalogModulationSource" or "IviUpconverterAnalogModulationSource". Drivers that implement multiple repeated capabilities with the name "AnalogModulationSource" shall use the latter form to disambiguate the name.

The Analog Modulation Source capability is used by the IviUpconverterAnalogModulationSource, IviUpconverterModulateAM, IviUpconverterModulateFM, and IviUpconverterModulatePM extension groups and shall be available only if the IviUpconverterAnalogModulationSource group is implemented.

2.4.Boolean Attribute and Parameter Values

This specification uses True and False as the values for Boolean attributes and parameters. The following table defines the identifiers that are used for True and False in the IVI.NET, IVI-COM, and IVI-C architectures.

Boolean Value	IVI.NET Identifier	IVI-COM Identifier	IVI-C Identifier
True	`true`	`VARIANT_TRUE`	`VI_TRUE`
False	`false`	`VARIANT_FALSE`	`VI_FALSE`

2.5..NET Namespace

The .NET namespace for the IviUpconverter class is Ivi.Upconverter.

2.6..NET IviUpconverter Session Factory

The IviUpconverter .NET assembly contains a factory method called Create for creating instances of IviUpconverter class-compliant IVI.NET drivers from driver sessions and logical names. Create is a static method accessible from the static IviUpconverter class.

Refer to *IVI-3.5: Configuration Server Specification* for a description of how logical names and session names are defined in the configuration store.

Refer to Section 8, *IVI.NET Specific Driver Constructor*, of *IVI-3.2: Inherent Capabilities Specification*, for more details on how the `idQuery`, `reset`, and `options` parameters affect the instantiation of the driver.

Refer to Section 4.3.11, *Multithread Safety*, of *IVI-3.1: Driver Architecture Specification* for a complete description of IVI.NET driver locking. Refer to Section 8, Table 8.2 *Required Lock Type Behavior for*

Drivers With the Same Access Key, of *IVI-3.2, Inherent Capability Specification*, for an explanation of how the values for `lockType` and `accessKey` are used to determine the kind of multithreaded lock to use for the driver instance.

.NET Method Prototype

```
IIviUpconverter Ivi.Upconverter.Create(String name);

IIviUpconverter Ivi.Upconverter.Create(String name,
                                        Boolean idQuery,
                                        Boolean reset);

IIviUpconverter Ivi.Upconverter.Create(String name,
                                        Boolean idQuery,
                                        Boolean reset,
                                        String options);

IIviUpconverter Ivi.Upconverter.Create(String resourceName,
                                        Boolean idQuery,
                                        Boolean reset,
                                        LockType lockType,
                                        String accessKey,
                                        String options);
```

Parameters

Inputs	Description	Base Type
`name`	A session name or a logical name that points to a session that uses an IVI.NET IviUpconverter class-compliant driver.	`String`
`idQuery`	Specifies whether to verify the ID of the instrument. The default is False.	`Boolean`
`reset`	Specifies whether to reset the instrument. The default is False.	`Boolean`
`lockType`	Specifies whether to use AppDomain-wide locking or machine-wide locking.	`Ivi.Driver.LockType`
`accessKey`	Specifies a user-selectable access key to identify the lock. Driver instances that are created with the same accessKey will be protected from simultaneous access by multiple threads within an AppDomain or across AppDomains, depending upon the value of the lockType parameter.	`String`
`options`	A string that allows the user to specify the initial values of certain inherent attributes. The default is an empty string.	`String`

Outputs	Description	Base Type
`Return Value`	Interface reference to the IIviUpconverter interface of the driver referenced by `session`.	`IIviUpconverter`

Defined Values

Name	*Description*		
		Language	*Identifier*
AppDomain	The lock is AppDomain-wide.		
		.NET	`Ivi.Driver.LockType.AppDomain`
Machine	The lock is machine-wide.		
		.NET	`Ivi.Driver.LockType.Machine`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

Usage

To create a driver that implements the IviUpconverter instrument class API from the logical name “My LogicalName”, use the following:

```
IIviUpconverter counter = IviUpconverter.Create(“MyLogicalName”);
```

In this case, the ID of the instrument will not be verified, the instrument will not be reset, and options will be supplied from the configuration store and/or driver defaults.

3.General Requirements

This section describes the general requirements a specific driver must meet in order to be compliant with this specification. In addition, it provides general requirements that specific drivers must meet in order to comply with a capability group, attribute, or function.

3.1.Minimum Class Compliance

To be compliant with the IviUpconverter Class Specification, an IVI specific driver shall conform to all of the requirements for an IVI class-compliant specific driver as specified in *IVI-3.1: Driver Architecture Specification,* implement the inherent capabilities that *IVI-3.2: Inherent IVI Capabilities Specification* defines, and implements the IviUpconverterBase capability group.

3.1.1.Disable

Refer to *IVI-3.2: Inherent Capabilities Specification* for the prototype of this function. The IviUpconverter specification does not define additional requirements on the Disable function.

3.2.Capability Group Compliance

IVI-3.1: Driver Architecture Specification defines the general rules for a specific driver to be compliant with a capability group.

4. IviUpconverterBase Capability Group

4.1. Overview

The IviUpconverterBase Capability Group supports basic upconverter operation. The IviUpconverterBase Capability Group defines attributes and functions for configuring and/or accessing IF input frequency, and attenuation as well as RF output frequency and gain. This Capability Group also defines functions for using an external local oscillator (LO).

This specification defines an IF Input repeated capability to accommodate devices that offer multiple IF inputs. Many devices, however, offer only a single IF input. The block diagram below presents the basic aspects of a single-input upconverter.

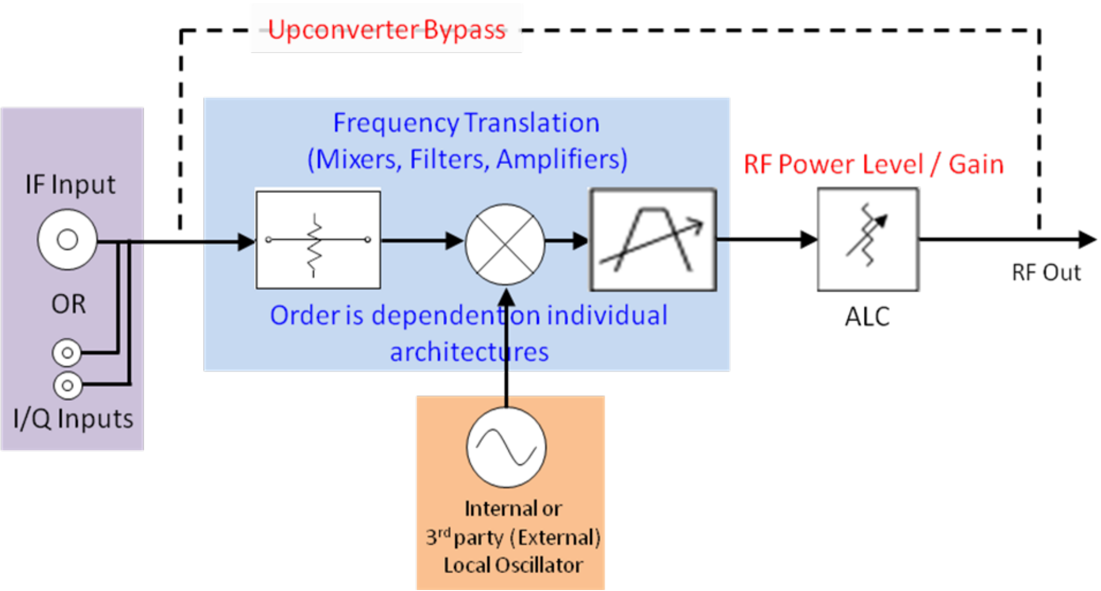


Figure 41. Single-Input IviUpconverter Block Diagram

The diagram above illustrates an upconverter that has a single IF or I/Q input feeding the front-end elements of the device. Figure 41 shows a frequency translation section which includes a mixer, filter, and amplifier followed by an RF gain stage. Depending upon the design of the upconverter, the order in which these elements appears may vary.

Some upconverters may offer multiple IF inputs along with multiple RF outputs. This specification does not dictate whether the RF outputs are dedicated to specific IF inputs or whether they are shared amongst the multiple IF inputs. Vendors are encouraged to use a repeated capability naming scheme that indicates whether or not RF outputs are associated with specific IF inputs.

Note: ***The IviUpconverterBase capability group does not support a specific mechanism for controlling the RF output power. Instead, an upconverter must support either the IviUpconverterOutputGain or IviUpconverterOutputPowerLevel Extension Groups or both. This organization is required because many upconverters support only one of these extension groups. If an upconverter supports more than one of these extensions, the most recent setting will be applied.***

4.2.IviUpconverterBase Attributes

The IviUpconverterBase capability group defines the following attributes:

- Active IF Input
- Active RF Output
- ALC Enabled
- Auto Corrections Enabled
- External LO Enabled
- External LO Frequency
- IF Input Attenuation
- IF Input Count
- IF Input Coupling
- IF Input Frequency
- IF Input Name (IVI-COM Only)
- Is Ready
- RF Output Bandwidth
- RF Output Count
- RF Output Enabled
- RF Output Frequency
- RF Output Name (IVI-COM Only)

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

4.2.1.Active IF Input

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	Set Active IF Input (IVI-C Only)

.NET Property Name

```
IFInput.ActiveIFInput
```

COM Property Name

```
IFInput.ActiveIFInput
```

C Constant Name

```
IVIUPCONVERTER_ATTR_ACTIVE_IF_INPUT
```

Description

Specifies the IF input that is currently active. Subsequent calls to functions and attributes that are based on the IF Input repeated capability will be applied to the Active IF Input specified here. The values for this attribute correspond to the allowed repeated capability names for the IF Input repeated capability. If the driver defines a qualified IF Input name, this attribute returns the qualified name. Use the IF Input Name attribute (for IVI-COM) or the Get IF Input Name function (for IVI-C) to read the allowed values for this attribute.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.2.Active RF Output

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	Set Active RF Output (IVI-C Only)

.NET Property Name

`RFOutput.ActiveRFOutput`

COM Property Name

`RFOutput.ActiveRFOutput`

C Constant Name

`IVIUPCONVERTER_ATTR_ACTIVE_RF_OUTPUT`

Description

Specifies the RF output that is currently active. Subsequent calls to functions and attributes that are based on the RF Output repeated capability will be applied to the Active RF Output specified here. The values for this attribute correspond to the allowed repeated capability names for the RF Output repeated capability. If the driver defines a qualified RF Output name, this attribute returns the qualified name. Use the RF Output Name attribute (for IVI-COM) or the Get RF Output Name function (for IVI-C) to read the allowed values for this attribute.

Note that the Active RF Output attribute does not enable the specified output. This attribute only controls the RF Output repeated capability instance to which other functions and attributes apply. Use the RF Output Enabled attribute to route the RF signal to a specific output.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.3.ALC Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	N/A

.NET Property Name

Alc.Enabled

COM Property Name

ALC.Enabled

C Constant Name

IVIUPCONVERTER_ATTR_ALC_ENABLED

Description

Enables or disables the Automatic Level Control (ALC).

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.4.Auto Corrections Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
ViBoolean	R/W	IFInput	None	N/A

.NET Property Name

```
IFInput.AutoCorrectionsEnabled
```

COM Property Name

```
IFInput.AutoCorrectionsEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_AUTO_CORRECTIONS_ENABLED
```

Description

Enables or disables automatic global corrections on the device.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.5.External LO Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	N/A

.NET Property Name

```
ExternalLO.Enabled
```

COM Property Name

```
ExternalLO.Enabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_EXTERNAL_LO_ENABLED
```

Description

Enables or disables the external LO.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.6.External LO Frequency

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	N/A	None	N/A

.NET Property Name

ExternalLO.Frequency

COM Property Name

ExternalLO.Frequency

C Constant Name

IVIUPCONVERTER_ATTR_EXTERNAL_LO_FREQUENCY

Description

Specifies the frequency of the external LO. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.7.IF Input Attenuation

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	IFInput	Up	Configure IF Input Attenuation

.NET Property Name

`IFInput.Attenuation`

COM Property Name

`IFInput.Attenuation`

C Constant Name

`IVIUPCONVERTER_ATTR_IF_INPUT_ATTENUATION`

Description

Specifies the amount of attenuation (or gain) to apply to the active IF input of the upconverter. The units are dB. Positive values for this attribute represent attenuation while negative values represent gain.

If an instrument only supports a fixed IF attenuation, the driver should accept that value for the IF Input Attenuation attribute and return Invalid Value error for any other value.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.8.IF Input Count

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	RO	N/A	None	N/A

.NET Property Name

```
IFInput.Count
```

COM Property Name

```
IFInput.Count
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IF_INPUT_COUNT
```

Description

Returns the number of IF Inputs available on the device.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.9.IF Input Coupling

Data Type	Access	Applies To	Coercion	High Level Functions
ViInt32	R/W	IFInput	None	N/A

.NET Property Name

IFInput.Coupling

.NET Enumeration Name

ExternalCoupling

COM Property Name

IFInput.Coupling

COM Enumeration Name

IviUpconverterIFInputCouplingEnum

C Constant Name

IVIUPCONVERTER_ATTR_IF_INPUT_COUPLING

Description

Specifies the coupling applied to active IF input.

Defined Values

Name	*Description*		
		Language	*Identifier*
AC	The upconverter AC couples the IF input signal.		
		.NET	ExternalCoupling.AC
		C	IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_AC
		COM	IviUpconverterIFInputCouplingAC
DC	The upconverter DC couples the IF input signal.		
		.NET	ExternalCoupling.DC
		C	IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_DC
		COM	IviUpconverterIFInputCouplingDC

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_CLASS_EXT_BASE` and less than `IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_EXT_BASE.`

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_SPECIFIC_EXT_BASE`.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IF Input Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of IF Input Coupling Specific Ext Base, `IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_SPECIFIC_EXT_BASE` and `IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_CLASS_EXT_BASE`.

4.2.10.IF Input Frequency

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	RO	IFInput	None	N/A

.NET Property Name

```
IFInput.Frequency
```

COM Property Name

```
IFInput.Frequency
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IF_INPUT_FREQUENCY
```

Description

Returns the frequency of the active IF input. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.11.IF Input Name (IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	RO	IFInput	None	N/A

.NET Property Name

```
N/A
```

(Use the `IFInput.GetName()` method).

COM Property Name

```
HRESULT IFInput.Name ([in] LONG Index, [out,retval] BSTR* Name);
```

C Constant Name

```
N/A
```

(Use the `IviUpconverter_GetIFInputName()` function).

Description

Returns the IF Input identifier that corresponds to the one-based index that the user specifies. If the driver defines a qualified IF Input name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the IF Input Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

4.2.12. Is Ready

Data Type	Access	Applies To	Coercion	High Level Functions
ViBoolean	RO	RFOutput	None	N/A

.NET Property Name

```
RFOutput.IsReady
```

COM Property Name

```
RFOutput.IsReady
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IS_READY
```

Description

Indicates whether the upconverter has settled from changes to either the IF input signal or changes to device control parameters, such as RF Output Gain, RF Frequency, or IF Attenuation. This attribute indicates whether or not the RF output is valid for processing by another downstream system component, such as a digitizer.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.13.RF Output Bandwidth

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	RFOutput	Up	N/A

.NET Property Name

```
RFOutput.Bandwidth
```

COM Property Name

```
RFOutput.Bandwidth
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_BANDWIDTH
```

Description

Specifies the maximum effective RF signal bandwidth of the upconverter's active RF Output . The units are Hertz. This value is a measure of the spectral width between two points for which the amplitude profile is 3 dB below a peak close to mid band.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.14.RF Output Count

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	RO	N/A	None	N/A

.NET Property Name

```
RFOutput.Count
```

COM Property Name

```
RFOutput.Count
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_COUNT
```

Description

Returns the number of RF Outputs available on the device.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.15.RF Output Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	RFOutput	None	Configure RF Output Enabled

.NET Property Name

```
RFOutput.Enabled
```

COM Property Name

```
RFOutput.Enabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_ENABLED
```

Description

Enables or disables the active RF output. Only a single RF Output can be enabled at a time. Thus, when this attribute is set to True for a particular RF output, all other RF outputs are disabled.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.16.RF Output Frequency

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	RFOutput	None	Configure RF Output Frequency

.NET Property Name

```
RFOutput.Frequency
```

COM Property Name

```
RFOutput.Frequency
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_FREQUENCY
```

Description

Specifies the frequency of the active RF output. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

4.2.17. RF Output Name (IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	RO	RFOutput	None	N/A

.NET Property Name

`N/A`

(Use the `RFOutput.GetName()` function).

COM Property Name

```
HRESULT RFOutput.Name ([in] LONG Index,
                       [out,retval] BSTR* Name);
```

C Constant Name

`N/A`

(Use the `IviUpconverter_GetRFOutputName()` function).

Description

Returns the RF Output identifier that corresponds to the one-based index that the user specifies. If the driver defines a qualified RF Output name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the RF Output Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

4.3.IviUpconverterBase Functions

The IviUpconverterBase capability group defines the following functions:

- Configure IF Input Attenuation (IVI-C Only)
- Configure IF Input Frequency (IVI-C Only)
- Configure RF Output Enabled (IVI-C Only)
- Get IF Input Name (IVI-C & IVI.NET Only)
- Get RF Output Name (IVI-C & IVI.NET Only)
- Set Active IF Input (IVI-C Only)
- Set Active RF Output (IVI-C Only)
- Wait Until Ready

This section describes the behavior and requirements of each function.

4.3.1.Configure IF Input Attenuation (IVI-C Only)

Description

Configures the active IF input attenuation.

.NET Method Prototype

```
N/A
```

(use the `IFInput.Attenuation` property)

COM Method Prototype

```
N/A
```

(use the `IFInput.Attenuation` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureIFInputAttenuation (ViSession Vi,
                                        ViReal64 Attenuation);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Attenuation`	Specifies the IF input attenuation. The driver uses this value to set the IF Input Attenuation attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

4.3.2. Configure RF Output Enabled (IVI-C Only)

Description

Configures whether or not the active RF output is enabled.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureRFOutputEnabled (ViSession Vi,
                                                   ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Specifies whether or not the RF output is enabled. The driver uses this value to set the RF Output Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

4.3.3.Configure RF Output Frequency (IVI-C Only)

Description

Configures the active RF output frequency.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.Frequency` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.Frequency` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureRFOutputFrequency (ViSession Vi,
                                    ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Frequency`	Specifies the RF output frequency. The driver uses this value to set the RF Output Frequency attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

4.3.4.Get IF Input Name (IVI-C & IVI.NET Only)

Description

This function returns the specific driver defined IF Input name that corresponds to the index that the user specifies. If the driver defines a qualified IF Input name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the IF Input Count, the function returns an empty string in the Name parameter and returns the Invalid Value error.

In C, the index is one-based. In .NET, the index is zero-based.

.NET Method Prototype

```
String IFInput.GetName (Int32 index);
```

COM Method Prototype

```
N/A
```

(use the `IFInput.Name` property)

C Prototype

```
ViStatus IviUpconverter_GetIFInputName (ViSession Vi,
                                        ViInt32 Index,
                                        ViInt32 NameBufferSize,
                                        ViChar Name[]);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Index`	An index (one-based for IVI-C, zero-based for IVI.NET) that defines which name to return.	`ViInt32`
`NameBufferSize`	The number of bytes in the ViChar array that the user specifies for the `Name` parameter.	`ViInt32`

Outputs	Description	Base Type
`Name`	A user-allocated (for IVI-C) or driver-allocated (for IVI-COM) buffer into which the driver stores the IF Input name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

4.3.5.Get RF Output Name (IVI-C & IVI.NET Only)

Description

This function returns the specific driver defined RF Output name that corresponds to the index that the user specifies. If the driver defines a qualified RF Output name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the RF Output Count, the function returns an empty string in the Name parameter and returns the Invalid Value error.

In C, the index is one-based. In .NET, the index is zero-based.

.NET Method Prototype

```
String RFOutput.GetName (Int32 index);
```

COM Method Prototype

```
N/A
```

(use the `RFOutput.Name` property)

C Prototype

```
ViStatus IviUpconverter_GetRFOutputName (ViSession Vi,
                                         ViInt32 Index,
                                         ViInt32 NameBufferSize,
                                         ViChar Name[]);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Index`	An index (one-based for IVI-C, zero-based for IVI.NET) that defines which name to return.	`ViInt32`
`NameBufferSize`	The number of bytes in the ViChar array that the user specifies for the `Name` parameter.	`ViInt32`

Outputs	Description	Base Type
`Name`	A user-allocated (for IVI-C) or driver-allocated (for IVI-COM) buffer into which the driver stores the RF Output name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

4.3.6.Set Active IF Input (IVI-C Only)

Description

This function selects one of the available IF inputs, and makes it the active IF input.

.NET Method Prototype

```
N/A
```

(use the `IFInput.ActiveIFInput` property)

COM Method Prototype

```
N/A
```

(use the `IFInput.ActiveIFInput` property)

C Prototype

```
ViStatus IviUpconverter_SetActiveIFInput (ViSession Vi,
                                          ViConstString Name);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Name`	IF Input to be selected. The driver uses this value to set the Active IF Input attribute. See the attribute description for more details.	`ViConstString`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

4.3.7.Set Active RF Output (IVI-C Only)

Description

This function selects one of the available RF outputs, and makes it the active RF output.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.ActiveRFOutput` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.ActiveRFOutput` property)

C Prototype

```
ViStatus IviUpconverter_SetActiveRFOutput (ViSession Vi,
                                           ViConstString Name);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Name`	RF Output to be selected. The driver uses this value to set the Active RF Output attribute. See the attribute description for more details.	`ViConstString`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

4.3.8.Wait Until Ready

Description

This function waits until all of the signals flowing through the upconverter have settled. For C and COM, if the signals did not settle within the time period the user specified with the MaxTimeMilliseconds parameter, the function returns the Max Time Exceeded error. For .NET, if the signals did not settle within the time period the user specified with the maximumTime parameter, the function throws the Max Time Exceeded exception.

.NET Method Prototype

```
void RFOutput.WaitUntilReady (Ivi.Driver.PrecisionTimeSpan maximumTime);
```

COM Method Prototype

```
HRESULT RFOutput.WaitUntilReady ([in] long MaxTimeMilliseconds);
```

C Prototype

```
ViStatus IviUpconverter_WaitUntilReady (ViSession Vi,
                                        ViInt32 MaxTimeMilliseconds);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
MaxTimeMilliseconds (C/COM)	Specifies the maximum time the end-user allows for this function to complete. The units are milliseconds. Defined values: Max Time Immediate - The function returns immediately. If the upconverter is not ready, the function returns an error. Max Time Infinite - The function waits indefinitely for the upconverter to settle.	ViInt32
maximumTime (.NET)	Specifies the maximum time the end-user allows for this function to complete. The units are implicit in the PrecisionTimeSpan type. Defined values: PrecisionTimeSpan.Zero - The function returns immediately. If the upconverter has not settled, the function returns an error. PrecisionTimeSpan.MaxValue - The function waits indefinitely for the upconverter to settle.	PrecisionTimeSpan

Defined Values for the MaxTimeMilliseconds Parameter (C/COM)

Name	*Description*		
		Language	*Identifier*
Max Time Immediate	The function returns immediately.		
		C	`IVIUPCONVERTER_VAL_MAX_TIME_IMMEDIATE`
		COM	`IviUpconverterTimeOutImmediate`
Max Time Infinite	The function waits indefinitely for the upconverter to settle.		
		C	`IVIUPCONVERTER_VAL_MAX_TIME_INFINITE`
		COM	`IviUpconverterTimeOutInfinite`

Defined Values for the maximumTime Parameter (.NET)

Name	*Description*		
		Language	*Identifier*
`Zero`	Sets timeout to immediate. The function returns immediately.		
		.NET	`TimeSpan.Zero`
`MaxValue`	Sets timeout to infinite. The function waits indefinitely for the settling to complete.		
		.NET	`TimeSpan.MaxValue`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return. The table below specifies additional class-defined status codes for this function.

Completion Codes	Description
Max Time Exceeded	Error: Maximum time exceeded before the operation completed.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

Compliance Notes

The specific instrument driver is not required to support any of the defined values for the `MaxTimeMilliseconds` parameter.

4.4.IviUpconverterBase Behavior Model

The following state diagram shows the basic operation of the IviUpconverter.

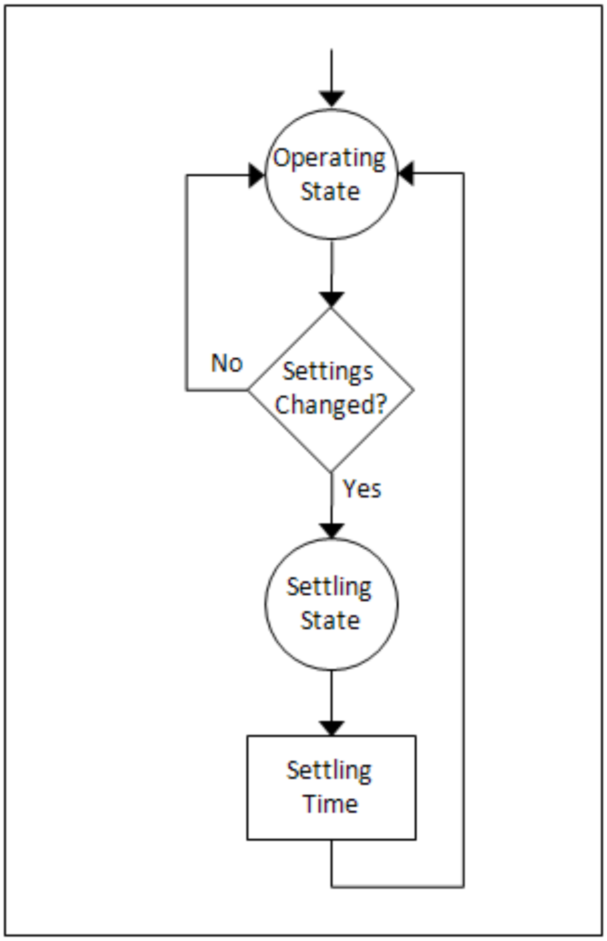


Figure 42: IviUpconverterBase Behavior Model

The behavior model for the IviUpconverterBase capability group is simple. The device is always operating (assuming power is applied) with an RF signal appearing at the output in response to an IF signal applied to the input. This specification defines a transient Settling state to account for the fact that the RF output signal may need to settle to a stable state after some operational parameters have been changed. Thus, in order to perform more accurate measurements, users will typically want to wait for a certain amount of time after changing some parameters to allow the signal to settle. A number of factors might require settling time, including (but not limited to) the following attributes defined by this specification:

- IF Input Attenuation
- IF Input Frequency
- RF Output Gain

5.IviUpconverterOutputGain Extension Group

5.1.IviUpconverterOutputGain Overview

The IviUpconverterOutputGain Extension Group supports upconverters with the ability to specify the output power gain.

5.2.IviUpconverterOutputGain Attributes

The IviUpconverterOutputGain extension group defines the following attributes:

- RF Output Gain

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

5.2.1.RF Output Gain

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	RFOutput	Down	Configure RF Output Gain

.NET Property Name

```
RFOutput.Gain
```

COM Property Name

```
RFOutput.Gain
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_GAIN
```

Description

Specifies the amount of gain (or attenuation) to apply to the active RF output of the upconverter. The units are dB. Positive values for this attribute represent signal gain while negative values represent attenuation.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

5.3.IviUpconverterOutputGain Functions

The IviUpconverterOutputGain extension group defines the following function:

- Configure RF Output Gain (IVI-C Only)

This section describes the behavior and requirements of this function.

5.3.1.Configure RF Output Gain (IVI-C Only)

Description

Configures the amount of gain (or attenuation) to apply to the active RF output of the upconverter.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.Gain` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.Gain` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureRFOutputGain (ViSession Vi,
                                               ViReal64 Gain);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Gain`	Specifies the RF output gain. The driver uses this value to set the RF Output Gain attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

5.4.IviUpconverterOutputGain Behavior Model

The IviUpconverterOutputGain Extension Group follows the behavior model of the IviUpconverterBase capability group.

5.5.IviUpconverterOutputGain Compliance Notes

A specific driver must implement either the *IviUpconverterOutputGain* Extension Group, the *IviUpconverterOutputPowerLevel* Extension group, or both.

6.IviUpconverterOutputPowerLevel Extension Group

6.1.IviUpconverterOutputPowerLevel Overview

The IviUpconverterOutputPowerLevel Extension Group supports upconverters with the ability to specify the absolute output power level.

6.2.IviUpconverterOutputPowerLevel Attributes

The IviUpconverterOutputPowerLevel extension group defines the following attributes:

- RF Output Level

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

6.2.1.RF Output Level

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	RFOutput	Down	Configure RF Output Level (IVI-C Only)

.NET Property Name

```
RFOutput.Level
```

COM Property Name

```
RFOutput.Level
```

C Constant Name

```
IVIUPCONVERTER_ATTR_RF_OUTPUT_LEVEL
```

Description

Specifies the output power that should appear at the active RF output of the upconverter. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

6.3.IviUpconverterOutputPowerLevel Functions

The IviUpconverterOutputPowerLevel extension group defines the following function:

- Configure RF Output Level (IVI-C Only)

This section describes the behavior and requirements of this function.

6.3.1.Configure RF Output Level (IVI-C Only)

Description

Configures the power level in dBm that should appear at the active RF output of the upconverter.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.Level` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.Level` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureRFOutputLevel (ViSession Vi,
                                                 ViReal64 Level);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Level`	Specifies the RF output level. The driver uses this value to set the RF Output Level attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

6.4.IviUpconverterOutputPowerLevel Behavior Model

The IviUpconverterOutputPowerLevel Extension Group follows the behavior model of the IviUpconverterBase capability group.

6.5.IviUpconverterOutputPowerLevel Compliance Notes

A specific driver must implement either the *IviUpconverterOutputGain* Extension Group, the *IviUpconverterOutputPowerLevel* Extension group, or both.

7.IviUpconverterModulateAM Extension Group

7.1.IviUpconverterModulateAM Overview

The IviUpconverterModulateAM Extension Group supports upconverters that can apply amplitude modulation to the RF output signal. The user can enable or disable amplitude modulation, specify the source and coupling of the modulating signal and the modulation depth with linear or logarithmic attenuation.

7.2.IviUpconverterModulateAM Attributes

The IviUpconverterModulateAM extension group defines the following attributes:

- AM Depth
- AM Enabled
- AM External Coupling
- AM Nominal Voltage
- AM Scaling
- AM Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

7.2.1.AM Depth

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure AM

.NET Property Name

```
AnalogModulation.AM.Depth
```

COM Property Name

```
AnalogModulation.AM.Depth
```

C Constant Name

```
IVIUPCONVERTER_ATTR_AM_DEPTH
```

Description

Specifies the extent of modulation the upconverter applies to the RF output signal with the modulating signal as a result of summing all sources -- internal and external. The amount of the specified modulation depth is achieved with a modulating voltage of AM Nominal Voltage. If the AM Scaling attribute is set to Linear, then the units are percent (%). If the AM Scaling attribute is set to logarithmic, then the units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

7.2.2.AM Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure AM Enabled (IVI-C Only)

.NET Property Name

```
AnalogModulation.AM.Enabled
```

COM Property Name

```
AnalogModulation.AM.Enabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_AM_ENABLED
```

Description

Specifies whether the upconverter enables or disables amplitude modulation of the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

7.2.3.AM External Coupling

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure AM External Coupling (IVI-C Only)

.NET Property Name

`AnalogModulation.AM.ExternalCoupling`

.NET Enumeration Name

`ExternalCoupling`

COM Property Name

`AnalogModulation.AM.ExternalCoupling`

COM Enumeration Name

`IviUpconverterAMExternalCouplingEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_AM_EXTERNAL_COUPLING`

Description

Specifies the coupling of the external source of the modulating signal.

Defined Values

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterAMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterAMExternalCouplingDC`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_EXT_BASE.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to AM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM External Coupling Specific Ext Base, IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE.

7.2.4.AM Nominal Voltage

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	RO	N/A	None	N/A

.NET Property Name

AnalogModulation.AM.NominalVoltage

COM Property Name

AnalogModulation.AM.NominalVoltage

C Constant Name

IVIUPCONVERTER_ATTR_AM_NOMINAL_VOLTAGE

Description

Returns the voltage at which the instrument achieves the amount of modulation specified by the AM Depth attribute.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

7.2.5.AM Scaling

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure AM

.NET Property Name

`AnalogModulation.AM.Scaling`

.NET Enumeration Name

`Scaling`

COM Property Name

`AnalogModulation.AM.Scaling`

COM Enumeration Name

`IviUpconverterAMScalingEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_AM_SCALING`

Description

Specifies linear or logarithmic characteristic for amplitude modulation. The units of the AM Depth attribute is changed with this setting.

Defined Values

Name	*Description*		
		Language	*Identifier*
Linear	Enables linear attenuation for amplitude modulation.		
		.NET	`Scaling.Linear`
		C	`IVIUPCONVERTER_VAL_AM_SCALING_LINEAR`
		COM	`IviUpconverterAMScalingLinear`
Logarithmic	Enables logarithmic attenuation for amplitude modulation.		
		.NET	`Scaling.Logarithmic`
		C	`IVIUPCONVERTER_VAL_AM_SCALING_LOGARITHMIC`
		COM	`IviUpconverterAMScalingLogarithmic`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_AM_SCALING_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_AM_SCALING_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_AM_SCALING_SPECIFIC_EXT_BASE.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to AM Scale Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM Scaling Specific Ext Base, IVIUPCONVERTER_VAL_AM_SCALING_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_AM_SCALING_CLASS_EXT_BASE.

7.2.6.AM Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	N/A	None	Configure AM

.NET Property Name

```
AnalogModulation.AM.Source
```

COM Property Name

```
AnalogModulation.AM.Source
```

C Constant Name

```
IVIUPCONVERTER_ATTR_AM_SOURCE
```

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all specified sources are summed. Multiple source names are separated by commas.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

7.3.IviUpconverterModulateAM Functions

The IviUpconverterModulateAM extension group defines the following function:

- Configure AM
- Configure AM Enabled (IVI-C Only)
- Configure AM External Coupling (IVI-C Only)

This section describes the behavior and requirements of this function.

7.3.1.Configure AM

Description

This function configures the attributes that control the upconverter's amplitude modulation.

COM Method Prototype

```
void AnalogModulation.AM.Configure (String source,
                                    Scaling scaling,
                                    Double depth);
```

COM Method Prototype

```
HRESULT AnalogModulation.AM.Configure ([in] BSTR Source,
                                       [in] IviUpconverterAMScalingEnum
Scaling,
                                       [in] double Depth);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureAM (ViSession Vi,
                                     ViConstString Source,
                                     ViInt32 Scaling,
                                     ViReal64 Depth);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Source	Specifies the source of the signal that the upconverter uses to modulate the output signal. The driver uses this value to set the AM Source attribute. See the attribute description for more details.	ViConstString
Scaling	Specifies the scaling of the modulation. The driver uses this value to set the AM Scaling attribute. See the attribute description for more details.	ViInt32
Depth	Specifies the extent of modulation. The driver uses this value to set the AM Depth attribute. See the attribute description for more details.	ViReal64

Defined Values for the Scaling Parameter

Name	*Description*		
		Language	*Identifier*
Linear	Enables linear attenuation for amplitude modulation.		
		.NET	`Scaling.Linear`
		C	`IVIUPCONVERTER_VAL_AM_SCALING_LINEAR`
		COM	`IviUpconverterAMScalingLinear`
Logarithmic	Enables logarithmic attenuation for amplitude modulation.		
		.NET	`Scaling.Logarithmic`
		C	`IVIUPCONVERTER_VAL_AM_SCALING_LOGARITHMIC`
		COM	`IviUpconverterAMScalingLogarithmic`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

7.3.2.Configure AM Enabled (IVI-C Only)

Description

Configures the upconverter to apply amplitude modulation to the RF output signal.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.AM.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.AM.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureAMEnabled (ViSession Vi,
                                            ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Specifies whether or not AM modulation is enabled. The driver uses this value to set the AM Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

7.3.3. Configure AM External Coupling (IVI-C Only)

Description

Configures the coupling of the external source the upconverter uses for amplitude modulation of the output signal.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.AM.ExternalCoupling` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.AM.ExternalCoupling` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureAMExternalCoupling (ViSession Vi,
                                           ViInt32 Coupling);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Coupling`	Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the AM External Coupling attribute. See the attribute description for more details.	`ViInt32`

Defined Values for the Coupling Parameter

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterAMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterAMExternalCouplingDC`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

7.4.IviUpconverterModulateAM Behavior Model

The IviUpconverterModulateAM Extension Group follows the behavior model of the IviUpconverterBase capability group.

7.5.IviUpconverterModulateAM Compliance Notes

1. If a specific driver implements the IviUpconverterModulateAM Extension Group, it shall also implement the IviUpconverterAnalogModulationSource Extension Group.
2. If a specific driver does not support an external modulation source, it shall not support the AM External Coupling attribute or the Configure AM External Coupling function.

8.IviUpconverterModulateFM Extension Group

8.1.IviUpconverterModulateFM Overview

The IviUpconverterModulateFM Extension Group supports upconverters that can apply frequency modulation to the RF output signal. The user can enable or disable frequency modulation, specify the source and coupling of the modulating signal and the peak frequency deviation.

8.2.IviUpconverterModulateFM Attributes

The IviUpconverterModulateFM extension group defines the following attributes:

- FM Deviation
- FM Enabled
- FM External Coupling
- FM Nominal Voltage
- FM Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

8.2.1.FM Deviation

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure FM

.NET Property Name

AnalogModulation.FM.Deviation

COM Property Name

AnalogModulation.FM.Deviation

C Constant Name

IVIUPCONVERTER_ATTR_FM_DEVIATION

Description

Specifies the extent of modulation (peak frequency deviation) the upconverter applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The amount of the specified FM modulation deviation is achieved with a modulating voltage of FM Nominal Voltage. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

8.2.2.FM Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure FM Enabled (IVI-C Only)

.NET Property Name

`AnalogModulation.FM.Enabled`

COM Property Name

`AnalogModulation.FM.Enabled`

C Constant Name

`IVIUPCONVERTER_ATTR_FM_ENABLED`

Description

Specifies whether the upconverter enables or disables frequency modulation of the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

8.2.3.FM External Coupling

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure FM External Coupling (IVI-C Only)

.NET Property Name

`AnalogModulation.FM.ExternalCoupling`

.NET Enumeration Name

`ExternalCoupling`

COM Property Name

`AnalogModulation.FM.ExternalCoupling`

COM Enumeration Name

`IviUpconverterFMExternalCouplingEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_FM_EXTERNAL_COUPLING`

Description

Specifies the coupling of the external source of the modulating signal.

Defined Values

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterFMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterFMExternalCouplingDC`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_EXT_BASE.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to FM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of FM External Coupling Specific Ext Base, IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE.

8.2.4.FM Nominal Voltage

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	RO	N/A	None	N/A

.NET Property Name

```
AnalogModulation.FM.NominalVoltage
```

COM Property Name

```
AnalogModulation.FM.NominalVoltage
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FM_NOMINAL_VOLTAGE
```

Description

Returns the voltage at which the instrument achieves the amount of modulation specified by the FM Deviation attribute. The units are Volts.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

8.2.5.FM Source

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	Configure FM

.NET Property Name

```
AnalogModulation.FM.Source
```

COM Property Name

```
AnalogModulation.FM.Source
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FM_SOURCE
```

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources are summed. Multiple source names are separated by commas.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

8.3.IviUpconverterModulateFM Functions

The IviUpconverterModulateFM extension group defines the following function:

- Configure FM
- Configure FM Enabled (IVI-C Only)
- Configure FM External Coupling (IVI-C Only)

This section describes the behavior and requirements of this function.

8.3.1.Configure FM

Description

This function configures the attributes that control the upconverter's frequency modulation.

.NET Method Prototype

```
void AnalogModulation.FM.Configure (String source,
                                    Double deviation);
```

COM Method Prototype

```
HRESULT AnalogModulation.FM.Configure ([in] BSTR Source,
                                       [in] double Deviation);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureFM (ViSession Vi,
                                     ViConstString Source,
                                     ViReal64 Deviation);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Source	Specifies the modulating source. The driver uses this value to set the FM Source attribute. See the attribute description for more details.	ViConstString
Deviation	Specifies the extent of modulation. The driver uses this value to set the FM Deviation attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

8.3.2.Configure FM Enabled (IVI-C Only)

Description

Configures the upconverter to apply frequency modulation to the RF output signal.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.FM.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.FM.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureFMEnabled (ViSession Vi,
                                            ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Specifies whether the upconverter enables or disables frequency modulation of the RF output signal. The driver uses this value to set the FM Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

8.3.3.Configure FM External Coupling (IVI-C Only)

Description

Configures the coupling of the external source the upconverter uses for frequency modulation of the output signal.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.FM.ExternalCoupling` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.FM.ExternalCoupling` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureFMExternalCoupling (ViSession Vi,
                                    ViInt32 Coupling);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Coupling`	Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the FM External Coupling attribute. See the attribute description for more details.	`ViInt32`

Defined Values for the Coupling Parameter

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterFMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterFMExternalCouplingDC`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

8.4.IviUpconverterModulateFM Behavior Model

The IviUpconverterModulateFM Extension Group follows the behavior model of the IviUpconverterBase capability group.

8.5.IviUpconverterModulateFM Compliance Notes

1. If a specific driver implements the IviUpconverterModulateFM Extension Group, it shall also implement the IviUpconverterAnalogModulationSource Extension Group.

2. If a specific driver does not support an external modulation source, it shall not support the FM External Coupling attribute or the Configure FM External Coupling function.

9.IviUpconverterModulatePM Extension Group

9.1.IviUpconverterModulatePM Overview

The IviUpconverterModulatePM Extension Group supports upconverters that can apply phase modulation to the RF output signal. The user can enable or disable phase modulation, specify the source and coupling of the modulating signal and the peak phase deviation.

9.2.IviUpconverterModulatePM Attributes

The IviUpconverterModulatePM extension group defines the following attributes:

- PM Deviation
- PM Enabled
- PM External Coupling
- PM Nominal Voltage
- PM Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

9.2.1.PM Deviation

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure PM

.NET Property Name

`AnalogModulation.PM.Deviation`

COM Property Name

`AnalogModulation.PM.Deviation`

C Constant Name

`IVIUPCONVERTER_ATTR_PM_DEVIATION`

Description

Specifies the extent of modulation (peak phase deviation) the upconveter applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The amount of the specified PM modulation deviation is achieved with a modulating voltage of PM Nominal Voltage. The units are radians.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

9.2.2.PM Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure PM Enabled (IVI-C Only)

.NET Property Name

`AnalogModulation.PM.Enabled`

COM Property Name

`AnalogModulation.PM.Enabled`

C Constant Name

`IVIUPCONVERTER_ATTR_PM_ENABLED`

Description

Specifies whether the upconverter enables or disables phase modulation of the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

9.2.3.PM External Coupling

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure PM External Coupling (IVI-C Only)

.NET Property Name

`AnalogModulation.PM.ExternalCoupling`

.NET Enumeration Name

`ExternalCoupling`

COM Property Name

`AnalogModulation.PM.ExternalCoupling`

COM Enumeration Name

`IviUpconverterPMExternalCouplingEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_PM_EXTERNAL_COUPLING`

Description

Specifies the coupling of the external source of the modulating signal.

Defined Values

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterPMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterPMExternalCouplingDC`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_EXT_BASE.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to PM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of PM External Coupling Specific Ext Base, IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE.

9.2.4.PM Nominal Voltage

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	RO	N/A	None	N/A

.NET Property Name

`AnalogModulation.PM.NominalVoltage`

COM Property Name

`AnalogModulation.PM.NominalVoltage`

C Constant Name

`IVIUPCONVERTER_ATTR_PM_NOMINAL_VOLTAGE`

Description

Returns the voltage at which the instrument achieves the amount of modulation specified by the PM Deviation attribute. The units are Volts.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

9.2.5.PM Source

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	Configure PM

.NET Property Name

`AnalogModulation.PM.Source`

COM Property Name

`AnalogModulation.PM.Source`

C Constant Name

`IVIUPCONVERTER_ATTR_PM_SOURCE`

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources are summed. Multiple source names are separated by commas.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

9.3.IviUpconverterModulatePM Functions

The IviUpconverterModulatePM extension group defines the following function:

- Configure PM
- Configure PM Enabled (IVI-C Only)
- Configure PM External Coupling (IVI-C Only)

This section describes the behavior and requirements of this function.

9.3.1.Configure PM

Description

This function configures the attributes that control the upconverter's phase modulation.

.NET Method Prototype

```
void AnalogModulation.PM.Configure (String source,
                                    Double deviation);
```

COM Method Prototype

```
HRESULT AnalogModulation.PM.Configure ([in] BSTR Source,
                                       [in] double Deviation);
```

C Prototype

```
ViStatus IviUpconverter_ConfigurePM (ViSession Vi,
                                     ViConstString Source,
                                     ViReal64 Deviation);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Source`	Specifies the source of the signal the upconverter uses to modulate the output signal. The driver uses this value to set the PM Source attribute. See the attribute description for more details.	`ViConstString`
`Deviation`	Specifies the extent of modulation. The driver uses this value to set the PM Deviation attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

9.3.2.Configure PM Enabled (IVI-C Only)

Description

Configures the upconverter to apply phase modulation to the RF output signal.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.PM.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.PM.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigurePMEnabled (ViSession Vi,
                                             ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Specifies whether the upconverter enables or disables phase modulation of the RF output signal. The driver uses this value to set the PM Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

9.3.3.Configure PM External Coupling (IVI-C Only)

Description

Configures the coupling of the external source for phase modulation.

.NET Method Prototype

```
N/A
```

(use the `AnalogModulation.PM.ExternalCoupling` property)

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.PM.ExternalCoupling` property)

C Prototype

```
ViStatus IviUpconverter_ConfigurePMExternalCoupling (ViSession Vi,
                                        ViInt32 Coupling);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Coupling`	Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the PM External Coupling attribute. See the attribute description for more details.	`ViInt32`

Defined Values for the Coupling Parameter

Name	*Description*		
		Language	*Identifier*
AC	The external source is coupled for AC only.		
		.NET	`ExternalCoupling.AC`
		C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_AC`
		COM	`IviUpconverterPMExternalCouplingAC`
DC	The external source is coupled for both DC and AC		
		.NET	`ExternalCoupling.DC`
		C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_DC`
		COM	`IviUpconverterPMExternalCouplingDC`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

9.4.IviUpconverterModulatePM Behavior Model

The IviUpconverterModulatePM Extension Group follows the behavior model of the IviUpconverterBase capability group.

9.5.IviUpconverterModulatePM Compliance Notes

1. If a specific driver implements the IviUpconverterModulatePM Extension Group, it shall also implement the IviUpconverterAnalogModulationSource Extension Group.

2. If a specific driver does not support an external modulation source, it shall not support the PM External Coupling attribute or the Configure PM External Coupling function.

10.IviUpconverterAnalogModulationSource Extension Group

10.1.IviUpconverterAnalogModulationSource Overview

The source of the modulating signal is a repeated capability. One or more internal sources and external sources can be combined. The voltage of all signals is summed before modulating the RF output signal. This applies to the following extension groups:

- IviUpconverterModulateAM
- IviUpconverterModulateFM
- IviUpconverterModulatePM

10.2.IviUpconverterAnalogModulationSource Attributes

The IviUpconverterAnalogModulationSource extension group defines the following attributes:

- Analog Modulation Source Count
- Analog Modulation Source Name (IVI.COM Only)

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

10.2.1.Analog Modulation Source Count

Data Type	Access	Applies To	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

.NET Property Name

```
AnalogModulation.Source.Count
```

COM Property Name

```
AnalogModulation.Source.Count
```

C Constant Name

```
IVIUPCONVERTER_ATTR_ANALOG_MODULATION_SOURCE_COUNT
```

Description

Returns the number of Analog Modulation Sources available on the device.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

10.2.2.Analog Modulation Source Name (IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	RO	AnalogModulationSource	None	N/A

.NET Constant Name

N/A

(Use the `AnalogModulation.Source.GetName()` method.

COM Property Name

```
HRESULT AnalogModulation.Source.Name ([in] LONG Index,
                                      [out,retval] BSTR* Name);
```

C Constant Name

N/A

(Use the `IviUpconverter_GetAnalogModulationSourceName()` method.

Description

Returns the Analog Modulation Source identifier that corresponds to the one-based index that the user specifies. If the driver defines a qualified Analog Modulation Source name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the Analog Modulation Source Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

10.3. IviUpconverterAnalogModulationSource Functions

The IviUpconverterAnalogModulationSource extension group defines the following function:

- Get Analog Modulation Source Name (IVI-C & IVI.NET Only)

This section describes the behavior and requirements of this function.

10.3.1.Get Analog Modulation Source Name (IVI-C & IVI.NET Only)

Description

This function returns the specific driver defined Analog Modulation Source name that corresponds to the index that the user specifies. If the driver defines a qualified Analog Modulation Source name, this function returns the qualified name. . If the value that the user passes for the `Index` parameter is less than one or greater than the value of the Analog Modulation Source Count, the function returns an empty string in the `Name` parameter and returns the Invalid Value error.

In C, the index is one-based. In .NET, the index is zero-based.

.NET Method Prototype

```
String AnalogModulation.Source.GetName (Int32 index);
```

COM Method Prototype

```
N/A
```

(use the `AnalogModulation.Source.Name` property)

C Prototype

```
ViStatus IviUpconverter_GetAnalogModulationSourceName (ViSession Vi,
                                        ViInt32 Index,
                                        ViInt32 NameBufferSize,
                                        ViChar Name[]);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Index`	An index (one-based for IVI-C, zero-based for IVI.NET) that defines which name to return.	`ViInt32`
`NameBufferSize`	The number of bytes in the ViChar array that the user specifies for the `Name` parameter.	`ViInt32`

Outputs	Description	Base Type
`Name`	A user-allocated (for IVI-C) or driver-allocated (for IVI-COM) buffer into which the driver stores the Analog Modulation Source name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

10.4.IviUpconverterAnalogModulationSource Behavior Model

The IviUpconverterAnalogModulationSource extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model*.

10.5.IviUpconverterAnalogModulationSource Compliance Notes

If the driver supports this extension group it shall support the IviUpconverterGenBase capabilities and at least one of the following extension groups:

- IviUpconverterModulateAM
- IviUpconverterModulateFM
- IviUpconverterModulatePM

11.IviUpconverterModulatePulse Extension Group

11.1.IviUpconverterModulatePulse Overview

The IviUpconverterModulatePulse Extension Group supports upconverters that can apply pulse modulation to the RF output signal. The user can enable or disable pulse modulation, and specify the source and the polarity of the modulating signal.

11.2.IviUpconverterModulatePulse Attributes

The IviUpconverterModulatePulse extension group defines the following attributes:

- Pulse Modulation Enabled
- Pulse Modulation External Polarity

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

11.2.1.Pulse Modulation Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure Pulse Modulation Enabled (IVI-C Only)

.NET Property Name

`PulseModulation.Enabled`

COM Property Name

`PulseModulation.Enabled`

C Constant Name

`IVIUPCONVERTER_ATTR_PULSE_MODULATION_ENABLED`

Description

Enables or disables pulse modulation of the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

11.2.2.Pulse Modulation External Polarity

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure Pulse Modulation External Polarity (IVI-C Only)

.NET Property Name

`PulseModulation.ExternalPolarity`

.NET Enumeration Name

`PulseModulationExternalPolarity`

COM Property Name

`PulseModulation.ExternalPolarity`

COM Enumeration Name

`IviUpconverterPulseModulationExternalPolarityEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY`

Description

Specifies the polarity of the external source signal.

Defined Values

Name	***Description***		
		Language	***Identifier***
Normal	The upconverter modulates the carrier signal with normal pulse polarity. Increasing the positive pulse voltage level results in higher RF level.		
		.NET	`PulseModulationExternalPolarity.Normal`
		C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL`
		COM	`IviUpconverterPulseModulationExternalPolarityNormal`
Inverse	The upconverter modulates the carrier signal with inverted pulse polarity. Increasing the positive pulse voltage level results in lower RF level.		
		.NET	`PulseModulationExternalPolarity.Inverse`
		C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE`
		COM	`IviUpconverterPulseModulationExternalPolarityInverse`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE` and less than `IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE.`

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE`.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse Modulation External Polarity Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM Scaling Specific Ext Base, `IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE` and `IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE`.

11.3.IviUpconverterModulatePulse Functions

The IviUpconverterModulatePulse extension group defines the following function:

- Configure Pulse Modulation Enabled (IVI-C Only)
- Configure Pulse Modulation External Polarity (IVI-C Only)

This section describes the behavior and requirements of this function.

11.3.1.Configure Pulse Modulation Enabled (IVI-C Only)

Description

Configures the upconverter to apply pulse modulation to the RF output signal.

.NET Method Prototype

```
N/A
```

(use the `PulseModulation.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `PulseModulation.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigurePulseModulationEnabled (ViSession Vi,
                                              ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Enables or disables pulse modulation of the RF output signal. The driver uses this value to set the Pulse Modulation Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

11.3.2.Configure Pulse Modulation External Polarity (IVI-C Only)

Description

Specifies the polarity of the external source signal.

.NET Method Prototype

```
N/A
```

(use the `PulseModulation.ExternalPolarity` property)

COM Method Prototype

```
N/A
```

(use the `PulseModulation.ExternalPolarity` property)

C Prototype

```
ViStatus IviUpconverter_ConfigurePulseModulationExternalPolarity (ViSession Vi,
                                                        ViInt32 Polarity);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Polarity`	Specifies the polarity of the external source signal. The driver uses this value to set the Pulse Modulation External Polarity attribute. See the attribute description for more details.	`ViInt32`

Defined Values for the Polarity Parameter

Name	*Description*		
		Language	*Identifier*
Normal	The upconverter modulates the carrier signal with normal pulse polarity. Increasing the positive pulse voltage level results in higher RF level.		
		.NET	`PulseModulationExternalPolarity.Normal`
		C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL`
		COM	`IviUpconverterPulseModulationExternalPolarityNormal`
Inverse	The upconverter modulates the carrier signal with inverted pulse polarity. Increasing the positive pulse voltage level results in lower RF level.		
		.NET	`PulseModulationExternalPolarity.Inverse`
		C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE`
		COM	`IviUpconverterPulseModulationExternalPolarityInverse`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

11.4.IviUpconverterModulatePulse Behavior Model

The IviUpconverterModulatePulse Extension Group follows the behavior model of the IviUpconverterBase capability group.

11.5.IviUpconverterModulatePulse Compliance Notes

For a specific driver to comply with the IviUpconverterModulatePulse extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

12.IviUpconverterBypass Extension Group

12.1.IviUpconverterBypass Overview

The IviUpconverterBypass extension group supports upconverters with the ability to route the IF input completely around the upconverter. It defines a single attribute and an associated configure function.

12.2.IviUpconverterBypass Attributes

The IviUpconverterBypass extension group defines the following attributes:

- Bypass

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions*.

12.2.1.Bypass

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	IFInput	None	Configure Bypass (IVI-C Only)

.NET Property Name

```
IFInput.Bypass
```

COM Property Name

```
IFInput.Bypass
```

C Constant Name

```
IVIUPCONVERTER_ATTR_BYPASS
```

Description

Specifies whether or not the IF input signal bypasses the entire upconverter. When set to True, the IF input signal is routed directly to the RF output indicated by the value of the Active RF Output attribute. When set to False, the IF input signal is routed into the front end of the upconverter and follows the normal signal path, as dictated by other downstream path control attributes, such as Preselector Enabled.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

12.3.IviUpconverterBypass Functions

The IviUpconverterBypass extension group defines the following function:

- Configure Bypass (IVI-C Only)

This section describes the behavior and requirements of this function.

12.3.1.Configure Bypass (IVI-C Only)

Description

Configures whether or not the IF input signal completely bypasses the upconverter.

.NET Method Prototype

```
N/A
```

(use the `IFInput.Bypass` property)

COM Method Prototype

```
N/A
```

(use the `IFInput.Bypass` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureBypass (ViSession Vi,
                                        ViBoolean Bypass);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Bypass`	Specifies whether or not the IF input signal bypasses the upconverter. The driver uses this value to set the Bypass attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

12.4.IviUpconverterBypass Behavior Model

The IviUpconverterBypass extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model.*

12.5.IviUpconverterBypass Compliance Notes

For a specific driver to comply with the IviUpconverterBypass extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

13.IviUpconverterOutputReadyTrigger Extension Group

13.1.IviUpconverterOutputReadyTrigger Overview

The IviUpconverterOutputReadyTrigger extension group supports upconverters with the ability to send an output trigger signal when the RF output has settled to a point where it is suitable for processing by downstream components.

13.2.IviUpconverterOutputReadyTrigger Attributes

The IviUpconverterOutputReadyTrigger extension group defines the following attributes:

- RF Output Ready Trigger

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

13.2.1.RF Output Ready Trigger

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	RFOutput	None	Configure RF Output Ready Trigger (IVI-C Only)

.NET Property Name

`RFOutput.ReadyTrigger`

COM Property Name

`RFOutput.ReadyTrigger`

C Constant Name

`IVIUPCONVERTER_ATTR_RF_OUTPUT_READY_TRIGGER`

Description

Specifies where a trigger should be fired when the active RF output has settled to a point that is suitable for processing by downstream components.

Defined Values

The defined values for this attribute are given in *Chapter 30: IviUpconverter Attribute Value Defintions.*

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.Compliance Notes

1. The specific driver need not support all of the trigger source values defined in *Chapter 30: IviUpconverter Attribute Value Defintions.*

13.3.IviUpconverterOutputReadyTrigger Functions

The IviUpconverterOutputReadyTrigger extension group defines the following function:

- Configure RF Output Ready Trigger (IVI-C Only)

This section describes the behavior and requirements of this function.

13.3.1.Configure RF Output Ready Trigger (IVI-C Only)

Description

This function configures the output trigger used to indicate when the active RF output has settled to a point suitable for processing by downstream components.

.NET Method Prototype

```
N/A
```

(use the `RFOuptut.ReadyTrigger` property)

COM Method Prototype

```
N/A
```

(use the `RFOuptut.ReadyTrigger` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureRFOutputReadyTrigger (ViSession Vi,
                                    ViConstString OutputTrigger);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`OutputTrigger`	Specifies where a trigger should be fired when the active RF output has settled to a point that is suitable for processing by downstream components. The driver uses this value to set the RF Output Ready Trigger attribute. See the attribute description for more details.	`ViConstString`

Defined Values for the OutputTrigger Parameter

The defined values for this parameter are listed with the RF Output Ready Trigger attribute.

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

13.4.IviUpconverterOutputReadyTrigger Behavior Model

The IviUpconverterOutputReadyTrigger extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model*.

13.5.IviUpconverterOutputReadyTrigger Compliance Notes

For a specific driver to comply with the IviUpconverterOutputReadyTrigger extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

14.IviUpconverterSweep Extension Group

14.1.IviUpconverterSweep Overview

The IviUpconverterSweep extension group supports upconverters with the ability to sweep (or step) the frequency, power, or gain of the RF output signal.

14.2.IviUpconverterSweep Attributes

The IviUpconverterSweep extension group defines the following attributes:

- Is Sweeping
- Sweep Mode
- Sweep Trigger Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

14.2.1.Is Sweeping

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	RO	N/A	None	N/A

.NET Property Name

`Sweep.IsSweeping`

COM Property Name

`Sweep.IsSweeping`

C Constant Name

`IVIUPCONVERTER_ATTR_IS_SWEEPING`

Description

Indicates if the upconverter is currently sweeping the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

14.2.2.Sweep Mode

Data Type	Access	Applies To	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Sweep

.NET Property Name

Sweep.Mode

.NET Enumeration Name

SweepMode

COM Property Name

Sweep.Mode

COM Enumeration Name

IviUpconverterSweepModeEnum

C Constant Name

IVIUPCONVERTER_ATTR_SWEEP_MODE

Description

Specifies whether the RF output signal is a continuous wave or the frequency, power level, or gain is swept or stepped.

Defined Values

Name	*Description*		
		Language	*Identifier*
None	The RF output of the upconverter is a non-swept signal (Continuous Wave). Frequency and power level settings from the base capability group are used.		
		.NET	`SweepMode.None`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_NONE`
		COM	`IviUpconverterSweepModeNone`
Frequency Sweep	The upconverter sweeps the RF output signal's frequency in an analog form (non-stepped). Refer to IviUpconverterFrequencySweep extension group.		
		.NET	`SweepMode.FrequencySweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_SWEEP`
		COM	`IviUpconverterSweepModeFrequencySweep`
Power Sweep	The upconverter sweeps the RF output signal's power in an analog form (non-stepped). Refer to IviUpconverterPowerSweep extension group.		
		.NET	`SweepMode.PowerSweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_SWEEP`
		COM	`IviUpconverterSweepModePowerSweep`
Gain Sweep	The upconverter sweeps the upconverter’s gain in an analog form (non-stepped). Refer to IviUpconverterGainSweep extension group.		
		.NET	`SweepMode.GainSweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_SWEEP`
		COM	`IviUpconverterSweepModeGainSweep`
Frequency Step	The upconverter sweeps the RF output signal's frequency in steps. Refer to IviUpconverterFrequencyStep extension group.		
		.NET	`SweepMode.FrequencyStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_STEP`
		COM	`IviUpconverterSweepModeFrequencyStep`
Power Step	The upconverter sweeps the RF output signal's power level in steps. Refer to IviUpconverterPowerStep extension group.		
		.NET	`SweepMode.PowerStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_STEP`
		COM	`IviUpconverterSweepModePowerStep`
Gain Step	The upconverter sweeps the upconverter’s gain in steps. Refer to IviUpconverterGainStep extension group.		
		.NET	`SweepMode.GainStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_STEP`

		COM	`IviUpconverterSweepModeGainStep`
List	The upconverter uses two lists with frequency and power level or gain values to sweep the RF output signal. Refer to IviUpconverterList extension group.		
		.NET	`SweepMode.List`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_LIST`
		COM	`IviUpconverterSweepModeList`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_SWEEP_MODE_CLASS_EXT_BASE` and less than `IVIUPCONVERTER_VAL_SWEEP_MODE_SPECIFIC_EXT_BASE`.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIUPCONVERTER_VAL_SWEEP_MODE_SPECIFIC_EXT_BASE`.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Sweep Mode Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Sweep Mode Specific Ext Base, `IVIUPCONVERTER_VAL_SWEEP_MODE_SPECIFIC_EXT_BASE` and `IVIUPCONVERTER_VAL_SWEEP_MODE_CLASS_EXT_BASE`.

14.2.3.Sweep Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	Configure Sweep

.NET Property Name

`Sweep.TriggerSource`

COM Property Name

`Sweep.TriggerSource`

C Constant Name

`IVIUPCONVERTER_ATTR_SWEEP_TRIGGER_SOURCE`

Description

Specifies the way to start the sweep or run the sweep continuously.

Defined Values

The defined values for this attribute are given in *Chapter 30: IviUpconverter Attribute Value Defintions.*

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in *Chapter 30: IviUpconverter Attribute Value Defintions.*

14.3.IviUpconverterSweep Functions

The IviUpconverterSweep extension group defines the following function:

- Configure Sweep

This section describes the behavior and requirements of this function.

14.3.1.Configure Sweep

Description

This function configures the upconverter whether the RF output signal is a continuous wave or the frequency, power level, or gain is swept or stepped.

COM Method Prototype

```
void Sweep.Configure (SweepMode mode,
                      String triggerSource);
```

COM Method Prototype

```
HRESULT Sweep.Configure ([in] IviUpconverterSweepModeEnum Mode,
                         [in] BSTR TriggerSource);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureSweep (ViSession Vi,
                                        ViInt32 Mode,
                                        ViConstString TriggerSource);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Mode`	Specifies the sweep mode of the Upconverter. The driver uses this value to set the Sweep Mode attribute. See the attribute description for more details.	`ViInt32`
`TriggerSource`	Specifies the way to start the sweep or running it continuously. The driver uses this value to set the Sweep Trigger Source attribute. See the attribute description for more details.	`ViConstString`

Defined Values for the Mode Parameter

Name	***Description***		
		Language	***Identifier***
None	The RF output of the upconverter is a non-swept signal (Continuous Wave). Frequency and power level settings from the base capability group are used.		
		.NET	`SweepMode.None`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_NONE`
		COM	`IviUpconverterSweepModeNone`
Frequency Sweep	The upconverter sweeps the RF output signal's frequency in an analog form (non-stepped). Refer to IviUpconverterFrequencySweep extension group.		
		.NET	`SweepMode.FrequencySweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_SWEEP`
		COM	`IviUpconverterSweepModeFrequencySweep`
Power Sweep	The upconverter sweeps the RF output signal's power level in an analog form (non-stepped). Refer to the IviUpconverterPowerSweep extension group.		
		.NET	`SweepMode.PowerSweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_SWEEP`
		COM	`IviUpconverterSweepModePowerSweep`
Gain Sweep	The upconverter sweeps the upconverter gain in an analog form (non-stepped). Refer to IviUpconverterGainSweep extension group.		
		.NET	`SweepMode.GainSweep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_SWEEP`
		COM	`IviUpconverterSweepModeGainSweep`
Frequency Step	The upconverter sweeps the RF output signal's frequency in steps. Refer to IviUpconverterFrequencyStep extension group.		
		.NET	`SweepMode.FrequencyStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_STEP`
		COM	`IviUpconverterSweepModeFrequencyStep`
Power Step	The upconverter sweeps the RF output signal's power level in steps. Refer to IviUpconverterPowerStep extension group.		
		.NET	`SweepMode.PowerStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_STEP`
		COM	`IviUpconverterSweepModePowerStep`
Gain Step	The upconverter sweeps the upconverter gain in steps. Refer to IviUpconverterGainStep extension group.		
		.NET	`SweepMode.GainStep`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_STEP`

		COM	`IviUpconverterSweepModeGainStep`
List	The upconverter uses two lists with frequency and power level or gain values to sweep the RF output signal. Refer to IviUpconverterList extension group.		
		.NET	`SweepMode.List`
		C	`IVIUPCONVERTER_VAL_SWEEP_MODE_LIST`
		COM	`IviUpconverterSweepModeList`

Defined Values for the TriggerSource Parameter

The defined values for this parameter are listed with the Sweep Trigger Source attribute.

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

14.4. IviUpconverterSweep Behavior Model

The following behavior models show the relationship between the IviUpconverterSweep extension group and upconverter behavior.

14.4.1. Frequency Sweep / Power Sweep/Gain Sweep

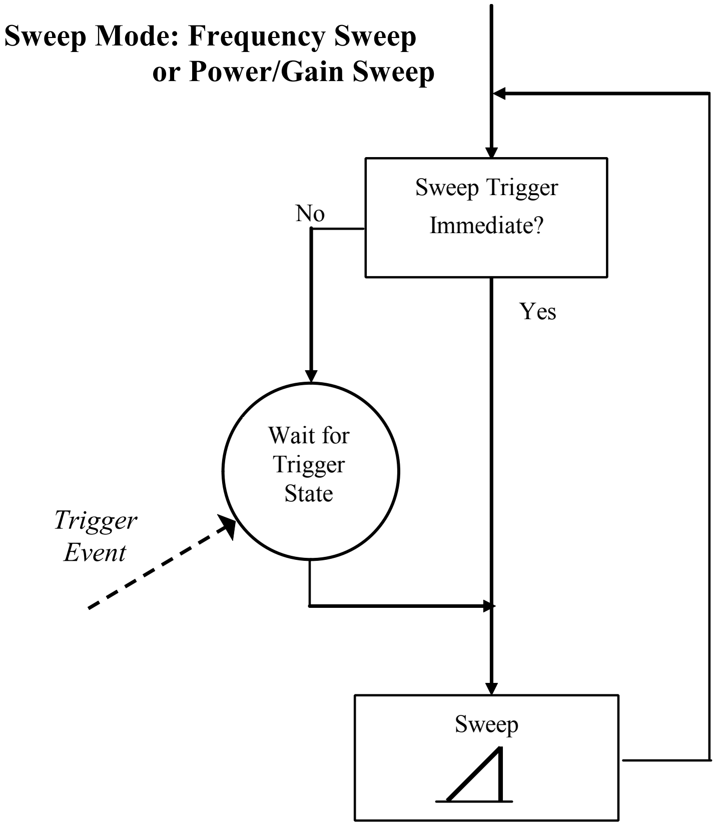


Figure 141: IviUpconverterSweep Behavior Model for Frequency and Power Sweeps

Setting the Sweep Trigger Source attribute to “Immediate” will continuously generate sweeps (either frequency, power, or gain sweeps). The duration of one sweep from start to stop is defined with Frequency Sweep Time, Power Sweep Time, or Gain Sweep Time. Setting the Sweep Trigger Source attribute to “External” or “Software” will delay the start of the next sweep until the specified trigger event occurs.

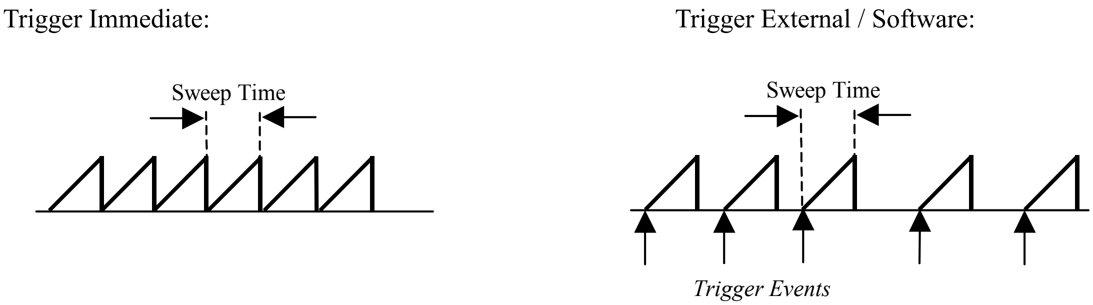


Figure 142: IviUpconverterSweep Trigger Diagrams Frequency, Power, and Gain Sweep Modes

14.4.2.Frequency Step / Power Step / Gain Step / List

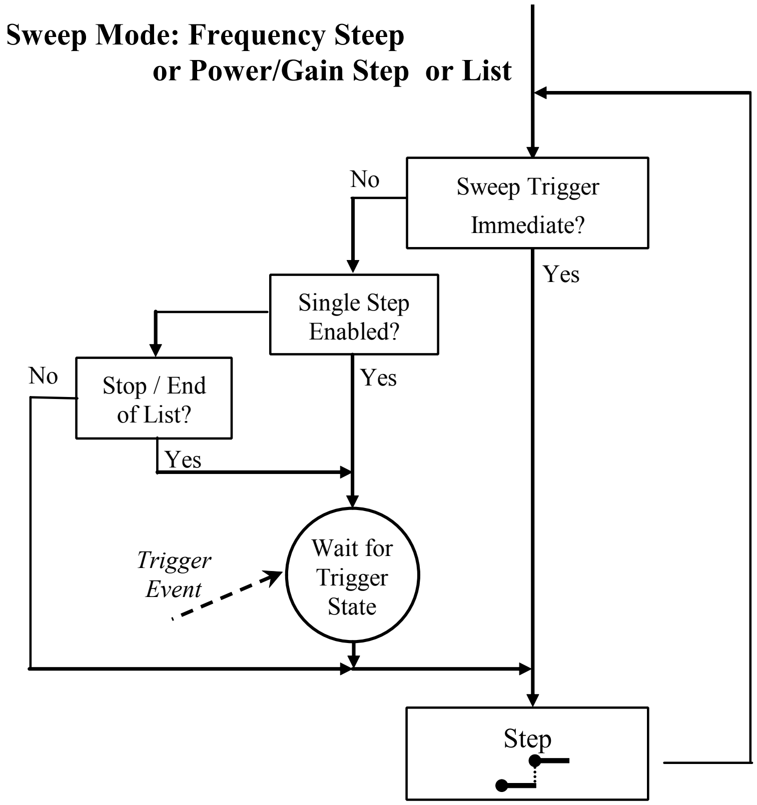
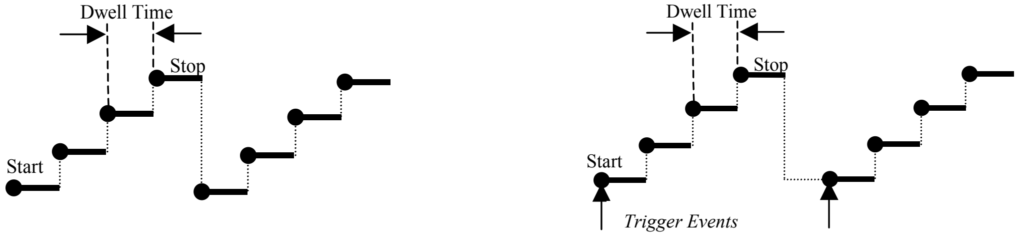


Figure 143: IviUpconverterSweep Behavior Model for Frequency, Power, and Gain Step and List Modes

Setting the Sweep Trigger Source attribute to "Immediate" will continuously generate steps (frequency, power, or gain) either from start to stop with fixed increments (steps) or from a list of arbitrary values. The duration of one step is defined with Dwell Time. Setting the Sweep Trigger Source attribute to “External” or “Software” will delay the start of the next step until the specified trigger event occurs.

Trigger Immediate:

Trigger External / Software:

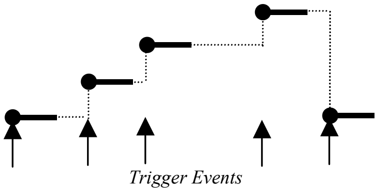


Single Step and Trigger External / Software:

Figure 144: IviUpconverterSweep Trigger Diagrams for Frequency, Power, and Gain Step and List Modes

14.5.IviUpconverterSweep Compliance Notes

For a specific driver to comply with the IviUpconverterSweep extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.



In addition the following rules apply:

1. When a call to Configure RF of the base capability group is made, the Sweep Mode attribute is set to None, hence stopping any sweeps that may have been in progress.
2. When the Sweep Mode attribute is set to Frequency Sweep, Frequency Step or List Mode with frequency or frequency and power list selected, setting Frequency of the base capability group will set the Sweep Mode attribute to None. Setting the value of Power Level or Gain will control the level of the swept signal.
3. When the Sweep Mode attribute is set to Power Sweep, Power Step, Gain Sweep, Gain Step, or List Mode with power, gain, or frequency and power level or frequency and gain list selected, setting Power Level or Gain of the base capability group will set the Sweep Mode attribute to None. Setting the value of Frequency will control the frequency of the swept signal.
4. When the Sweep Mode attribute is set to List and a frequency and power level or gain list is selected, setting Frequency, Power Level, or Gain of the base capability group will set the Sweep Mode attribute to None.
5. When the Sweep Mode attribute is changed from any sweep mode to None, the upconverter will generate the signal accordingly to the previously set values of Frequency and Power Level or Gain of the base capability group.

15.IviUpconverterFrequencySweep Extension Group

15.1.IviUpconverterFrequencySweep Overview

The IviUpconverterFrequencySweep Extension Group supports upconverters that can apply a frequency sweep to the RF output signal. The user may configure the sweep with start and stop, or center and span frequencies. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterSweepFrequencySweep extension group is active when the Sweep Mode attribute is set to Frequency Sweep.

15.2.IviUpconverterFrequencySweep Attributes

The IviUpconverterFrequencySweep extension group defines the following attributes:

- Frequency Sweep Start
- Frequency Sweep Stop
- Frequency Sweep Time

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

15.2.1.Frequency Sweep Start

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Frequency Sweep Start Stop, Configure Frequency Sweep Center Span

.NET Property Name

```
Sweep.FrequencySweep.Start
```

COM Property Name

```
Sweep.FrequencySweep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_START
```

Description

Specifies the start frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

15.2.2.Frequency Sweep Stop

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Frequency Sweep Start Stop, Configure Frequency Sweep Center Span

.NET Property Name

```
Sweep.FrequencySweep.Stop
```

COM Property Name

```
Sweep.FrequencySweep.Stop
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_STOP
```

Description

Specifies the stop frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

15.2.3.Frequency Sweep Time

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure Frequency Sweep Time (IVI-C Only)

.NET Property Name

```
Sweep.FrequencySweep.Time
```

COM Property Name

```
Sweep.FrequencySweep.Time
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_TIME
```

Description

Specifies the duration of one sweep from start to stop frequency. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

15.3.IviUpconverterFrequencySweep Functions

The IviUpconverterFrequencySweep extension group defines the following function:

- Configure Frequency Sweep Center Span
- Configure Frequency Sweep Start Stop
- Configure Frequency Sweep Time (IVI-C Only)

This section describes the behavior and requirements of this function.

15.3.1.Configure Frequency Sweep Center Span

Description

This function configures the center frequency and the frequency span for the sweep.

.NET Method Prototype

```
void Sweep.FrequencySweep.ConfigureCenterSpan (Double center,
                                               Double span);
```

COM Method Prototype

```
HRESULT Sweep.FrequencySweep.ConfigureCenterSpan ([in] double Center,
                                        [in] double Span);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureFrequencySweepCenterSpan (ViSession Vi,
                                        ViReal64 Center,
                                        ViReal64 Span);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Center`	Specifies the center frequency of the sweep. This value, along with the Span parameter, sets the Frequency Sweep Start and Frequency Sweep Stop attributes.	`ViReal64`
`Span`	Specifies the frequency span of the sweep. This value, along with the Center parameter, sets the Frequency Sweep Start and Frequency Sweep Stop attributes.	`ViReal64`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

15.3.2. Configure Frequency Sweep Start Stop

Description

This function configures the start and the stop frequency for the sweep.

.NET Method Prototype

```
void Sweep.FrequencySweep.ConfigureStartStop (Double start,
                                              Double stop);
```

COM Method Prototype

```
HRESULT Sweep.FrequencySweep.ConfigureStartStop ([in] double Start,
                                       [in] double Stop);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureFrequencySweepStartStop (ViSession Vi,
                                       ViReal64 Start,
                                       ViReal64 Stop);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Start`	Specifies the start frequency of the sweep. The driver uses this value to set the Frequency Sweep Start attribute. See the attribute description for more details.	`ViReal64`
`Stop`	Specifies the stop frequency of the sweep. The driver uses this value to set the Frequency Sweep Stop attribute. See the attribute description for more details.	`ViReal64`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

15.3.3.Configure Frequency Sweep Time (IVI-C Only)

Description

Configures the duration of one frequency sweep.

.NET Method Prototype

```
N/A
```

(use the `Sweep.FrequencySweep.Time` property)

COM Method Prototype

```
N/A
```

(use the `Sweep.FrequencySweep.Time` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureFrequencySweepTime (ViSession Vi,
                                    ViReal64 SweepTime);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Frequency Sweep Time`	Specifies the duration of one sweep from start to stop frequency. The driver uses this value to set the Frequency Sweep Time attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

15.4.IviUpconverterFrequencySweep Behavior Model

The IviUpconverterFrequencySweep extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

15.5.IviUpconverterFrequencySweep Compliance Notes

1. If a specific driver implements the IviUpconverterFrequencySweep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Frequency Sweep for the Sweep Mode attribute.

16.IviUpconverterPowerSweep Extension Group

16.1.IviUpconverterPowerSweep Overview

The IvUpconverterPowerSweep Extension Group supports upconverters that can apply a power sweep to the output signal. The user may configure the sweep with start and stop power. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterPowerSweep extension group is active when the Sweep Mode attribute is set to Power Sweep.

16.2.IviUpconverterPowerSweep Attributes

The IviUpconverterPowerSweep extension group defines the following attributes:

- Power Sweep Start
- Power Sweep Stop
- Power Sweep Time

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

16.2.1.Power Sweep Start

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Power Sweep Start Stop

.NET Property Name

```
Sweep.PowerSweep.Start
```

COM Property Name

```
Sweep.PowerSweep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_SWEEP_START
```

Description

Specifies the start power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

16.2.2.Power Sweep Stop

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Power Sweep Start Stop

.NET Property Name

```
Sweep.PowerSweep.Stop
```

COM Property Name

```
Sweep.PowerSweep.Stop
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_SWEEP_STOP
```

Description

Specifies the stop power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

16.2.3.Power Sweep Time

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure Power Sweep Time (IVI-C Only)

.NET Property Name

```
Sweep.PowerSweep.Time
```

COM Property Name

```
Sweep.PowerSweep.Time
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_SWEEP_TIME
```

Description

Specifies the duration of one sweep from start to stop power. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

16.3.IviUpconverterPowerSweep Functions

The IviUpconverterPowerSweep extension group defines the following function:

- Configure Power Sweep Start Stop
- Configure Power Sweep Time (IVI-C Only)

This section describes the behavior and requirements of this function.

16.3.1.Configure Power Sweep Start Stop

Description

This function configures the start and the stop power for the sweep.

.NET Method Prototype

```
void Sweep.PowerSweep.ConfigureStartStop (Double start,
                                          Double stop);
```

COM Method Prototype

```
HRESULT Sweep.PowerSweep.ConfigureStartStop ([in] double Start,
                                            [in] double Stop);
```

C Prototype

```
ViStatus IviUpconverter_ConfigurePowerSweepStartStop (ViSession Vi,
                                        ViReal64 Start,
                                        ViReal64 Stop);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Start`	Specifies the start power of the sweep. The driver uses this value to set the Power Sweep Start attribute. See the attribute description for more details.	`ViReal64`
`Stop`	Specifies the stop power of the sweep. The driver uses this value to set the Power Sweep Stop attribute. See the attribute description for more details.	`ViReal64`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

16.3.2.Configure Power Sweep Time (IVI-C Only)

Description

Configures the duration of one power sweep.

.NET Method Prototype

```
N/A
```

(use the `Sweep.PowerSweep.Time` property)

COM Method Prototype

```
N/A
```

(use the `Sweep.PowerSweep.Time` property)

C Prototype

```
ViStatus IviUpconverter_ConfigurePowerSweepTime (ViSession Vi,
                                                 ViReal64 SweepTime);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Power Sweep Time`	Specifies the duration of one sweep from start to stop power. The driver uses this value to set the Power Sweep Time attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

16.4.IviUpconverterPowerSweep Behavior Model

The IviUpconverterPowerSweep extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

16.5.IviUpconverterPowerSweep Compliance Notes

1. If a specific driver implements the IvUpconverterPowerSweep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Power Sweep for the Sweep Mode attribute.

17.IviUpconverterGainSweep Extension Group

17.1.IviUpconverterGainSweep Overview

The IvUpconverterGainSweep Extension Group supports upconverters that can apply a gain sweep to the output signal. The user may configure the sweep with start and stop gain. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterGainSweep extension group is active when the Sweep Mode attribute is set to Gain Sweep.

17.2.IviUpconverterGainSweep Attributes

The IviUpconverterGainSweep extension group defines the following attributes:

- Gain Sweep Start
- Gain Sweep Stop
- Gain Sweep Time

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

17.2.1.Gain Sweep Start

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Gain Sweep Start Stop

.NET Property Name

```
Sweep.GainSweep.Start
```

COM Property Name

```
Sweep.GainSweep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_GAIN_SWEEP_START
```

Description

Specifies the start gain of the sweep. If the stop gain is less than the start gain, the power decreases in value during the sweep. The units are dB.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

17.2.2.Gain Sweep Stop

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Gain Sweep Start Stop

.NET Property Name

```
Sweep.GainSweep.Stop
```

COM Property Name

```
Sweep.GainSweep.Stop
```

C Constant Name

```
IVIUPCONVERTER_ATTR_GAIN_SWEEP_STOP
```

Description

Specifies the stop gain of the sweep. If the stop gain is less than the start gain, the gain decreases in value during the sweep. The units are dB.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

17.2.3.Gain Sweep Time

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Gain Sweep Time (IVI-C Only)

.NET Property Name

Sweep.GainSweep.Time

COM Property Name

Sweep.GainSweep.Time

C Constant Name

IVIUPCONVERTER_ATTR_GAIN_SWEEP_TIME

Description

Specifies the duration of one sweep from start to stop gain. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

17.3.IviUpconverterGainSweep Functions

The IviUpconverterGainSweep extension group defines the following function:

- Configure Gain Sweep Start Stop
- Configure Gain Sweep Time (IVI-C Only)

This section describes the behavior and requirements of this function.

17.3.1.Configure Gain Sweep Start Stop

Description

This function configures the start and the stop gain for the sweep.

.NET Method Prototype

```
void Sweep.GainSweep.ConfigureStartStop (Double start,
                                         Double stop);
```

COM Method Prototype

```
HRESULT Sweep.GainSweep.ConfigureStartStop ([in] double Start,
                                            [in] double Stop);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureGainSweepStartStop (ViSession Vi,
                                        ViReal64 Start,
                                        ViReal64 Stop);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Start	Specifies the start gain of the sweep. The driver uses this value to set the Gain Sweep Start attribute. See the attribute description for more details.	ViReal64
Stop	Specifies the stop gain of the sweep. The driver uses this value to set the Gain Sweep Stop attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

17.3.2. Configure Gain Sweep Time (IVI-C Only)

Description

Configures the duration of one gain sweep.

.NET Method Prototype

```
N/A
```

(use the `Sweep.GainSweep.Time` property)

COM Method Prototype

```
N/A
```

(use the `Sweep.GainSweep.Time` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureGainSweepTime (ViSession Vi,
                                                ViReal64 SweepTime);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Power Sweep Time`	Specifies the duration of one sweep from start to stop gain. The driver uses this value to set the Gain Sweep Time attribute. See the attribute description for more details.	`ViReal64`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

17.4.IviUpconverterGainSweep Behavior Model

The IviUpconverterGainSweep extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

17.5.IviUpconverterGainSweep Compliance Notes

If a specific driver implements the IviUpconverterGainSweep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Gain Sweep for the Sweep Mode attribute.

18.IviUpconverterFrequencyStep Extension Group

18.1.IviUpconverterFrequencyStep Overview

The IviUpconverterFrequencyStep Extension Group supports upconverters that can sweep the frequency of the RF output signal in steps. The user can specify the start, stop and step frequency and set linear or logarithmic spacing. Setting single step and dwell time are also included.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterFrequencyStep extension group is active when the Sweep Mode attribute is set to Frequency Step.

18.2.IviUpconverterFrequencyStep Attributes

The IviUpconverterFrequencyStep extension group defines the following attributes:

- Frequency Step Dwell
- Frequency Step Scaling
- Frequency Step Single Step Enabled
- Frequency Step Size
- Frequency Step Start
- Frequency Step Stop

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

18.2.1.Frequency Step Dwell

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure Frequency Step Dwell

.NET Property Name

```
Sweep.FrequencyStep.Dwell
```

COM Property Name

```
Sweep.FrequencyStep.Dwell
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_STEP_DWELL
```

Description

Specifies the duration of one step. For C and COM, the units are seconds. For .NET, the units are implicit in the implementation of PrecisionTimeSpan. Dwell time starts immediately after a trigger or next step; no settling time is added. This attribute is ignored if the Frequency Step Single Step Enabled attribute is set to True.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.2.2.Frequency Step Scaling

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure Frequency Step Start Stop

.NET Property Name

`Sweep.FrequencyStep.Scaling`

.NET Enumeration Name

`Scaling`

COM Property Name

`Sweep.FrequencyStep.Scaling`

COM Enumeration Name

`IviUpconverterFrequencyStepScalingEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SCALING`

Description

Specifies the spacing of the steps.

Defined Values

Name	*Description*		
		Language	*Identifier*
Linear	Enables linear scaling.		
		.NET	`Scaling.Linear`
		C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LINEAR`
		COM	`IviUpconverterFrequencyStepScalingLinear`
Logarithmic	Enables logarithmic scaling.		
		.NET	`Scaling.Logarithmic`
		C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC`
		COM	`IviUpconverterFrequencyStepScalingLogarithmic`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Frequency Step Scaling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Frequency Step Scaling Specific Ext Base, IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE.

18.2.3.Frequency Step Single Step Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure Frequency Step Dwell

.NET Property Name

```
Sweep.FrequencyStep.SingleStepEnabled
```

COM Property Name

```
Sweep.FrequencyStep.SingleStepEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED
```

Description

Enables or disables single step mode.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.2.4.Frequency Step Size

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Frequency Step Start Stop

.NET Property Name

```
Sweep.FrequencyStep.Size
```

COM Property Name

```
Sweep.FrequencyStep.Size
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SIZE
```

Description

Specifies the step size. The units are Hertz if Frequency Step Scaling attribute is set to Linear and is unitless (factor) if Frequency Step Scaling attribute is set to Logarithmic.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.2.5.Frequency Step Start

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Frequency Step Start Stop

.NET Property Name

```
Sweep.FrequencyStep.Start
```

COM Property Name

```
Sweep.FrequencyStep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_STEP_START
```

Description

Specifies the start frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.2.6.Frequency Step Stop

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Frequency Step Start Stop

.NET Property Name

```
Sweep.FrequencyStep.Stop
```

COM Property Name

```
Sweep.FrequencyStep.Stop
```

C Constant Name

```
IVIUPCONVERTER_ATTR_FREQUENCY_STEP_STOP
```

Description

Specifies the stop frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.3.IviUpconverterFrequencyStep Functions

The IviUpconverterFrequencyStep extension group defines the following function:

- Configure Frequency Step Dwell
- Configure Frequency Step Start Stop
- Reset Frequency Step

This section describes the behavior and requirements of this function.

18.3.1.Configure Frequency Step Dwell

Description

This function configures the attributes that control the stepping.

.NET Method Prototype

```
void Sweep.FrequencyStep.ConfigureDwell (Boolean singleStepEnabled,
                                         Ivi.Driver.PrecisionTimeSpan dwell);
```

COM Method Prototype

```
HRESULT Sweep.FrequencyStep.ConfigureDwell ([in] VARIANT_BOOL
SingleStepEnabled,
                                        [in] double Dwell);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureFrequencyStepDwell (ViSession Vi,
                                        ViBoolean SingleStepEnabled,
                                        ViReal64 Dwell);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
SingleStepEnabled	Specifies whether the trigger initiates the next step. The driver uses this value to set the Frequency Step Single Step Enabled attribute. See the attribute description for more details.	ViBoolean
Dwell	Specifies the duration of one frequency step. The driver uses this value to set the Frequency Step Dwell attribute. See the attribute description for more details.	ViReal64 (C/COM) PrecisionTimeSpan (.NET)

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

18.3.2.Configure Frequency Step Start Stop

Description

This function configures the attributes that control the step frequencies of the upconverter's RF output signal.

.NET Method Prototype

```
void Sweep.FrequencyStep.ConfigureStartStop (Double start,
                                             Double stop,
                                             Scaling scaling,
                                             double stepSize);
```

COM Method Prototype

```
HRESULT Sweep.FrequencyStep.ConfigureStartStop ([in] double Start,
                                        [in] double Stop,
                                        [in]
IviUpconverterFrequencyStepScalingEnum Scaling,
                                        [in] double StepSize);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureFrequencyStepStartStop (ViSession Vi,
                                        ViReal64 Start,
                                        ViReal64 Stop,
                                        ViInt32 Scaling,
                                        ViReal64 StepSize);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Start`	Specifies the start frequency of the step sweep. The driver uses this value to set the Frequency Step Start attribute. See the attribute description for more details.	`ViReal64`
`Stop`	Specifies the stop frequency of the step sweep. The driver uses this value to set the Frequency Step Stop attribute. See the attribute description for more details.	`ViReal64`
`Scaling`	Specifies the scaling of the step sweep. The driver uses this value to set the Frequency Step Scaling attribute. See the attribute description for more details.	`ViReal64`
`StepSize`	Specifies the size of one step. The driver uses this value to set the Frequency Step Size attribute. See the attribute description for more details.	`ViReal64`

Defined Values for the Scaling Parameter

Name	*Description*		
		Language	*Identifier*
Linear	Enables linear scaling.		
		.NET	`Scaling.Linear`
		C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LINEAR`
		COM	`IviUpconverterFrequencyStepScalingLinear`
Logarithmic	Enables logarithmic scaling.		
		.NET	`Scaling.Logarithmic`
		C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC`
		COM	`IviUpconverterFrequencyStepScalingLogarithmic`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

18.3.3.Reset Frequency Step

Description

Resets the current frequency step to the frequency step start value.

.NET Method Prototype

```
void Sweep.FrequencyStep.Reset ();
```

COM Method Prototype

```
HRESULT Sweep.FrequencyStep.Reset ();
```

C Prototype

```
ViStatus IviUpconverter_ResetFrequencyStep (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

18.4.IviUpconverterFrequencyStep Behavior Model

The IviUpconverterFrequencyStep extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

18.5.IviUpconverterFrequencyStep Compliance Notes

1. If a specific driver implements the IviUpconverterFrequencyStep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Frequency Step for the Sweep Mode attribute.

19.IviUpconverterPowerStep Extension Group

19.1.IviUpconverterPowerStep Overview

The IviUpconverterPowerStep Extension Group supports upconverters that can sweep the power of the RF output signal in steps. The user can enable or disable stepping, specify the start, stop and step power. Setting single step and dwell time are also included.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterPowerStep extension group is active when the Sweep Mode attribute is set to Power Step.

19.2.IviUpconverterPowerStep Attributes

The IviUpconverterPowerStep extension group defines the following attributes:

- Power Step Dwell
- Power Step Single Step Enabled
- Power Step Size
- Power Step Start
- Power Step Stop

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

19.2.1.Power Step Dwell

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure Power Step Dwell

.NET Property Name

```
Sweep.PowerStep.Dwell
```

COM Property Name

```
Sweep.PowerStep.Dwell
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_STEP_DWELL
```

Description

Specifies the duration of one step. For C and COM, the units are seconds. For .NET, the units are implicit in the implementation of PrecisionTimeSpan. This attribute is ignored if Power Step Single Step Enabled is set to True.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.2.Power Step Single Step Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure Power Step Dwell

.NET Property Name

```
Sweep.PowerStep.SingleStepEnabled
```

COM Property Name

```
Sweep.PowerStep.SingleStepEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_STEP_SINGLE_STEP_ENABLED
```

Description

Enables or disables single step mode.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.3.Power Step Size

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Power Step Start Stop

.NET Property Name

```
Sweep.PowerStep.Size
```

COM Property Name

```
Sweep.PowerStep.Size
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_STEP_SIZE
```

Description

Specifies the step size. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.4.Power Step Start

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Power Step Start Stop

.NET Property Name

```
Sweep.PowerStep.Start
```

COM Property Name

```
Sweep.PowerStep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_STEP_START
```

Description

Specifies the start power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.5.Power Step Stop

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Power Step Start Stop

.NET Property Name

```
Sweep.PowerStep.Stop
```

COM Property Name

```
Sweep.PowerStep.Stop
```

C Constant Name

```
IVIUPCONVERTER_ATTR_POWER_STEP_STOP
```

Description

Specifies the stop power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.3.IviUpconverterPowerStep Functions

The IviUpconverterPowerStep extension group defines the following function:

- Configure Power Step Dwell
- Configure Power Step Start Stop
- Reset Power Step

This section describes the behavior and requirements of this function.

19.3.1.Configure Power Step Dwell

Description

This function configures the attributes that control the stepping.

.NET Method Prototype

```
void Sweep.PowerStep.ConfigureDwell (Boolean singleStepEnabled,
                                     Ivi.Driver.PrecisionTimeSpan dwell);
```

COM Method Prototype

```
HRESULT Sweep.PowerStep.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled,
                                        [in] double Dwell);
```

C Prototype

```
ViStatus IviUpconverter_ConfigurePowerStepDwell (ViSession Vi,
                                        ViBoolean SingleStepEnabled,
                                        ViReal64 Dwell);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
SingleStepEnable d	Specifies whether the trigger initiates the next step. The driver uses this value to set the Power Step Single Step Enabled attribute. See the attribute description for more details.	ViBoolean
Dwell	Specifies the duration of one power step. The driver uses this value to set the Power Step Dwell attribute. See the attribute description for more details.	ViReal64 (C/COM) PrecisionimeSpan (.NET)

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

19.3.2.Configure Power Step Start Stop

Description

This function configures the attributes that control the power steps of the upconverter's RF output signal.

.NET Method Prototype

```
void Sweep.PowerStep.ConfigureStartStop (Double start,
                                         Double stop,
                                         Double stepSize);
```

COM Method Prototype

```
HRESULT Sweep.PowerStep.ConfigureStartStop ([in] double Start,
                                            [in] double Stop,
                                            [in] double StepSize);
```

C Prototype

```
ViStatus IviUpconverter_ConfigurePowerStepStartStop (ViSession Vi,
                                        ViReal64 Start,
                                        ViReal64 Stop,
                                        ViReal64 StepSize);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Start	Specifies the start power of the stepping. The driver uses this value to set the Power Step Start attribute. See the attribute description for more details.	ViReal64
Stop	Specifies the stop power of the stepping. The driver uses this value to set the Power Step Stop attribute. See the attribute description for more details.	ViReal64
StepSize	Specifies the size of one step. The driver uses this value to set the Power Step Size attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

19.3.3.Reset Power Step

Description

This function resets the stepping if single step is enabled.

.NET Method Prototype

```
void Sweep.PowerStep.Reset ();
```

COM Method Prototype

```
HRESULT Sweep.PowerStep.Reset ();
```

C Prototype

```
ViStatus IviUpconverter_ResetPowerStep (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

19.4.IviUpconverterPowerStep Behavior Model

The IviUpconverterPowerStep extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

19.5.IviUpconverterPowerStep Compliance Notes

1. If a specific driver implements the IviUpconverterPowerStep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Power Step for the Sweep Mode attribute.

20.IviUpconverterGainStep Extension Group

20.1.IviUpconverterGainStep Overview

The IviUpconverterGainStep Extension Group supports upconverters that can sweep the gain of the upconverterin steps. The user can enable or disable stepping, specify the start, stop and step gain. Setting single step and dwell time are also included.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. The IviUpconverterGainStep extension group is active when the Sweep Mode attribute is set to Gain Step.

20.2.IviUpconverterGainStep Attributes

The IviUpconverterGainStep extension group defines the following attributes:

- Gain Step Dwell
- Gain Step Single Step Enabled
- Gain Step Size
- Gain Step Start
- Gain Step Stop

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

20.2.1.Gain Step Dwell

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure Gain Step Dwell

.NET Property Name

`Sweep.GainStep.Dwell`

COM Property Name

`Sweep.GainStep.Dwell`

C Constant Name

`IVIUPCONVERTER_ATTR_GAIN_STEP_DWELL`

Description

Specifies the duration of one step. For C and COM, the units are seconds. For .NET, the units are implicit in the implementation of PrecisionTimeSpan. This attribute is ignored if Gain Step Single Step Enabled is set to True.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

20.2.2.Gain Step Single Step Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure Gain Step Dwell

.NET Property Name

```
Sweep.GainStep.SingleStepEnabled
```

COM Property Name

```
Sweep.GainStep.SingleStepEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_GAIN_STEP_SINGLE_STEP_ENABLED
```

Description

Enables or disables single step mode.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

20.2.3.Gain Step Size

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Gain Step Start Stop

.NET Property Name

```
Sweep.GainStep.Size
```

COM Property Name

```
Sweep.GainStep.Size
```

C Constant Name

```
IVIUPCONVERTER_ATTR_GAIN_STEP_SIZE
```

Description

Specifies the step size. The units are dB.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

20.2.4.Gain Step Start

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Gain Step Start Stop

.NET Property Name

```
Sweep.GainStep.Start
```

COM Property Name

```
Sweep.GainStep.Start
```

C Constant Name

```
IVIUPCONVERTER_ATTR_GAIN_STEP_START
```

Description

Specifies the start gain of the stepped sweep. If the stop gain is less than the start gain, the gain decreases in value during the sweep. The units are dB.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

20.2.5.Gain Step Stop

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Gain Step Start Stop

.NET Property Name

`Sweep.GainStep.Stop`

COM Property Name

`Sweep.GainStep.Stop`

C Constant Name

`IVIUPCONVERTER_ATTR_GAIN_STEP_STOP`

Description

Specifies the stop gain of the stepped sweep. If the stop gain is less than the start gain, the power decreases in value during the sweep. The units are dB.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

20.3.IviUpconverterGainStep Functions

The IviUpconverterPowerStep extension group defines the following function:

- Configure Gain Step Dwell
- Configure Gain Step Start Stop
- Reset Gain Step

This section describes the behavior and requirements of this function.

20.3.1.Configure Gain Step Dwell

Description

This function configures the attributes that control the stepping.

.NET Method Prototype

```
void Sweep.GainStep.ConfigureDwell (Boolean singleStepEnabled,
                                    Ivi.Driver.PrecisionTimeSpan dwell);
```

COM Method Prototype

```
HRESULT Sweep.GainStep.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled,
                                       [in] double Dwell);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureGainStepDwell (ViSession Vi,
                                        ViBoolean SingleStepEnabled,
                                        ViReal64 Dwell);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`SingleStepEnabled`	Specifies whether the trigger initiates the next step. The driver uses this value to set the Gain Step Single Step Enabled attribute. See the attribute description for more details.	`ViBoolean`

`Dwell`	Specifies the duration of one power step. The driver uses this value to set the Gain Step Dwell attribute. See the attribute description for more details.	`ViReal64 (C/COM)`
		`PrecisionimeSpan (.NET)`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

20.3.2.Configure Gain Step Start Stop

Description

This function configures the attributes that control the gain steps of the upconverter sweep.

.NET Method Prototype

```
void Sweep.GainStep.ConfigureStartStop (Double start,
                                        Double stop,
                                        Double stepSize);
```

COM Method Prototype

```
HRESULT Sweep.GainStep.ConfigureStartStop ([in] double Start,
                                           [in] double Stop,
                                           [in] double StepSize);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureGainStepStartStop (ViSession Vi,
                                          ViReal64 Start,
                                          ViReal64 Stop,
                                          ViReal64 StepSize);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Start	Specifies the start power of the stepping. The driver uses this value to set the Gain Step Start attribute. See the attribute description for more details.	ViReal64
Stop	Specifies the stop power of the stepping. The driver uses this value to set the Gain Step Stop attribute. See the attribute description for more details.	ViReal64
StepSize	Specifies the size of one step. The driver uses this value to set the Gain Step Size attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

20.3.3.Reset Gain Step

Description

This function resets the stepping if single step is enabled.

.NET Method Prototype

```
void Sweep.GainStep.Reset ();
```

COM Method Prototype

```
HRESULT Sweep.GainStep.Reset ();
```

C Prototype

```
ViStatus IviUpconverter_ResetGainStep (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

20.4.IviUpconverterGainStep Behavior Model

The IviUpconverterGainStep extension group follows the same behavior model as the IviUpconvertersweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

20.5.IviUpconverterGainStep Compliance Notes

2. If a specific driver implements the IviUpconverterGainStep Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value Gain Step for the Sweep Mode attribute.

21.IviUpconverterList Extension Group

21.1.IviUpconverterList Overview

The IviUpconverterList Extension Group supports upconverters that can set the frequency and power of the RF output signal to values given as a list of values. The user can enable or disable stepping the frequency and power list, specify the name of the list and set its values. The active list can be selected using the list name. Setting single step and dwell time are also included.

In order to support this extension group, a driver shall first support the IviUpconverterSweep Extension Group. This IviUpconverterList extension group is active when the Sweep Mode attribute is set to List.

21.2.IviUpconverterList Attributes

The IviUpconverterList extension group defines the following attributes:

- List Dwell
- List Selected Name
- List Single Step Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

21.2.1.List Dwell

Data Type	Access	Applies To	Coercion	High Level Functions

`ViReal64 (C/COM)`
`PrecisionTimeSpan (.NET)`
	R/W	N/A	None	Configure List Dwell

.NET Property Name

```
Sweep.List.Dwell
```

COM Property Name

```
Sweep.List.Dwell
```

C Constant Name

```
IVIUPCONVERTER_ATTR_LIST_DWELL
```

Description

Specifies the duration of one step. For C and COM, the units are seconds. For .NET, the units are implicit in the implementation of PrecisionTimeSpan. This attribute is ignored if the List Single Step Enabled attribute is set to True.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

21.2.2.List Selected Name

Data Type	Access	Applies To	Coercion	High Level Functions
`ViString`	R/W	N/A	None	N/A

.NET Property Name

`Sweep.List.SelectedName`

COM Property Name

`Sweep.List.SelectedName`

C Constant Name

`IVIUPCONVERTER_ATTR_LIST_SELECTED_NAME`

Description

Specifies the name of the selected list to become active. The name shall be one of the lists created.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

21.2.3.List Single Step Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure List Dwell

.NET Property Name

```
Sweep.List.SingleStepEnabled
```

COM Property Name

```
Sweep.List.SingleStepEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_LIST_SINGLE_STEP_ENABLED
```

Description

Enables or disables single step mode.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

21.3.IviUpconverterList Functions

The IviUpconverterList extension group defines the following function:

- Clear All Lists
- Configure List Dwell
- Create Gain List
- Create Frequency List
- Create Frequency Power List
- Create Frequency Gain List
- Create Power List
- Reset List

This section describes the behavior and requirements of this function.

21.3.1.Clear All Lists

Description

This function deletes all lists from the pool of lists.

.NET Method Prototype

```
void Sweep.List.ClearAll ();
```

COM Method Prototype

```
HRESULT Sweep.List.ClearAll ();
```

C Prototype

```
ViStatus IviUpconverter_ClearAllLists (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.2.Configure List Dwell

Description

This function configures the attributes that control the stepping.

.NET Method Prototype

```
void Sweep.List.ConfigureDwell (Boolean singleStepEnabled,
                                Ivi.Driver.PrecisionTimeSpan dwell);
```

COM Method Prototype

```
HRESULT Sweep.List.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled,
                                   [in] double Dwell);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureListDwell (ViSession Vi,
                                            ViBoolean SingleStepEnabled,
                                            ViReal64 Dwell);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`SingleStepEnabled`	Specifies whether the trigger initiates transitions to the next list step. The driver uses this value to set the List Single Step Enabled attribute. See the attribute description for more details.	`ViBoolean`

`Dwell`	Specifies the duration of one list step. The driver uses this value to set the List Dwell attribute. See the attribute description for more details.	`ViReal64 (C/COM)`
		`PrecisionimeSpan (.NET)`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.3.Create Gain List

Description

This function creates a named list of gain values.

.NET Method Prototype

```
void Sweep.List.CreateGain (String name,
                            Double[] gainList);
```

COM Method Prototype

```
HRESULT Sweep.List.CreateGain ([in] BSTR Name,
                               [in] SAFEARRAY(double)* Gain);
```

C Prototype

```
ViStatus IviUpconverter_CreateGainList (ViSession Vi,
                                        ViConstString Name,
                                        ViInt32 GainBufferSize,
                                        ViReal64 Gain[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Name	Specifies the name of the list to be created.	ViConstString
GainBufferSize	Specifies the number of elements in the gain list.	ViInt32
Gain (C/COM) gainList (.NET)	Specifies the array of Gain values to become elements of the list. The array must have at least as many elements as the value in the GainBufferSize parameter. The units are dB.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.4.Create Frequency List

Description

This function creates a named list of frequency values.

.NET Method Prototype

```
void Sweep.List.CreateFrequency (String name,
                                 Double[] frequencyList);
```

COM Method Prototype

```
HRESULT Sweep.List.CreateFrequency ([in] BSTR Name,
                                    [in] SAFEARRAY(double)* Frequency);
```

C Prototype

```
ViStatus IviUpconverter_CreateFrequencyList (ViSession Vi,
                                             ViConstString Name,
                                             ViInt32 FrequencyBufferSize,
                                             ViReal64 Frequency[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Name	Specifies the name of the list to be created.	ViConstString
FrequencyBufferSize	Specifies the number of elements in the frequency list.	ViInt32
Frequency (C/COM) frequencyList (.NET)	Specifies the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the FrequencyBufferSize parameter. The units are Hertz.	ViReal64[]

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.5.Create Frequency Power List

Description

This function creates a named list of frequency and power value pairs.

.NET Method Prototype

```
struct FrequencyPower
{
    public FrequencyPower (Double frequency,, Double power);
    public Double Frequency { get; }
    public Double Power { get; }
}


void Sweep.List.CreateFrequencyPower (String name,
                                      FrequencyPower[] frequencyPowerList);
);
```

COM Method Prototype

```
HRESULT Sweep.List.CreateFrequencyPower ([in] BSTR Name,
                                        [in] SAFEARRAY(double)* Frequency,
                                        [in] SAFEARRAY(double)* Power);
```

C Prototype

```
ViStatus IviUpconverter_CreateFrequencyPowerList (ViSession Vi,
                                        ViConstString Name,
                                        ViInt32 BufferSize,
                                        ViReal64 Frequency[],
                                        ViReal64 Power[]);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Name`	Specifies the name of the list to be created.	`ViConstString`
`BufferSize`	Specifies the number of elements in the frequency and power lists.	`ViInt32`
`Frequency`	Specifies the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the BufferSize parameter. The units are Hertz.	`ViReal64[]`
`Power`	Specifies the array of power values to become elements of the list. The array must have at least as many elements as the value in the BufferSize parameter. The units are dBm.	`ViReal64[]`
`frequencyPowerList`	Specifies the array of frequency and power values to become elements of the list. The units for frequency are Hertz and for power are dBm.	`FrequencyPower[]`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.6.Create Frequency Gain List

Description

This function creates a named list of frequency and gain value pairs.

.NET Method Prototype

```
struct FrequencyGain
{
    public FrequencyGain (Double frequency,, Double gain);
    public Double Frequency { get; }
    public Double Gain { get; }
}


void Sweep.List.CreateFrequencyGain (String name,
                                     FrequencyGain[] frequencyGainList);
n);
```

COM Method Prototype

```
HRESULT Sweep.List.CreateFrequencyGain ([in] BSTR Name,
                                        [in] SAFEARRAY(double)* Frequency,
                                        [in] SAFEARRAY(double)* Gain);
```

C Prototype

```
ViStatus IviUpconverter_CreateFrequencyGainList (ViSession Vi,
                                        ViConstString Name,
                                        ViInt32 BufferSize,
                                        ViReal64 Frequency[],
                                        ViReal64 Gain[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Name	Specifies the name of the list to be created.	ViConstString
BufferSize	Specifies the number of elements in the frequency and gain lists.	ViInt32
Frequency	Specifies the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the BufferSize parameter. The units are Hertz.	ViReal64[]
Gain	Specifies the array of gain values to become elements of the list. The array must have at least as many elements as the value in the BufferSize parameter. The units are dB.	ViReal64[]
frequencyGainList	Specifies the array of frequency and gain values to become elements of the list. The units for frequency are Hertz and for power are dB.	FrequencyGain[]

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.7.Create Power List

Description

This function creates a named list of power values.

.NET Method Prototype

```
void Sweep.List.CreatePower (String name,
                             Double[] powerList);
```

COM Method Prototype

```
HRESULT Sweep.List.CreatePower ([in] BSTR Name,
                                [in] SAFEARRAY(double)* Power);
```

C Prototype

```
ViStatus IviUpconverter_CreatePowerList (ViSession Vi,
                                         ViConstString Name,
                                         ViInt32 PowerBufferSize,
                                         ViReal64 Power[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Name	Specifies the name of the list to be created.	ViConstString
PowerBufferSize	Specifies the number of elements in the power list.	ViInt32
Power (C/COM) powerList (.NET)	Specifies the array of power values to become elements of the list. The array must have at least as many elements as the value in the PowerBufferSize parameter. The units are dBm.	ViReal64

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.3.8.Reset List

Description

Resets the current list to the first entry value

.NET Method Prototype

```
void Sweep.List.Reset ();
```

COM Method Prototype

```
HRESULT Sweep.List.Reset ();
```

C Prototype

```
ViStatus IviUpconverter_ResetList (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

21.4.IviUpconverterList Behavior Model

The IviUpconverterList extension group follows the same behavior model as the IviUpconverterSweep capability group described in Section 14.4, *IviUpconverterSweep Behavior Model.*

21.5.IviUpconverterList Compliance Notes

1. If a specific driver implements the IviUpconverterList Extension Group, it shall also implement the IviUpconverterSweep Extension Group and support the defined value List for the Sweep Mode attribute.

22.IviUpconverterALC Extension Group

22.1.IviUpconverterALC Overview

The IviUpconverterALC extension group supports upconverters with configurable automatic level control (ALC).

22.2.IviUpconverterALC Attributes

The IviUpconverterALC extension group defines the following attributes:

- ALC Bandwidth
- ALC Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

22.2.1.ALC Bandwidth

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	N/A	Up	Configure ALC

.NET Property Name

```
Alc.Bandwidth
```

COM Property Name

```
ALC.Bandwidth
```

C Constant Name

```
IVIUPCONVERTER_ATTR_ALC_BANDWIDTH
```

Description

Specifies the bandwidth of the level control. Narrow bandwidth improves noise and allows AM with modulation frequencies beyond bandwidth frequency. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

22.2.2.ALC Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure ALC

.NET Property Name

`Alc.Source`

.NET Enumeration Name

`AlcSource`

COM Property Name

`ALC.Source`

COM Enumeration Name

`IviUpconverterALCSourceEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_ALC_SOURCE`

Description

Specifies the source of the controlling voltage for the Automatic Level Control. The RF level at the sensor point is held constant.

Defined Values

Name	*Description*		
		Language	*Identifier*
Internal	The ALC is controlled by an internal measurement source.		
		.NET	`AlcSource.Internal`
		C	`IVIUPCONVERTER_VAL_ALC_SOURCE_INTERNAL`
		COM	`IviUpconverterALCSourceInternal`
External	The ALC is controlled by an external voltage.		
		.NET	`AlcSource.External`
		C	`IVIUPCONVERTER_VAL_ALC_SOURCE_EXTERNAL`
		COM	`IviUpconverterALCSourceExternal`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_ALC_SOURCE_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to ALC Source Specific Ext Base.

   See Section Attribute Value Definitions, for the definitions of ALC Source Specific Ext Base, IVIUPCONVERTER_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_ALC_SOURCE_CLASS_EXT_BASE.

22.3.IviUpconverterALC Functions

The IviUpconverterALC extension group defines the following function:

- Configure ALC

This section describes the behavior and requirements of this function.

22.3.1.Configure ALC

Description

This function configures the ALC (Automatic Level Control) of the upconverter's RF output.

.NET Method Prototype

```
void Alc.Configure (AlcSource source, Double bandwidth);
```

COM Method Prototype

```
HRESULT ALC.Configure ([in] IviUpconverterALCSourceEnum Source,
                       [in] double Bandwidth);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureALC (ViSession Vi,
                                      ViInt32 Source,
                                      ViReal64 Bandwidth);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Source`	Specifies the source for the ALC. The driver uses this value to set the ALC Source attribute. See the attribute description for more details.	`ViConstString`
`Bandwidth`	Specifies the ALC bandwidth. The driver uses this value to set the ALC Bandwidth attribute. See the attribute description for more details.	`ViReal64`

Defined Values for the Source Parameter

Name	*Description*		
		Language	*Identifier*
Internal	The ALC is controlled by an internal measurement source.		
		.NET	`AlcSource.Internal`
		C	`IVIUPCONVERTER_VAL_ALC_SOURCE_INTERNAL`
		COM	`IviUpconverterALCSourceInternal`
External	The ALC is controlled by an external voltage.		
		.NET	`AlcSource.External`
		C	`IVIUPCONVERTER_VAL_ALC_SOURCE_EXTERNAL`
		COM	`IviUpconverterALCSourceExternal`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

22.4.IviUpconverterALC Behavior Model

The IviUpconverterALC extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model.*

22.5.IviUpconverterALC Compliance Notes

For a specific driver to comply with the IviUpconverterALC extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

23.IviUpconverterCalibration Extension Group

23.1.IviUpconverterCalibration Overview

The IviUpconverterCalibration extension capability group supports upconverters that can perform self-calibration. The IviUpconverterCalibration capability group defines functions to perform the calibration.

23.2.IviUpconverterCalibration Functions

The IviUpconverterCalibration extension group defines the following function:

- Calibrate
- Is Calibration Complete

This section describes the behavior and requirements of this function.

23.2.1.Calibrate

Description

This function performs calibration on the entire device. This call can be blocking or can be non-blocking, depending on the instrument implementation. If it is non-blocking, the user may use the Is Calibration Complete function to determine when the calibration is complete.

.NET Method Prototype

```
void RFOutput.Calibrate();
```

COM Method Prototype

```
HRESULT RFOutput.Calibrate();
```

C Prototype

```
ViStatus IviUpconverter_Calibrate (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

23.2.2.Is Calibration Complete

Description

This function queries the instrument to determine the status of all calibration operations initiated by the Calibrate function. This function returns the Calibration Complete value in the `Status` parameter only when calibration is complete.

If some calibration operations are still in progress, the driver returns the Calibration In Progress value. If the driver cannot query the instrument to determine its state, the driver returns the Calibration Status Unknown value.

This function does not check the instrument status. Typically, the end-user calls this function only in a sequence of calls to other low-level driver functions. The sequence performs one operation. The end-user uses the low-level functions to optimize one or more aspects of interaction with the instrument. To check the instrument status, call the Error Query function at the conclusion of the sequence.

.NET Method Prototype

```
CalibrationStatus RFOutput.IsCalibrationComplete();
```

COM Method Prototype

```
HRESULT RFOutput.IsCalibrationComplete(IviUpconverterCalibrationStatusEnum*
Status);
```

C Prototype

```
ViStatus IviUpconverter_IsCalibrationComplete (ViSession Vi,
                                               ViInt32* Status);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`

Outputs	Description	Base Type

`Status (C/COM)`
`Return value (.NET)`
	Returns the status of the calibration operation.	`ViInt32`

Defined Values for the Status Parameter

Name	*Description*		
		Language	*Identifier*
Calibration Complete	The upconverter has completed the calibration.		
		.NET	`CalibrationStatus.Complete`
		C	`IVIUPCONVERTER_VAL_CALIBRATION_COMPLETE`
		COM	`IviUpconverterCalibrationComplete`
Calibration In Progress	The upconverter is still performing the calibration.		
		.NET	`CalibrationStatus.InProgress`
		C	`IVIUPCONVERTER_VAL_CALIBRATION_IN_PROGRESS`
		COM	`IviUpconverterCalibrationInProgress`
Calibration Status Unknown	The upconverter cannot determine the status of the calibration.		
		.NET	`CalibrationStatus.StatusUnknown`
		C	`IVIUPCONVERTER_VAL_CALIBRATION_STATUS_UNKNOWN`
		COM	`IviUpconverterCalibrationStatusUnknown`
Calibration Failed	`The upconverter calibration failed.`		
		.NET	`CalibrationStatus.Failed`
		C	`IVIUPCONVERTER_VAL_CALIBRATION_FAILED`
		COM	`IviUpconverterCalibrationFailed`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

23.3.IviUpconverterCalibration Behavior Model

The IviUpconverterCalibration extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model.*

23.4.IviUpconverterCalibration Compliance Notes

For a specific driver to comply with the IviUpconverterCalibration extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

24.IviUpconverterAttenuatorHold Extension Group

24.1.IviUpconverterAttenuatorHold Overview

The IviUpconverterAttenuatorHold extension group supports upconverters with the ability to hold the attenuators at a constant value.

24.2.IviUpconverterAttenuatorHold Attributes

The IviUpconverterAttenuatorHold extension group defines the following attributes:

- Attenuator Hold Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions*.

24.2.1.Attenuator Hold Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	RFOutput	None	ConfigureAttenuatorHoldEnabled (IVI-C Only)

.NET Property Name

`RFOutput.AttenuatorHoldEnabled`

COM Property Name

`RFOutput.AttenuatorHoldEnabled`

C Constant Name

`IVIUPCONVERTER_ATTR_ATTENUATOR_HOLD_ENABLED`

Description

Enables or disables the attenuator hold on the active RF output.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

24.3.IviUpconverterAttenuatorHold Functions

The IviUpconverterAttenuatorHold extension group defines the following function:

- Configure Attenuator Hold Enabled (IVI-C Only)

This section describes the behavior and requirements of this function.

24.3.1.Configure Attenuator Hold Enabled (IVI-C Only)

Description

This function configures the attenuator hold enabled setting.

.NET Method Prototype

```
N/A
```

(use the `RFOutput.AttenuatorHoldEnabled` property)

COM Method Prototype

```
N/A
```

(use the `RFOutput.AttenuatorHoldEnabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureAttenuatorHoldEnabled (ViSession Vi,
                                       ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Specifies whether or not the attenuator hold is enabled. The driver uses this value to set the Attenuator Hold Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

24.4.IviUpconverterAttenuatorHold Behavior Model

The IviUpconverterAttenuatorHold extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model.*

24.5.IviUpconverterAttenuatorHold Compliance Notes

For a specific driver to comply with the IviUpconverterAttenuatorHold extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

25.IviUpconverterReferenceOscillator Extension Group

25.1.IviUpconverterReferenceOscillator Overview

The IviUpconverterReferenceOscillator extension group supports upconverters with a configurable frequency reference.

25.2.IviUpconverterReferenceOscillator Attributes

The IviUpconverterReferenceOscillator extension group defines the following attributes:

- Reference Oscillator External Frequency
- Reference Oscillator Source
- Reference Oscillator Output Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

25.2.1.Reference Oscillator External Frequency

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure Reference Oscillator

.NET Property Name

```
ReferenceOscillator.ExternalFrequency
```

COM Property Name

```
ReferenceOscillator.ExternalFrequency
```

C Constant Name

```
IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY
```

Description

Specifies the frequency of the external signal that is used as reference for internal RF frequency generation. This value is used only if Reference Oscillator Source is set to External. The units are Hertz.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

25.2.2.Reference Oscillator Source

Data Type	Access	Applies To	Coercion	High Level Functions
`ViInt32`	R/W	N/A	None	Configure Reference Oscillator

.NET Property Name

`ReferenceOscillator.Source`

.NET Enumeration Name

`ReferenceOscillatorSource`

COM Property Name

`ReferenceOscillator.Source`

COM Enumeration Name

`IviUpconverterReferenceOscillatorSourceEnum`

C Constant Name

`IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_SOURCE`

Description

Specifies the reference oscillator source used to generate the precise RF output frequency.

Defined Values

Name	***Description***		
		Language	***Identifier***
Internal	The upconverter uses the internal reference oscillator.		
		.NET	`ReferenceOscillatorSource.Internal`
		C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL`
		COM	`IviUpconverterReferenceOscillatorSourceInternal`
External	The upconverter uses an external reference oscillator.		
		.NET	`ReferenceOscillatorSource.External`
		C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL`
		COM	`IviUpconverterReferenceOscillatorSourceExternal`

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_CLASS_EXT_BASE and less than IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SPECIFIC_EXT_BASE.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SPECIFIC_EXT_BASE.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Reference Oscillator Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Reference Oscillator Specific Ext Base, IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SPECIFIC_EXT_BASE and IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_CLASS_EXT_BASE.

25.2.3.Reference Oscillator Output Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure Reference Oscillator Output Enabled (IVI-C Only)

.NET Property Name

```
ReferenceOscillator.OutputEnabled
```

COM Property Name

```
ReferenceOscillator.OutputEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_OUTPUT_ENABLED
```

Description

Enables or disables the Reference output. Many upconverters support the ability to output their frequency reference. This attribute allows the user to enable or disable that output.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

25.3.IviUpconverterReferenceOscillator Functions

The IviUpconverterReferenceOscillator extension group defines the following function:

- Configure Reference Oscillator
- Configure Reference Oscillator Output Enabled (IVI-C Only)

This section describes the behavior and requirements of this function.

25.3.1.Configure Reference Oscillator

Description

Configures the upconverter's reference oscillator.

.NET Method Prototype

```
void ReferenceOscillator.Configure (ReferenceOscillatorSource source,
                                    Double frequency);
```

COM Method Prototype

```
HRESULT ReferenceOscillator.Configure ([in]
IviUpconverterReferenceOscillatorSourceEnum Source,
                                       [in] double Frequency);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureReferenceOscillator (ViSession Vi,
                                        ViInt32 Source,
                                        ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Source	Specifies the source of the reference frequency signal. The driver uses this value to set the Reference Oscillator Source attribute. See the attribute description for more details.	ViConstString
Frequency	Specifies the frequency of the external reference oscillator. This parameter is only used if the Source is set to External. The driver uses this value to set the Reference Oscillator External Frequency attribute. See the attribute description for more details.	ViReal64

Defined Values for the Source Parameter

Name	*Description*		
		Language	*Identifier*
Internal	The upconverter uses the internal reference oscillator.		
		.NET	`ReferenceOscillatorSource.Internal`
		C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL`
		COM	`IviUpconverterReferenceOscillatorSourceInternal`
External	The upconverter uses an external reference oscillator.		
		.NET	`ReferenceOscillatorSource.External`
		C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL`
		COM	`IviUpconverterReferenceOscillatorSourceExternal`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

25.3.2. Configure Reference Oscillator Output Enabled (IVI-C Only)

Description

Configures the upconverter's reference oscillator output. Many upconverters support the ability to output their frequency reference. This function allows the user to enable or disable that output.

.NET Method Prototype

```
N/A
```

(use the `ReferenceOscillator.OutputEnabled` property)

COM Method Prototype

```
N/A
```

(use the `ReferenceOscillator.OutputEnabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureReferenceOscillatorOutputEnabled (
                                                        ViSession Vi,
                                                        ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Enables the frequency reference output.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

25.4.IviUpconverterReferenceOscillator Behavior Model

The IviUpconverterReferenceOscillator extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model*.

25.5.IviUpconverterReferenceOscillator Compliance Notes

For a specific driver to comply with the IviUpconverterReferenceOscillator extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

26.IviUpconverterSoftwareTrigger Extension Group

26.1.IviUpconverterSoftwareTrigger Overview

The IviUpconverterSoftwareTrigger Extension Group supports upconverters that can generate output based on a software trigger signal. The user can send a software trigger to start a sweep (frequency, power level, or gain), or to initiate the next step (frequency, power level, or gain).

This extension affects instrument behavior when the Sweep Trigger Source attribute is set to “Software.”

26.2.IviUpconverterSoftwareTrigger Functions

The IviUpconverterSoftwareTrigger extension group defines the following function:

- Send Software Trigger

This section describes the behavior and requirements of this function.

26.2.1.Send Software Trigger

Refer to *IVI-3.3: Standard Cross Class Capabilities, Section 2 Software Triggering Capability* for the prototype and complete description of this function.

26.3.IviUpconverterSoftwareTrigger Behavior Model

The IviUpconverterSoftwareTrigger Extension Group follows the behavior model of the IviUpconverterBase capability group.

26.4.IviUpconverterSoftwareTrigger Compliance Notes

1. If a specific driver implements the IviUpconverterSoftwareTrigger Extension Group, it shall implement the IviUpconverterSweep Extenstion Group.

2. If a specific driver implements the IviUpconverterSoftwareTrigger Extension Group together with the IviUpconverterSweep Extension Group, it shall implement the value Software Trigger for the Sweep Trigger Source attribute.

27.IviUpconverterModulateIQ Extension Group

27.1.IviUpconverterModulateIQ Overview

The IviUpconverterModulateIQ Extension Group supports upconverters that can apply IQ (vector) modulation to the RF output signal. The user can enable or disable IQ modulation and specify the source of the modulating signal.

27.2.IviUpconverterModulateIQ Attributes

The IviUpconverterModulateIQ extension group defines the following attributes:

- IQ Enabled
- IQ Nominal Voltage
- IQ Swap Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

27.2.1.IQ Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure IQ Enabled (IVI-C Only)

.NET Property Name

```
IQ.Enabled
```

COM Property Name

```
IQ.Enabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IQ_ENABLED
```

Description

Enables or disables IQ (vector) modulation of the RF output signal.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

27.2.2.IQ Nominal Voltage

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	RO	N/A	None	N/A

.NET Property Name

```
IQ.NominalVoltage
```

COM Property Name

```
IQ.NominalVoltage
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IQ_NOMINAL_VOLTAGE
```

Description

Returns the voltage at which the instrument achieves full modulation. The value is calculated by SQRT(I2+Q2). The units are Volts.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

27.2.3.IQ Swap Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	N/A

.NET Property Name

```
IQ.SwapEnabled
```

COM Property Name

```
IQ.SwapEnabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IQ_SWAP_ENABLED
```

Description

Enables or disables the inverse phase rotation of the IQ signal by swapping the I and Q inputs.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

27.3.IviUpconverterModulateIQ Functions

The IviUpconverterModulateIQ extension group defines the following function:

- Calibrate IQ
- Configure IQ Enabled (IVI-C Only)

This section describes the behavior and requirements of this function.

27.3.1.Calibrate IQ

Description

This function performs a calibration of the IQ modulator.

.NET Method Prototype

```
void IQ.Calibrate ();
```

COM Method Prototype

```
HRESULT IQ.Calibrate ();
```

C Prototype

```
ViStatus IviUpconverter_CalibrateIQ (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

27.3.2.Configure IQ Enabled (IVI-C Only)

Description

Configures the upconverter to apply IQ (vector) modulation to the RF output signal.

.NET Method Prototype

```
N/A
```

(use the `IQ.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `IQ.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureIQEnabled (ViSession Vi,
                                            ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`Enabled`	Enables or disables IQ (vector) modulation of the RF output signal. The driver uses this value to set the IQ Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

27.4.IviUpconverterModulateIQ Behavior Model

The IviUpconverterModulateIQ extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model.*

27.5.IviUpconverterModulateIQ Compliance Notes

For a specific driver to comply with the IviUpconverterModulateIQ extension, it shall be compliant with the IviUpconverterBase capability group and it shall implement all of the attributes and functions listed in this section.

28. IviUpconverterIQImpairment Extension Group

28.1. IviUpconverterIQImpairment Overview

The IviUpconverterIQImpairment extension group supports upconverters that can simulate or correct impairment on IQ (vector) modulation. This group is an extension of the IviUpconverterModulateIQ Extension Group.

28.2. IviUpconverterIQImpairment Attributes

The IviUpconverterIQImpairment extension group defines the following attributes:

- IQ Impairment Enabled
- IQ Impairment I Offset
- IQ Impairment Q Offset
- IQ Impairment Ratio
- IQ Impairment Skew

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 22, *IviUpconverter Attribute ID Definitions.*

28.2.1.IQ Impairment Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
`ViBoolean`	R/W	N/A	None	Configure IQ Impairment Enabled (IVI-C Only)

.NET Property Name

```
IQ.Impairment.Enabled
```

COM Property Name

```
IQ.Impairment.Enabled
```

C Constant Name

```
IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_ENABLED
```

Description

Enables or disables impairment attributes.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

28.2.2.IQ Impairment I Offset

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

```
IQ.Impairment.IOffset
```

COM Property Name

```
IQ.Impairment.IOffset
```

C Constant Name

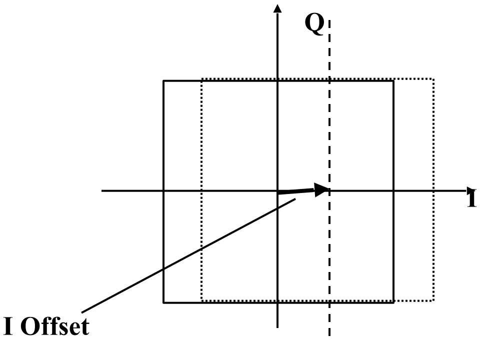
```
IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_I_OFFSET
```

Description

Specifies an origin offset voltage to the I signal. The range of values allowed is ?100% to +100%. The value is expressed as percentage (%).

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.



28.2.3.IQ Impairment Q Offset

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

```
IQ.Impairment.QOffset
```

COM Property Name

```
IQ.Impairment.QOffset
```

C Constant Name

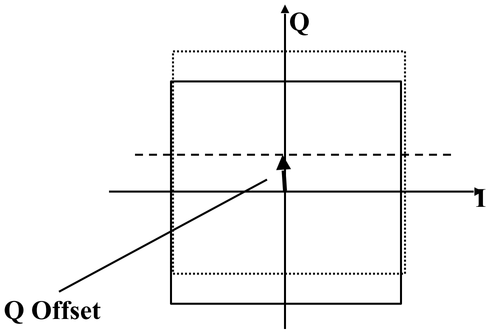
```
IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_Q_OFFSET
```

Description

Specifies an origin offset voltage to the Q signal. The range of values allowed is ?100% to +100%. The value is expressed as percentage (%).

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.



28.2.4.IQ Impairment Ratio

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

`IQ.Impairment.Ratio`

COM Property Name

`IQ.Impairment.Ratio`

C Constant Name

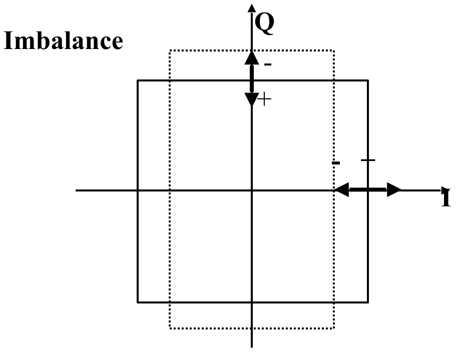
`IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_RATIO`

Description

Specifies the gain imbalance between the I and Q channels. For no imbalance this value is set to 0 %. The value is expressed as percentage (%).

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.



28.2.5.IQ Impairment Skew

Data Type	Access	Applies To	Coercion	High Level Functions
`ViReal64`	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

`IQ.Impairment.Skew`

COM Property Name

`IQ.Impairment.Skew`

C Constant Name

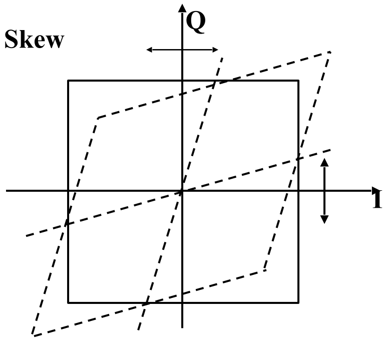
`IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_SKEW`

Description

Specifies the adjustment of the phase angle between the I and Q vectors. If this skew is zero, the phase angle is 90 degrees. The units are degrees.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.



28.3.IviUpconverterIQImpairment Functions

The IviUpconverterIQImpairment extension group defines the following function:

- Configure IQ Impairment
- Configure IQ Impairment Enabled (IVI-C Only)

This section describes the behavior and requirements of this function.

28.3.1.Configure IQ Impairment

Description

This function configures the attributes that simulate or correct impairment for the upconverter's IQ (vector) modulation.

.NET Method Prototype

```
void IQ.Impairment.Configure (Double iOffset,
                              Double qOffset,
                              Double ratio,
                              Double skew);
```

COM Method Prototype

```
HRESULT IQ.Impairment.Configure ([in] double IOffset,
                                 [in] double QOffset,
                                 [in] double Ratio,
                                 [in] double Skew);
```

C Prototype

```
ViStatus IviUpconverter_ConfigureIQImpairment (ViSession Vi,
                                        ViReal64 IOffset,
                                        ViReal64 QOffset,
                                        ViReal64 Ratio,
                                        ViReal64 Skew);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`IOffset`	Specifies an offset to the I-path of IQ signals. The driver uses this value to set the IQ Impairment I Offset attribute. See the attribute description for more details.	`ViReal64`
`QOffset`	Specifies an offset to the Q-path of IQ signals. The driver uses this value to set the IQ Impairment Q Offset attribute. See the attribute description for more details.	`ViReal64`
`Ratio`	Specifies a gain imbalance to the IQ signals. The driver uses this value to set the IQ Impairment Ratio attribute. See the attribute description for more details.	`ViReal64`
`Skew`	Specifies an angle offset to the IQ signals. The driver uses this value to set the IQ Impairment Skew attribute. See the attribute description for more details.	`ViReal64`

Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

28.3.2.Configure IQ Impairment Enabled (IVI-C Only)

Description

Configures the IQ modulation to allow controlled impairment for test or external corrections.

.NET Method Prototype

```
N/A
```

(use the `IQ.Impairment.Enabled` property)

COM Method Prototype

```
N/A
```

(use the `IQ.Impairment.Enabled` property)

C Prototype

```
ViStatus IviUpconverter_ConfigureIQImpairmentEnabled (ViSession Vi,
                                    ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle.	`ViSession`
`IQImpairmentEnabled`	Enables or disables impairment attributes. The driver uses this value to set the IQ Impairment Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

28.4.IviUpconverterIQImpairment Behavior Model

The IviUpconverterIQImpairment extension group follows the same behavior model as the IviUpconverterBase capability group described in Section 4.4, *IviUpconverterBase Behavior Model*.

28.5.IviUpconverterIQImpairment Compliance Notes

1. If a specific driver implements the IviUpconverterIQImpairment Extension Group, it shall also implement the IviUpconverterModulateIQ Extension Group.

29.IviUpconverter Attribute ID Definitions

The following table defines the ID value for all IviUpconverter class attributes.

Table 29-1. IviUpconverter Attributes ID Values

Attribute Name	ID Definition
Active IF Input	`IVI_CLASS_ATTR_BASE + 0`
Active RF Output	`IVI_CLASS_ATTR_BASE + 1`
ALC Enabled	`IVI_CLASS_ATTR_BASE + 2`
Auto Corrections Enabled	`IVI_CLASS_ATTR_BASE + 3`
External LO Enabled	`IVI_CLASS_ATTR_BASE + 4`
External LO Frequency	`IVI_CLASS_ATTR_BASE + 5`
IF Input Attenuation	`IVI_CLASS_ATTR_BASE + 6`
IF Input Count	`IVI_CLASS_ATTR_BASE + 7`
IF Input Coupling	`IVI_CLASS_ATTR_BASE + 8`
IF Input Frequency	`IVI_CLASS_ATTR_BASE + 9`
Is Ready	`IVI_CLASS_ATTR_BASE + 10`
RF Output Bandwidth	`IVI_CLASS_ATTR_BASE + 11`
RF Output Count	`IVI_CLASS_ATTR_BASE + 12`
RF Output Enabled	`IVI_CLASS_ATTR_BASE + 13`
RF Output Frequency	`IVI_CLASS_ATTR_BASE + 14`
RF Output Gain	`IVI_CLASS_ATTR_BASE + 100`
RF Output Level	`IVI_CLASS_ATTR_BASE + 110`
AM Depth	`IVI_CLASS_ATTR_BASE + 120`
AM Enabled	`IVI_CLASS_ATTR_BASE + 121`
AM External Coupling	`IVI_CLASS_ATTR_BASE + 122`
AM Nominal Voltage	`IVI_CLASS_ATTR_BASE + 123`
AM Scaling	`IVI_CLASS_ATTR_BASE + 124`
AM Source	`IVI_CLASS_ATTR_BASE + 125`
FM Deviation	`IVI_CLASS_ATTR_BASE + 130`
FM Enabled	`IVI_CLASS_ATTR_BASE + 131`
FM External Coupling	`IVI_CLASS_ATTR_BASE + 132`
FM Nominal Voltage	`IVI_CLASS_ATTR_BASE + 133`
FM Source	`IVI_CLASS_ATTR_BASE + 134`
PM Deviation	`IVI_CLASS_ATTR_BASE + 140`
PM Enabled	`IVI_CLASS_ATTR_BASE + 141`

Table 29-1. IviUpconverter Attributes ID Values

Attribute Name	ID Definition
PM External Coupling	`IVI_CLASS_ATTR_BASE + 142`
PM Nominal Voltage	`IVI_CLASS_ATTR_BASE + 143`
PM Source	`IVI_CLASS_ATTR_BASE + 144`
Analog Modulation Source Count	`IVI_CLASS_ATTR_BASE + 150`
Pulse Modulation Enabled	`IVI_CLASS_ATTR_BASE + 160`
Pulse Modulation External Polarity	`IVI_CLASS_ATTR_BASE + 161`
Bypass	`IVI_CLASS_ATTR_BASE + 170`
RF Output Ready Trigger	`IVI_CLASS_ATTR_BASE + 180`
Is Sweeping	`IVI_CLASS_ATTR_BASE + 200`
Sweep Mode	`IVI_CLASS_ATTR_BASE + 201`
Sweep Trigger Source	`IVI_CLASS_ATTR_BASE + 202`
Frequency Sweep Start	`IVI_CLASS_ATTR_BASE + 210`
Frequency Sweep Stop	`IVI_CLASS_ATTR_BASE + 211`
Frequency Sweep Time	`IVI_CLASS_ATTR_BASE + 212`
Power Sweep Start	`IVI_CLASS_ATTR_BASE + 220`
Power Sweep Stop	`IVI_CLASS_ATTR_BASE + 221`
Power Sweep Time	`IVI_CLASS_ATTR_BASE + 222`
Gain Sweep Start	`IVI_CLASS_ATTR_BASE + 230`
Gain Sweep Stop	`IVI_CLASS_ATTR_BASE + 231`
Gain Sweep Time	`IVI_CLASS_ATTR_BASE + 232`
Frequency Step Dwell	`IVI_CLASS_ATTR_BASE + 240`
Frequency Step Scaling	`IVI_CLASS_ATTR_BASE + 241`
Frequency Step Single Step Enabled	`IVI_CLASS_ATTR_BASE + 242`
Frequency Step Size	`IVI_CLASS_ATTR_BASE + 243`
Frequency Step Start	`IVI_CLASS_ATTR_BASE + 244`
Frequency Step Stop	`IVI_CLASS_ATTR_BASE + 245`
Power Step Dwell	`IVI_CLASS_ATTR_BASE + 250`
Power Step Single Step Enabled	`IVI_CLASS_ATTR_BASE + 251`
Power Step Size	`IVI_CLASS_ATTR_BASE + 252`
Power Step Start	`IVI_CLASS_ATTR_BASE + 253`
Power Step Stop	`IVI_CLASS_ATTR_BASE + 254`
Gain Step Dwell	`IVI_CLASS_ATTR_BASE + 260`

Table 29-1. IviUpconverter Attributes ID Values

Attribute Name	ID Definition
Gain Step Single Step Enabled	`IVI_CLASS_ATTR_BASE + 261`
Gain Step Size	`IVI_CLASS_ATTR_BASE + 262`
Gain Step Start	`IVI_CLASS_ATTR_BASE + 263`
Gain Step Stop	`IVI_CLASS_ATTR_BASE + 264`
List Dwell	`IVI_CLASS_ATTR_BASE + 270`
List Selected Name	`IVI_CLASS_ATTR_BASE + 271`
List Single Step Enabled	`IVI_CLASS_ATTR_BASE + 272`
ALC Bandwidth	`IVI_CLASS_ATTR_BASE + 300`
ALC Source	`IVI_CLASS_ATTR_BASE + 301`
Attenuator Hold Enabled	`IVI_CLASS_ATTR_BASE + 310`
Reference Oscillator External Frequency	`IVI_CLASS_ATTR_BASE + 320`
Reference Oscillator Source	`IVI_CLASS_ATTR_BASE + 321`
Reference Oscillator Output Enabled	`IVI_CLASS_ATTR_BASE + 322`
IQ Enabled	`IVI_CLASS_ATTR_BASE + 330`
IQ Nominal Voltage	`IVI_CLASS_ATTR_BASE + 331`
IQ Swap Enabled	`IVI_CLASS_ATTR_BASE + 332`
IQ Impairment Enabled	`IVI_CLASS_ATTR_BASE + 340`
IQ Impairment I Offset	`IVI_CLASS_ATTR_BASE + 341`
IQ Impairment Q Offset	`IVI_CLASS_ATTR_BASE + 342`
IQ Impairment Ratio	`IVI_CLASS_ATTR_BASE + 343`
IQ Impairment Skew	`IVI_CLASS_ATTR_BASE + 344`

30.IviUpconverter Attribute Value Definitions

This section specifies the actual value for each defined attribute value.

ALC Source

Value Name	*Language*	*Identifier*	*Actual Value*
Internal	.NET	`AlcSource.Internal`	0
	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_INTERNAL`	0
	COM	`IviUpconverterALCSourceInternal`	0
External	.NET	`AlcSource.External`	1
	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_EXTERNAL`	1
	COM	`IviUpconverterALCSourceExternal`	1
ALC Source Class Ext Base	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_CLASS_EXT_BASE`	100
ALC Source Specific Ext Base	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

AM External Coupling

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_AC`	0
	COM	`IviUpconverterAMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterAMExternalCouplingDC`	1
AM External Coupling Class Ext Base	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE`	100
AM External Coupling Specific Ext Base	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

AM Scaling

Value Name	*Language*	*Identifier*	*Actual Value*
Linear	.NET	`Scaling.Linear`	0
	C	`IVIUPCONVERTER_VAL_AM_SCALING_LINEAR`	0
	COM	`IviUpconverterAMScalingLinear`	0
Logarithmic	.NET	`Scaling.Logarithmic`	1
	C	`IVIUPCONVERTER_VAL_AM_SCALING_LOGARITHMIC`	1
	COM	`IviUpconverterAMScalingLogarithmic`	1
AM Scaling Class Ext Base	C	`IVIUPCONVERTER_VAL_AM_SCALING_ CLASS_EXT_BASE`	100
AM Scaling Specific Ext Base	C	`IVIUPCONVERTER_VAL_AM_SCALING_ SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

FM External Coupling

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_AC`	0
	COM	`IviUpconverterFMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterFMExternalCouplingDC`	1
FM External Coupling Class Ext Base	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE`	100
FM External Coupling Specific Ext Base	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

Frequency Step Scaling

Value Name	*Language*	*Identifier*	*Actual Value*
Linear	.NET	`Scaling.Linear`	0
	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LINEAR`	0

Value Name	*Language*	*Identifier*	*Actual Value*
	COM	`IviUpconverterFrequencyStepScalingLinear`	0
Logarithmic	.NET	`Scaling.Logarithmic`	1
	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC`	1
	COM	`IviUpconverterFrequencyStepScalingLogarithmic`	1
Frequency Step Scaling Class Ext Base	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE`	100
Frequency Step Scaling Specific Ext Base	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

IF Input Coupling

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_AC`	0
	COM	`IviUpconverterIFInputCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_DC`	1
	COM	`IviUpconverterIFInputCouplingDC`	1
IF Input Coupling Class Ext Base	C	`IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_CLASS_EXT_BASE`	100
IF Input Coupling Specific Ext Base	C	`IVIUPCONVERTER_VAL_IF_INPUT_COUPLING_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

Output Ready Trigger and Sweep Trigger Source

Name	***Description***			
		Language	***Identifier***	***Actual Value***
None	No trigger source			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_NONE`	`“”`
		COM/.NET	`N/A`	`“”`
Immediate	Trigger Immediately			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_IMMEDIATE`	`“Immediate”`
		COM/.NET	`N/A`	`“Immediate”`
External	External trigger source			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_EXTERNAL`	`“External”`
		COM/.NET	`N/A`	`“External”`
Internal	Internal trigger source			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_INTERNAL`	`"Internal"`
		COM/.NET	`N/A`	`"Internal"`
Software	Software trigger			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_SOFTWARE`	`“Software”`
		COM/.NET	`N/A`	`“Software”`
LAN0	LAN0 (LXI defined “LAN0” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN0`	`“LAN0”`
		COM/.NET	`N/A`	`“LAN0”`
LAN1	LAN1 (LXI defined “LAN1” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN1`	`“LAN1”`
		COM/.NET	`N/A`	`“LAN1”`
LAN2	LAN2(LXI defined “LAN2” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN2`	`“LAN2”`
		COM/.NET	`N/A`	`“LAN2”`
LAN3	LAN3 (LXI defined “LAN3” LAN message)			

Name	*Description*			
		Language	*Identifier*	*Actual Value*
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN3`	`“LAN3”`
		COM/.NET	`N/A`	`“LAN3”`
LAN4	LAN4 (LXI defined “LAN4” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN4`	`“LAN4”`
		COM/.NET	`N/A`	`“LAN4”`
LAN5	LAN5 (LXI defined “LAN5” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN5`	`"LAN5”`
		COM/.NET	`N/A`	`"LAN5"`
LAN6	LAN6 (LXI defined “LAN6” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN6`	`"LAN6”`
		COM/.NET	`N/A`	`"LAN6"`
LAN7	LAN7 (LXI defined “LAN6” LAN message)			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LAN7`	`“LAN7”`
		COM/.NET	`N/A`	`"LAN7"`
LXI0	LXI Trigger Bus Line 0			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI0`	`“LXI0”`
		COM/.NET	`N/A`	`“LXI0”`
LXI1	LXI Trigger Bus Line 1			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI1`	`"LXI1”`
		COM/.NET	`N/A`	`"LXI1"`
LXI2	LXI Trigger Bus Line 2			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI2`	`"LXI2”`
		COM/.NET	`N/A`	`"LXI2"`
LXI3	LXI Trigger Bus Line 3			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI3`	`"LXI3”`
		COM/.NET	`N/A`	`"LXI3"`

Name	*Description*			
		Language	*Identifier*	*Actual Value*
LXI4	LXI Trigger Bus Line 4			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI4`	`"LXI4”`
		COM/.NET	`N/A`	`"LXI4"`
LXI5	LXI Trigger Bus Line 5			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI5`	`"LXI5"`
		COM/.NET	`N/A`	`"LXI5"`
LXI6	LXI Trigger Bus Line 6			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI6`	`"LXI6"`
		COM/.NET	`N/A`	`"LXI6"`
LXI7	LXI Trigger Bus Line 7			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_LXI7`	`"LXI7"`
		COM/.NET	`N/A`	`"LXI7"`
TTL0	TTL Interface 0			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL0`	`"TTL0"`
		COM/.NET	`N/A`	`"TTL0"`
TTL1	TTL Interface 1			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL1`	`"TTL1"`
		COM/.NET	`N/A`	`"TTL1"`
TTL2	TTL Interface 2			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL2`	`"TTL2"`
		COM/.NET	`N/A`	`"TTL2"`
TTL3	TTL Interface 3			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL3`	`"TTL3"`
		COM/.NET	`N/A`	`"TTL3"`
TTL4	TTL Interface 4			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL4`	`"TTL4"`

Name	Description			
		Language	Identifier	Actual Value
		COM/.NET	`N/A`	`"TTL4"`
TTL5	TTL Interface 5			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL5`	`"TTL5"`
		COM/.NET	`N/A`	`"TTL5"`
TTL6	TTL Interface 6			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL6`	`"TTL6"`
		COM/.NET	`N/A`	`"TTL6"`
TTL7	TTL Interface 7			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_TTL7`	`"TTL7"`
		COM/.NET	`N/A`	`"TTL7"`
PXI_STAR	PXI Star Interface			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_STAR`	`"PXI_STAR"`
		COM/.NET	`N/A`	`"PXI_STAR"`
PXI_TRIG0	PXI Trigger Bus Line 0			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG0`	`"PXI_TRIG0"`
		COM/.NET	`N/A`	`"PXI_TRIG0"`
PXI_TRIG1	PXI Trigger Bus Line 1			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG1`	`"PXI_TRIG1"`
		COM/.NET	`N/A`	`"PXI_TRIG1"`
PXI_TRIG2	PXI Trigger Bus Line 2			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG2`	`"PXI_TRIG2"`
		COM/.NET	`N/A`	`"PXI_TRIG2"`
PXI_TRIG3	PXI Trigger Bus Line 3			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG3`	`"PXI_TRIG3"`

Name	*Description*			
		Language	*Identifier*	*Actual Value*
		COM/.NET	`N/A`	`"PXI_TRIG3"`
PXI_TRIG4	PXI Trigger Bus Line 4			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG4`	`"PXI_TRIG4"`
		COM/.NET	`N/A`	`"PXI_TRIG4"`
PXI_TRIG5	PXI Trigger Bus Line 5			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG5`	`"PXI_TRIG5"`
		COM/.NET	`N/A`	`"PXI_TRIG5"`
PXI_TRIG6	PXI Trigger Bus Line 6			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG6`	`"PXI_TRIG6"`
		COM/.NET	`N/A`	`"PXI_TRIG6"`
PXI_TRIG7	PXI Trigger Bus Line 7			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXI_TRIG7`	`"PXI_TRIG7"`
		COM/.NET	`N/A`	`"PXI_TRIG7"`
PXIe_DSTARA	PXI Express DStar Line A			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXIE_DSTARA`	`"PXIe_DSTARA"`
		COM/.NET	`N/A`	`"PXIe_DSTARA"`
PXIe_DSTARB	PXI Express DStar Line B			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXIE_DSTARB`	`"PXIe_DSTARB"`
		COM/.NET	`N/A`	`"PXIe_DSTARB"`
PXIe_DSTARC	PXI Express DStar Line C			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_PXIE_DSTARC`	`"PXIe_DSTARC"`
		COM/.NET	`N/A`	`"PXIe_DSTARC"`
RTSI0	RTSI Bus Line 0			

Name	*Description*			
		Language	*Identifier*	*Actual Value*
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI0`	`"RTSI0"`
		COM/.NET	`N/A`	`"RTSI0"`
RTSI1	RTSI Bus Line 1			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI1`	`"RTSI1"`
		COM/.NET	`N/A`	`"RTSI1"`
RTSI2	RTSI Bus Line 2			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI2`	`"RTSI2"`
		COM/.NET	`N/A`	`"RTSI2"`
RTSI3	RTSI Bus Line 3			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI3`	`"RTSI3"`
		COM/.NET	`N/A`	`"RTSI3"`
RTSI4	RTSI Bus Line 4			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI4`	`"RTSI4"`
		COM/.NET	`N/A`	`"RTSI4"`
RTSI5	RTSI Bus Line 5			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI5`	`"RTSI5"`
		COM/.NET	`N/A`	`"RTSI5"`
RTSI6	RTSI Bus Line 6			
		C	`IVIUPCONVERTER_VAL_TRIGGER_SOURCE_RTSI6`	`"RTSI6"`
		COM/.NET	`N/A`	`"RTSI6"`

PM External Coupling

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_AC`	0

Value Name	*Language*	*Identifier*	*Actual Value*
	COM	`IviUpconverterPMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterPMExternalCouplingDC`	1
PM External Coupling Class Ext Base	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE`	100
PM External Coupling Specific Ext Base	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

Pulse Modulation External Polarity

Value Name	*Language*	*Identifier*	*Actual Value*
Normal	.NET	`PulseModulationExternalPolarity.Normal`	0
	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL`	0
	COM	`IviUpconverterPulseModulationExternalPolarityNormal`	0
Inverse	.NET	`PulseModulationExternalPolarity.Inverse`	1
	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE`	1
	COM	`IviUpconverterPulseModulationExternalPolarityInverse`	1
Pulse Modulation External Polarity Class Ext Base	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE`	100
Pulse Modulation External Polarity Specific Ext Base	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

Reference Oscillator Source

Value Name	*Language*	*Identifier*	*Actual Value*
Internal	.NET	`ReferenceOscillatorSource.Internal`	0
	C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL`	0

Value Name	*Language*	*Identifier*	*Actual Value*
	COM	`IviUpconverterReferenceOscillatorSourceInternal`	0
External	.NET	`ReferenceOscillatorSource.External`	1
	C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL`	1
	COM	`IviUpconverterReferenceOscillatorSourceExternal`	1
Reference Oscillator Source Class Ext Base	C	`IVIUPCONVERTER_VAL_REFERENCE_ OSCILLATOR_SOURCE_CLASS_EXT_BASE`	100
Reference Oscillator Source Specific Ext Base	C	`IVIUPCONVERTER_VAL_REFERENCE_ OSCILLATOR_SOURCE_SPECIFIC_EXT_BASE`	1000
	COM	`N/A`	1000

Sweep Mode

Value Name	*Language*	*Identifier*	*Actual Value*
None	.NET	`SweepMode.None`	0
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_NONE`	0
	COM	`IviUpconverterSweepModeNone`	0
Frequency Sweep	.NET	`SweepMode.FrequencySweep`	1
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_SWEEP`	1
	COM	`IviUpconverterSweepModeFrequencySweep`	1
Power Sweep	.NET	`SweepMode.PowerSweep`	2
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_SWEEP`	2
	COM	`IviUpconverterSweepModePowerSweep`	2
Gain Sweep	.NET	`SweepMode.GainSweep`	3
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_SWEEP`	3
	COM	`IviUpconverterSweepModeGainSweep`	3
Frequency Step	.NET	`SweepMode.FrequencyStep`	4
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_STEP`	4
	COM	`IviUpconverterSweepModeFrequencyStep`	4
Power Step	.NET	`SweepMode.PowerStep`	5

Value Name	*Language*	*Identifier*	*Actual Value*
	C	IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_STEP	5
	COM	IviUpconverterSweepModePowerStep	5
Gain Step	.NET	SweepMode.GainStep	6
	C	IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_STEP	6
	COM	IviUpconverterSweepModeGainStep	6
List	.NET	SweepMode.List	7
	C	IVIUPCONVERTER_VAL_SWEEP_MODE_LIST	7
	COM	IviUpconverterSweepModeList	7
Sweep Mode Class Ext Base	C	IVIUPCONVERTER_VAL_SWEEP_MODE_ CLASS_EXT_BASE	100
Sweep Mode Specific Ext Base	C	IVIUPCONVERTER_VAL_SWEEP_MODE_ SPECIFIC_EXT_BASE	1000
	COM	N/A	1000

31. IviUpconverter Function Parameter Value Definitions

This section specifies the actual values for each function parameter that defines values.

Configure ALC

Parameter: `Source`

Value Name	*Language*	*Identifier*	*Actual Value*
Internal	.NET	`AlcSource.Internal`	0
	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_INTERNAL`	0
	COM	`IviUpconverterALCSourceInternal`	0
External	.NET	`AlcSource.External`	1
	C	`IVIUPCONVERTER_VAL_ALC_SOURCE_EXTERNAL`	1
	COM	`IviUpconverterALCSourceExternal`	1

Configure AM

Parameter: `Scaling`

Value Name	*Language*	*Identifier*	*Actual Value*
Linear	.NET	`Scaling.Linear`	0
	C	`IVIUPCONVERTER_VAL_AM_SCALING_LINEAR`	0
	COM	`IviUpconverterAMScalingLinear`	0
Logarithmic	.NET	`Scaling.Logarithmic`	1
	C	`IVIUPCONVERTER_VAL_AM_SCALING_LOGARITHMIC`	1
	COM	`IviUpconverterAMScalingLogarithmic`	1

Configure AM External Coupling

Parameter: `Coupling`

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_AC`	0
	COM	`IviUpconverterAMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_AM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterAMExternalCouplingDC`	1

Configure FM External Coupling

Parameter: `Coupling`

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_AC`	0
	COM	`IviUpconverterFMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_FM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterFMExternalCouplingDC`	1

Configure Frequency Step Start Stop

Parameter: `Scaling`

Value Name	*Language*	*Identifier*	*Actual Value*
Linear	.NET	`Scaling.Linear`	0
	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LINEAR`	0
	COM	`IviUpconverterFrequencyStepScalingLinear`	0
Logarithmic	.NET	`Scaling.Logarithmic`	1
	C	`IVIUPCONVERTER_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC`	1
	COM	`IviUpconverterFrequencyStepScalingLogarithmic`	1

Configure PM External Coupling

Parameter: `Coupling`

Value Name	*Language*	*Identifier*	*Actual Value*
AC	.NET	`ExternalCoupling.AC`	0
	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_AC`	0
	COM	`IviUpconverterPMExternalCouplingAC`	0
DC	.NET	`ExternalCoupling.DC`	1
	C	`IVIUPCONVERTER_VAL_PM_EXTERNAL_COUPLING_DC`	1
	COM	`IviUpconverterPMExternalCouplingDC`	1

Configure Pulse Modulation External Polarity

Parameter: `Polarity`

Value Name	*Language*	*Identifier*	*Actual Value*
Normal	.NET	`PulseModulationExternalPolarity.Normal`	0
	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL`	0
	COM	`IviUpconverterPulseModulationExternalPolarityNormal`	0
Inverse	.NET	`PulseModulationExternalPolarity.Inverse`	1
	C	`IVIUPCONVERTER_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE`	1
	COM	`IviUpconverterPulseModulationExternalPolarityInverse`	1

Configure Reference Oscillator

Parameter: `Source`

Value Name	*Language*	*Identifier*	*Actual Value*
Internal	.NET	`ReferenceOscillatorSource.Internal`	0
	C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL`	0
	COM	`IviUpconverterReferenceOscillatorSourceInternal`	0
External	.NET	`ReferenceOscillatorSource.External`	1
	C	`IVIUPCONVERTER_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL`	1
	COM	`IviUpconverterReferenceOscillatorSourceExternal`	1

Configure Sweep

Parameter: `Mode`

Value Name	*Language*	*Identifier*	*Actual Value*
None	.NET	`SweepMode.None`	0
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_NONE`	0
	COM	`IviUpconverterSweepModeNone`	0
Frequency Sweep	.NET	`SweepMode.FrequencySweep`	1
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_SWEEP`	1
	COM	`IviUpconverterSweepModeFrequencySweep`	1
Power Sweep	.NET	`SweepMode.PowerSweep`	2
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_SWEEP`	2
	COM	`IviUpconverterSweepModePowerSweep`	2
Gain Sweep	.NET	`SweepMode.GainSweep`	3
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_SWEEP`	3
	COM	`IviUpconverterSweepModeGainSweep`	3
Frequency Step	.NET	`SweepMode.FrequencyStep`	4
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_FREQUENCY_STEP`	4
	COM	`IviUpconverterSweepModeFrequencyStep`	4
Power Step	.NET	`SweepMode.PowerStep`	5
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_POWER_STEP`	5
	COM	`IviUpconverterSweepModePowerStep`	5
Gain Step	.NET	`SweepMode.GainStep`	6
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_GAIN_STEP`	6
	COM	`IviUpconverterSweepModeGainStep`	6
List	.NET	`SweepMode.List`	7
	C	`IVIUPCONVERTER_VAL_SWEEP_MODE_LIST`	7
	COM	`IviUpconverterSweepModeList`	7

Is Calibration Complete

Parameter: `Status`

Value Name	*Language*	*Identifier*	*Actual Value*
Calibration Complete	.NET	`CalibrationStatus.Complete`	0
	C	`IVIUPCONVERTER_VAL_CALIBRATION_COMPLETE`	0
	COM	`IviUpconverterCalibrationComplete`	0
Calibration In Progress	.NET	`CalibrationStatus.InProgress`	1
	C	`IVIUPCONVERTER_VAL_CALIBRATION_IN_PROGRESS`	1
	COM	`IviUpconverterCalibrationInProgress`	1
Calibration Status Unknown	.NET	`CalibrationStatus.StatusUnknown`	2
	C	`IVIUPCONVERTER_VAL_CALIBRATION_STATUS_UNKNOWN`	2
	COM	`IviUpconverterCalibrationStatusUnknown`	2
Calibration Failed	.NET	CalibrationStatus`.Failed`	3
	C	`IVIUPCONVERTER_VAL_CALIBRATION_FAILED`	3
	COM	`IviUpconverterCalibrationFailed`	3

Wait Until Ready

Parameter: `MaxTimeMilliseconds`

Value Name	Language	Identifier	Actual Value
Max Time Immediate	C	`IVIUPCONVERTER_VAL_MAX_TIME_IMMEDIATE`	0x0
	COM	`IviUpconverterTimeOutImmediate`	0x0
Max Time Infinite	C	`IVIUPCONVERTER_VAL_MAX_TIME_INFINITE`	0xFFFFFFFFUL
	COM	`IviDownconverterTimeOutInfinite`	0xFFFFFFFFUL

Wait Until Ready

Parameter: `maximumTime`

Value Name	Language	Identifier	Actua l Value
Max Time Immediate	.NET	`PrecisionTimeSpan.Zero`	
Max Time Infinite	.NET	`PrecisionTimeSpan.MaxValue`	

32.IviUpconverter Error and Completion Code Value Definitions

The table below specifies the actual value for each status code that the IviUpconverter class specification defines.

Table 32-1. IviUpconverter Error and Completion Codes

Error Name	*Description*		
	Language	*Identifier*	*Value(hex)*
Trigger Not Software	The trigger source is not set to software trigger.		
	.NET	`Ivi.Driver.TriggerNotSoftwareException`	`IVI Defined Exception (See IVI-3.2)`
	C	`IVIUPCONVERTER_ERROR_TRIGGER_NOT_SOFTWARE`	`0xBFFA1001`
	COM	`E_IVIUPCONVERTER_TRIGGER_NOT_SOFTWARE`	`0x80041001`
Max Time Exceeded	Maximum time exceeded before the operation completed.		
	.NET	`Ivi.Driver.MaxTimeExceededException`	`IVI Defined Exception (See IVI-3.2)`
	C	`IVIUPCONVERTER_ERROR_MAX_TIME_EXCEEDED`	`0xBFFA2001`
	COM	`E_IVIUPCONVERTER_MAX_TIME_EXCEEDED`	`0x80042001`
List Unknown	The selected list is not defined.		
	.NET	`ListUnknownException`	
	C	`IVIUPCONVERTER_ERROR_LIST_UNKNOWN`	`0xBFFA2002`
	COM	`E_IVIUPCONVERTER_LIST_UNKNOWN`	`0x80042002`

Table 322 defines the recommended format of the message string associated with the errors. In C, these strings are returned by the Get Error function. In COM, these strings are the description contained in the ErrorInfo object. For .NET, exception default message strings are defined with the exception.

Note: In the description string table entries listed below, **{0}** is always used to represent the component name.

Table 32-2. IviUpconverter Error Message Strings

Name	Message String
Trigger Not Software	"{0}: Trigger source is not set to software trigger."
Max Time Exceeded	"{0}: Maximum time exceeded before the operation completed."
List Unknown	"{0}: The selected list is not defined."

32.1.IVI.NET IviUpconverter Exceptions and Warnings

This section defines the list of IVI.NET exceptions and warnings that are specific to the IviUpconverter class. For general information on IVI.NET exceptions and warnings, refer to *IVI-3.1: Driver Architecture Specification* and section 12, *Common IVI.NET Exceptions and Warnings*, of *IVI-3.2: Inherent Capabilities Specification*.

The IVI.NET exceptions defined in this specification are declared in the Ivi.Upconverter namespace.

- ListUnknownException

32.1.1.ListUnknownException

Description

This exception is used when the driver finds that a specified list is not defined.

Constructors

```
Ivi.Upconverter.FrequencyListUnknownException(String message,
                                              String listName);

Ivi.Upconverter.FrequencyListUnknownException();

Ivi.Upconverter.FrequencyListUnknownException(String message);

Ivi.Upconverter.FrequencyListUnknownException(String message,
                                              System.Exception innerException);
```

Message String

```
The specified list is not defined.
List name: <listName>
```

Parameters

Inputs	Description	Base Type
`listName`	The name of the list that is undefined.	`String`

Usage

If driver developers use constructors that take a message string, they are responsible for message string localization.

33.IviUpconverter Hierarchies

33.1.IviUpconverter .NET Hierarchy

The full IviUpconverter .NET Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.1, *.NET Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, it is omitted from Table 26-1.

Table 33-1. IviUpconverter .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
SendSoftwareTrigger	Send Software Trigger	M
Alc		
Configure	Configure ALC	M
Bandwidth	ALC Bandwidth	P
Enabled	ALC Enabled	P
Source	ALC Source	P
AnalogModulation		
AM		
Configure	Configure AM	M
Depth	AM Depth	P
Enabled	AM Enabled	P
ExternalCoupling	AM External Coupling	P
NominalVoltage	AM Nominal Voltage	P
Scaling	AM Scaling	P
Source	AM Source	P
FM		
Configure	Configure FM	M
Deviation	FM Deviation	P
Enabled	FM Enabled	P
ExternalCoupling	FM External Coupling	P
NominalVoltage	FM Nominal Voltage	P
Source	FM Source	P
PM		
Configure	Configure PM	M
Deviation	PM Deviation	P
Enabled	PM Enabled	P
ExternalCoupling	PM External Coupling	P

Table 33-1. IviUpconverter .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
NominalVoltage	PM Nominal Voltage	P
Source	PM Source	P
Source		
Count	Analog Modulation Source Count	P
GetName	Get Analog Modulation Source Name	P
ExternalLO		
Enabled	External LO Enabled	P
Frequency	External LO Frequency	P
IFInput		
Attenuation	IF Input Attenuation	P
AutoCorrectionsEnabled	Auto Corrections Enabled	P
Bypass	Bypass	P
Coupling	IF Input Coupling	P
Frequency	IF Input Frequency	P
ActiveIFInput	Active IF Input	P
Count	IF Input Count	P
GetName	Get IF Input Name	P
IQ		
Calibrate	Calibrate IQ	M
Enabled	IQ Enabled	P
NominalVoltage	IQ Nominal Voltage	P
SwapEnabled	IQ Swap Enabled	P
Impairment		
Configure	Configure IQ Impairment	M
Enabled	IQ Impairment Enabled	P
IOffset	IQ Impairment I Offset	P
QOffset	IQ Impairment Q Offset	P
Ratio	IQ Impairment Ratio	P
Skew	IQ Impairment Skew	P
PulseModulation		
Enabled	Pulse Modulation Enabled	P
ExternalPolarity	Pulse Modulation External Polarity	P

Table 33-1. IviUpconverter .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
ReferenceOscillator		
Configure	Configure Reference Oscillator	M
ExternalFrequency	Reference Oscillator External Frequency	P
OutputEnabled	Reference Oscillator Output Enabled	P
Source	Reference Oscillator Source	P
RFOutput		
WaitUntilReady	Wait Until Ready	M
Calibrate	Calibrate	M
IsCalibrationComplete	Is Calibration Complete	M
Bandwidth	RF Output Bandwidth	P
Enabled	RF Output Enabled	P
Frequency	RF Output Frequency	P
Gain	RF Output Gain	P
IsReady	Is Ready	P
Level	RF Output Level	P
ReadyTrigger	RF Output Ready Trigger	P
AttenuatorHoldEnabled	Attenuator Hold Enabled	P
ActiveRFOutput	Active RF Output	P
Count	RF Output Count	P
GetName	Get RF Output Name	P
Sweep		
Configure	Configure Sweep	M
IsSweeping	Is Sweeping	P
Mode	Sweep Mode	P
TriggerSource	Sweep Trigger Source	P
FrequencyStep		
ConfigureDwell	Configure Frequency Step Dwell	M
ConfigureStartStop	Configure Frequency Step Start Stop	M
Reset	Reset Frequency Step	M
Dwell	Frequency Step Dwell	P
Scaling	Frequency Step Scaling	P
SingleStepEnabled	Frequency Step Single Step Enabled	P

Table 33-1. IviUpconverter .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Size	Frequency Step Size	P
Start	Frequency Step Start	P
Stop	Frequency Step Stop	P
FrequencySweep		
ConfigureCenterSpan	Configure Frequency Sweep Center Span	M
ConfigureStartStop	Configure Frequency Sweep Start Stop	M
Start	Frequency Sweep Start	P
Stop	Frequency Sweep Stop	P
Time	Frequency Sweep Time	P
List		
ClearAll	Clear All Lists	M
ConfigureDwell	Configure List Dwell	M
CreateFrequency	Create Frequency List	M
CreateFrequencyPower	Create Frequency Power List	M
CreateFrequencyGain	Create Frequency Gain List	M
CreatePower	Create Power List	M
CreateGain	Create Gain List	M
Reset	Reset List	M
Dwell	List Dwell	P
SelectedName	List Selected Name	P
SingleStepEnabled	List Single Step Enabled	P
PowerStep		
ConfigureDwell	Configure Power Step Dwell	M
ConfigureStartStop	Configure Power Step Start Stop	M
Reset	Reset Power Step	M
Dwell	Power Step Dwell	P
SingleStepEnabled	Power Step Single Step Enabled	P
Size	Power Step Size	P
Start	Power Step Start	P
Stop	Power Step Stop	P
GainStep		
ConfigureDwell	Configure Gain Step Dwell	M

Table 33-1. IviUpconverter .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
ConfigureStartStop	Configure Gain Step Start Stop	M
Reset	Reset Gain Step	M
Dwell	Gain Step Dwell	P
SingleStepEnabled	Gain Step Single Step Enabled	P
Size	Gain Step Size	P
Start	Gain Step Start	P
Stop	Gain Step Stop	P
PowerSweep		
ConfigureStartStop	Configure Power Sweep Start Stop	M
Start	Power Sweep Start	P
Stop	Power Sweep Stop	P
Time	Power Sweep Time	P
GainSweep		
ConfigureStartStop	Configure Gain Sweep Start Stop	M
Start	Gain Sweep Start	P
Stop	Gain Sweep Stop	P
Time	Gain Sweep Time	P

33.1.1.IviUpconverter .NET Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviUpconverter-interfaces contain interface reference properties for accessing the following IviUpconverter interfaces:

- IIviUpconverterReferenceOscillator
- IIviUpconverterIFInput
- IIviUpconverterRFOutput
- IIviUpconverterExternalLO
- IIviUpconverterAlc
- IIviUpconverterIQ
- IIviUpconverterAnalogModulation
- IIviUpconverterPulseModulation
- IIviUpconverterSweep
- IIviUpconverterAnalogModulationSource

The IIviUpconverterIQ interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterIQImpairment

The IIviUpconverterAnalogModulation interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterAM
- IIviUpconverterFM
- IIviUpconverterPM
- IIviUpconverterAnalogModulationSource

The IIviUpconverterSweep interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterFrequencySweep
- IIviUpconverterPowerSweep
- IIviUpconverterGainSweep
- IIviUpconverterFrequencyStep
- IIviUpconverterPowerStep
- IIviUpconverterGainStep
- IIviUpconverterList

33.1.2..NET Interface Reference Properties

Interface reference properties are used to navigate the IviUpconverter COM hierarchy. This section describes the interface reference properties that the IviUpconverter interfaces define. All interface reference properties are read-only.

Table 33-2. **IviUpconverter .NET Interface Reference Properties**

Interface	Interface Reference Propertyu
`IIviUpconverterAlc`	`Alc`
`IIviUpconverterAM`	`AnanlogModulation.AM`

Table 33-2. **IviUpconverter .NET Interface Reference Properties**

Interface	Interface Reference Propertyu
IIviUpconverterAnalogModulation	AnanlogModulation
IIviUpconverterAnalogModulationSource	AnanlogModulation.Source
IIviUpconverterExternalLO	ExternalLO
IIviUpconverterFM	AnanlogModulation.FM
IIviUpconverterFrequencyStep	Sweep.FrequencyStep
IIviUpconverterFrequencySweep	Sweep.FrequencySweep
IIviUpconverterGainStep	Sweep.GainStep
IIviUpconverterGainSweep	Sweep.GainSweep
IIviUpconverterIFInput	IFInput
IIviUpconverterIQ	IQ
IIviUpconverterIQImpairment	IQ.Impairment
IIviUpconverterList	Sweep.List
IIviUpconverterPM	AnanlogModulation.PM
IIviUpconverterPowerStep	Sweep.PowerStep
IIviUpconverterPowerSweep	Sweep.PowerSweep
IIviUpconverterPulseModulation	PulseModulation
IIviUpconverterReferenceOscillator	ReferenceOscillator
IIviUpconverterRFOutput	RFOutput
IIviUpconverterSweep	Sweep

33.2.IviUpconverter COM Hierarchy

The full IviUpconverter COM Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *COM Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, it is omitted from Table 26-1.

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
SendSoftwareTrigger	Send Software Trigger	M
ALC		
Configure	Configure ALC	M
Bandwidth	ALC Bandwidth	P
Enabled	ALC Enabled	P
Source	ALC Source	P
AnalogModulation		

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
AM		
Configure	Configure AM	M
Depth	AM Depth	P
Enabled	AM Enabled	P
ExternalCoupling	AM External Coupling	P
NominalVoltage	AM Nominal Voltage	P
Scaling	AM Scaling	P
Source	AM Source	P
FM		
Configure	Configure FM	M
Deviation	FM Deviation	P
Enabled	FM Enabled	P
ExternalCoupling	FM External Coupling	P
NominalVoltage	FM Nominal Voltage	P
Source	FM Source	P
PM		
Configure	Configure PM	M
Deviation	PM Deviation	P
Enabled	PM Enabled	P
ExternalCoupling	PM External Coupling	P
NominalVoltage	PM Nominal Voltage	P
Source	PM Source	P
Source		
Count	Analog Modulation Source Count	P
Name	Analog Modulation Source Name	P
ExternalLO		
Enabled	External LO Enabled	P
Frequency	External LO Frequency	P
IFInput		
Attenuation	IF Input Attenuation	P
AutoCorrectionsEnabled	Auto Corrections Enabled	P
Bypass	Bypass	P

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Coupling	IF Input Coupling	P
Frequency	IF Input Frequency	P
ActiveIFInput	Active IF Input	P
Count	IF Input Count	P
Name	IF Input Name	P
IQ		
Calibrate	Calibrate IQ	M
Enabled	IQ Enabled	P
NominalVoltage	IQ Nominal Voltage	P
SwapEnabled	IQ Swap Enabled	P
Impairment		
Configure	Configure IQ Impairment	M
Enabled	IQ Impairment Enabled	P
IOffset	IQ Impairment I Offset	P
QOffset	IQ Impairment Q Offset	P
Ratio	IQ Impairment Ratio	P
Skew	IQ Impairment Skew	P
PulseModulation		
Enabled	Pulse Modulation Enabled	P
ExternalPolarity	Pulse Modulation External Polarity	P
ReferenceOscillator		
Configure	Configure Reference Oscillator	M
ExternalFrequency	Reference Oscillator External Frequency	P
OutputEnabled	Reference Oscillator Output Enabled	P
Source	Reference Oscillator Source	P
RFOutput		
WaitUntilReady	Wait Until Ready	M
Calibrate	Calibrate	M
IsCalibrationComplete	Is Calibration Complete	M
Bandwidth	RF Output Bandwidth	P
Enabled	RF Output Enabled	P
Frequency	RF Output Frequency	P

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Gain	RF Output Gain	P
IsReady	Is Ready	P
Level	RF Output Level	P
ReadyTrigger	RF Output Ready Trigger	P
AttenuatorHoldEnabled	Attenuator Hold Enabled	P
ActiveRFOutput	Active RF Output	P
Count	RF Output Count	P
Name	RF Output Name	P
Sweep		
Configure	Configure Sweep	M
IsSweeping	Is Sweeping	P
Mode	Sweep Mode	P
TriggerSource	Sweep Trigger Source	P
FrequencyStep		
ConfigureDwell	Configure Frequency Step Dwell	M
ConfigureStartStop	Configure Frequency Step Start Stop	M
Reset	Reset Frequency Step	M
Dwell	Frequency Step Dwell	P
Scaling	Frequency Step Scaling	P
SingleStepEnabled	Frequency Step Single Step Enabled	P
Size	Frequency Step Size	P
Start	Frequency Step Start	P
Stop	Frequency Step Stop	P
FrequencySweep		
ConfigureCenterSpan	Configure Frequency Sweep Center Span	M
ConfigureStartStop	Configure Frequency Sweep Start Stop	M
Start	Frequency Sweep Start	P
Stop	Frequency Sweep Stop	P
Time	Frequency Sweep Time	P
List		
ClearAll	Clear All Lists	M
ConfigureDwell	Configure List Dwell	M

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
CreateFrequency	Create Frequency List	M
CreateFrequencyPower	Create Frequency Power List	M
CreateFrequencyGain	Create Frequency Gain List	M
CreatePower	Create Power List	M
CreateGain	Create Gain List	M
Reset	Reset List	M
Dwell	List Dwell	P
SelectedName	List Selected Name	P
SingleStepEnabled	List Single Step Enabled	P
PowerStep		
ConfigureDwell	Configure Power Step Dwell	M
ConfigureStartStop	Configure Power Step Start Stop	M
Reset	Reset Power Step	M
Dwell	Power Step Dwell	P
SingleStepEnabled	Power Step Single Step Enabled	P
Size	Power Step Size	P
Start	Power Step Start	P
Stop	Power Step Stop	P
GainStep		
ConfigureDwell	Configure Gain Step Dwell	M
ConfigureStartStop	Configure Gain Step Start Stop	M
Reset	Reset Gain Step	M
Dwell	Gain Step Dwell	P
SingleStepEnabled	Gain Step Single Step Enabled	P
Size	Gain Step Size	P
Start	Gain Step Start	P
Stop	Gain Step Stop	P
PowerSweep		
ConfigureStartStop	Configure Power Sweep Start Stop	M
Start	Power Sweep Start	P
Stop	Power Sweep Stop	P
Time	Power Sweep Time	P

Table 33-3. IviUpconverter COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
GainSweep		
ConfigureStartStop	Configure Gain Sweep Start Stop	M
Start	Gain Sweep Start	P
Stop	Gain Sweep Stop	P
Time	Gain Sweep Time	P

33.2.1.IviUpconverter COM Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviUpconverter-interfaces contain interface reference properties for accessing the following IviUpconverter interfaces:

- IIviUpconverterReferenceOscillator
- IIviUpconverterIFInput
- IIviUpconverterRFOutput
- IIviUpconverterExternalLO
- IIviUpconverterALC
- IIviUpconverterIQ
- IIviUpconverterAnalogModulation
- IIviUpconverterPulseModulation
- IIviUpconverterSweep
- IIviUpconverterAnalogModulationSource

The IIviUpconverterIQ interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterIQImpairment

The IIviUpconverterAnalogModulation interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterAM
- IIviUpconverterFM
- IIviUpconverterPM
- IIviUpconverterAnalogModulationSource

The IIviUpconverterSweep interface contains interface reference properties for accessing additional the following additional IviUpconverter interface(s):

- IIviUpconverterFrequencySweep
- IIviUpconverterPowerSweep
- IIviUpconverterGainSweep
- IIviUpconverterFrequencyStep
- IIviUpconverterPowerStep
- IIviUpconverterGainStep
- IIviUpconverterList

Table 332. IviUpconverter lists the interfaces that this specification defines and their GUIDs.

Table 33-4. IviUpconverter Interface GUIDs

Interface	GUID
IIviUpconverter	{47ed53dc-a398-11d4-ba58-000064657374}
IIviUpconverterALC	{47ed53dd-a398-11d4-ba58-000064657374}
IIviUpconverterAM	{47ed53de-a398-11d4-ba58-000064657374}
IIviUpconverterAnalogModulation	{47ed53df-a398-11d4-ba58-000064657374}

Table 33-4. IviUpconverter Interface GUIDs

Interface	GUID
IIviUpconverterAnalogModulationSource	{47ed53e0-a398-11d4-ba58-000064657374}
IIviUpconverterExternalLO	{47ed53e1-a398-11d4-ba58-000064657374}
IIviUpconverterFM	{47ed53e2-a398-11d4-ba58-000064657374}
IIviUpconverterFrequencyStep	{47ed53e3-a398-11d4-ba58-000064657374}
IIviUpconverterFrequencySweep	{47ed53e4-a398-11d4-ba58-000064657374}
IIviUpconverterGainStep	{47ed53f0-a398-11d4-ba58-000064657374}
IIviUpconverterGainSweep	{47ed53f1-a398-11d4-ba58-000064657374}
IIviUpconverterIFInput	{47ed53e5-a398-11d4-ba58-000064657374}
IIviUpconverterIQ	{47ed53e6-a398-11d4-ba58-000064657374}
IIviUpconverterIQImpairment	{47ed53e7-a398-11d4-ba58-000064657374}
IIviUpconverterList	{47ed53e8-a398-11d4-ba58-000064657374}
IIviUpconverterPM	{47ed53e9-a398-11d4-ba58-000064657374}
IIviUpconverterPowerStep	{47ed53ea-a398-11d4-ba58-000064657374}
IIviUpconverterPowerSweep	{47ed53eb-a398-11d4-ba58-000064657374}
IIviUpconverterPulseModulation	{47ed53ec-a398-11d4-ba58-000064657374}
IIviUpconverterReferenceOscillator	{47ed53ed-a398-11d4-ba58-000064657374}
IIviUpconverterRFOutput	{47ed53ee-a398-11d4-ba58-000064657374}
IIviUpconverterSweep	{47ed53ef-a398-11d4-ba58-000064657374}

33.2.2.COM Interface Reference Properties

Interface reference properties are used to navigate the IviUpconverter COM hierarchy. This section describes the interface reference properties that the IviUpconverter interfaces define. All interface reference properties are read-only.

Table 33-5. IviUpconverter COM Interface Reference Properties

Interface	Interface Reference Propertyu
IIviUpconverterALC	ALC
IIviUpconverterAM	AnanlogModulation.AM
IIviUpconverterAnalogModulation	AnanlogModulation
IIviUpconverterAnalogModulationSource	AnanlogModulation.Source
IIviUpconverterExternalLO	ExternalLO
IIviUpconverterFM	AnanlogModulation.FM
IIviUpconverterFrequencyStep	Sweep.FrequencyStep
IIviUpconverterFrequencySweep	Sweep.FrequencySweep
IIviUpconverterGainStep	Sweep.GainStep
IIviUpconverterGainSweep	Sweep.GainSweep
IIviUpconverterIFInput	IFInput
IIviUpconverterIQ	IQ
IIviUpconverterIQImpairment	IQ.Impairment
IIviUpconverterList	Sweep.List
IIviUpconverterPM	AnanlogModulation.PM
IIviUpconverterPowerStep	Sweep.PowerStep
IIviUpconverterPowerSweep	Sweep.PowerSweep
IIviUpconverterPulseModulation	PulseModulation
IIviUpconverterReferenceOscillator	ReferenceOscillator
IIviUpconverterRFOutput	RFOutput
IIviUpconverterSweep	Sweep

33.2.3.IviUpconverter COM Category

The IviUpconverter class COM Category shall be “IviUpconverter”, and the Category ID (CATID) shall be {47ed515f-a398-11d4-ba58-000064657374}.

33.3.IviUpconverter C Function Hierarchy

The IviUpconverter class function hierarchy is shown in the following table.

Table 33-6. IviUpconverter Function Hierarchy

Name or Class	Function Name
ALC…	
Configure ALC	`IviUpconverter_ConfigureALC`
Analog Modulation…	
AM…	
Configure AM	`IviUpconverter_ConfigureAM`
Configure AM Enabled	`IviUpconverter_ConfigureAMEnabled`
Configure AM External Coupling	`IviUpconverter_ConfigureAMExternalCouplin g`
FM…	
Configure FM	`IviUpconverter_ConfigureFM`
Configure FM Enabled	`IviUpconverter_ConfigureFMEnabled`
Configure FM External Coupling	`IviUpconverter_ConfigureFMExternalCouplin g`
PM…	
Configure PM	`IviUpconverter_ConfigurePM`
Configure PM Enabled	`IviUpconverter_ConfigurePMEnabled`
Configure PM External Coupling	`IviUpconverter_ConfigurePMExternalCouplin g`
Source…	
Get Analog Modulation Source Name	`IviUpconverter_GetAnalogModulationSourceN ame`
Attribute Accessors…	
GetAttributeViBoolean	`IviUpconverter_GetAttributeViBoolean`
GetAttributeViInt32	`IviUpconverter_GetAttributeViInt32`
GetAttributeViReal64	`IviUpconverter_GetAttributeViReal64`
GetAttributeViSession	`IviUpconverter_GetAttributeViSession`
GetAttributeViString	`IviUpconverter_GetAttributeViString`
SetAttributeViBoolean	`IviUpconverter_SetAttributeViBoolean`
SetAttributeViInt32	`IviUpconverter_SetAttributeViInt32`
SetAttributeViReal64	`IviUpconverter_SetAttributeViReal64`
SetAttributeViSession	`IviUpconverter_SetAttributeViSession`
SetAttributeViString	`IviUpconverter_SetAttributeViString`
Close	`IviUpconverter_close`

Name or Class	Function Name
IF Input…	
Configure Bypass	`IviUpconverter_ConfigureBypass`
Configure IF Input Attenuation	`IviUpconverter_ConfigureIFInputAttenuation`
Get IF Input Name	`IviUpconverter_GetIFInputName`
Set Active IF Input	`IviUpconverter_SetActiveIFInput`
Initialize	`IviUpconverter_init`
Initialize With Options	`IviUpconverter_InitWithOptions`
IQ…	
Calibrate IQ	`IviUpconverter_CalibrateIQ`
Configure IQ Enabled	`IviUpconverter_ConfigureIQEnabled`
Impairment…	
Configure IQ Impairment	`IviUpconverter_ConfigureIQImpairment`
Configure IQ Impairment Enabled	`IviUpconverter_ConfigureIQImpairmentEnabled`
Pulse Modulation…	
Configure Pulse Modulation Enabled	`IviUpconverter_ConfigurePulseModulationEnabled`
Configure Pulse Modulation External Polarity	`IviUpconverter_ConfigurePulseModulationExternalPolarity`
Reference Oscillator…	
Configure Reference Oscillator	`IviUpconverter_ConfigureReferenceOscillator`
Configure Reference Oscillator Output Enabled	`IviUpconverter_ConfigureReferenceOscillatorOutputEnabled`
RF Output…	
Configure RF Output Enabled	`IviUpconverter_ConfigureRFOutputEnabled`
Configure RF Output Frequency	`IviUpconverter_ConfigureRFOutputFrequency`
Configure RF Output Gain	`IviUpconverter_ConfigureRFOutputGain`
Configure RF Output Level	`IviUpconverter_ConfigureRFOutputLevel`
Configure RF Output Ready Trigger	`IviUpconverter_ConfigureRFOutputReadyTrigger`
Configure Attenuator Hold Enabled	`IviUpconverter_ConfigureAttenuatorHoldEnabled`
Get RF Output Name	`IviUpconverter_GetRFOutputName`
Set Active RF Output	`IviUpconverter_SetActiveRFOutput`
Sweep…	

Name or Class	Function Name
Configure Sweep	IviUpconverter_ConfigureSweep
Frequency Step…	
Configure Frequency Step Dwell	IviUpconverter_ConfigureFrequencyStepDwell
Configure Frequency Step Start Stop	IviUpconverter_ConfigureFrequencyStepStartStop
Reset Frequency Step	IviUpconverter_ResetFrequencyStep
Frequency Sweep…	
Configure Frequency Sweep Center Span	IviUpconverter_ConfigureFrequencySweepCenterSpan
Configure Frequency Sweep Start Stop	IviUpconverter_ConfigureFrequencySweepStartStop
Configure Frequency Sweep Time	IviUpconverter_ConfigureFrequencySweepTime
List…	
Clear All Lists	IviUpconverter_ClearAllLists
Configure List Dwell	IviUpconverter_ConfigureListDwell
Create Frequency List	IviUpconverter_CreateFrequencyList
Create Frequency Power List	IviUpconverter_CreateFrequencyPowerList
Create Frequency Gain List	IviUpconverter_CreateFrequencyGainList
Create Power List	IviUpconverter_CreatePowerList
Create Gain List	IviUpconverter_CreateGainList
Reset List	IviUpconverter_ResetList
Power Step…	
Configure Power Step Dwell	IviUpconverter_ConfigurePowerStepDwell
Configure Power Step Start Stop	IviUpconverter_ConfigurePowerStepStartStop
Reset Power Step	IviUpconverter_ResetPowerStep
Gain Step…	
Configure Gain Step Dwell	IviUpconverter_ConfigureGainStepDwell
Configure Gain Step Start Stop	IviUpconverter_ConfigureGainStepStartStop
Reset Gain Step	IviUpconverter_ResetGainStep
Power Sweep…	
Configure Power Sweep Start Stop	IviUpconverter_ConfigurePowerSweepStartStop
Configure Power Sweep Time	IviUpconverter_ConfigurePowerSweepTime
Gain Sweep…	

Name or Class	Function Name
Configure Gain Sweep Start Stop	`IviUpconverter_ConfigureGainSweepStartStop`
Configure Gain Sweep Time	`IviUpconverter_ConfigureGainSweepTime`
Action…	
Calibrate	`IviUpconverter_Calibrate`
Is Calibration Complete	`IviUpconverter_IsCalibrationComplete`
Send Software Trigger	`IviUpconverter_SendSoftwareTrigger`
Wait Until Ready	`IviUpconverter_WaitUntilReady`
Utility…	
Clear Error	`IviUpconverter_ClearError`
Clear Interchange Warnings	`IviUpconverter_ClearInterchangeWarnings`
Disable	`IviUpconverter_Disable`
Error Message	`IviUpconverter_error_message`
Error Query	`IviUpconverter_error_query`
Get Error	`IviUpconverter_GetError`
Get Next Coercion Record	`IviUpconverter_GetNextCoercionRecord`
Get Next Interchange Warning	`IviUpconverter_GetNextInterchangeWarning`
Get Specific Driver C Handle	`IviUpconverter_GetSpecificDriverCHandle`
Invalidate All Attributes	`IviUpconverter_InvalidateAllAttributes`
Lock Session	`IviUpconverter_LockSession`
Reset	`IviUpconverter_reset`
Reset Interchange Check	`IviUpconverter_ResetInterchangeCheck`
ResetWithDefaults	`IviUpconverter_ResetWithDefaults`
Revision Query	`IviUpconverter_revision_query`
Self Test	`IviUpconverter_self_test`
Unlock Session	`IviUpconverter_UnlockSession`

33.4.IviUpconverter C Attribute Hierarchy

The IviUpconverter class attribute hierarchy is shown in the following table.

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
ALC	
ALC Bandwidth	IVIUPCONVERTER_ATTR_ALC_BANDWIDTH
ALC Enabled	IVIUPCONVERTER_ATTR_ALC_ENABLED
ALC Source	IVIUPCONVERTER_ATTR_ALC_SOURCE
Analog Modulation	
AM	
AM Depth	IVIUPCONVERTER_ATTR_AM_DEPTH
AM Enabled	IVIUPCONVERTER_ATTR_AM_ENABLED
AM External Coupling	IVIUPCONVERTER_ATTR_AM_EXTERNAL_COUPLING
AM Nominal Voltage	IVIUPCONVERTER_ATTR_AM_NOMINAL_VOLTAGE
AM Scaling	IVIUPCONVERTER_ATTR_AM_SCALING
AM Source	IVIUPCONVERTER_ATTR_AM_SOURCE
FM	
FM Deviation	IVIUPCONVERTER_ATTR_FM_DEVIATION
FM Enabled	IVIUPCONVERTER_ATTR_FM_ENABLED
FM External Coupling	IVIUPCONVERTER_ATTR_FM_EXTERNAL_COUPLING
FM Nominal Voltage	IVIUPCONVERTER_ATTR_FM_NOMINAL_VOLTAGE
FM Source	IVIUPCONVERTER_ATTR_FM_SOURCE
PM	
PM Deviation	IVIUPCONVERTER_ATTR_PM_DEVIATION
PM Enabled	IVIUPCONVERTER_ATTR_PM_ENABLED
PM External Coupling	IVIUPCONVERTER_ATTR_PM_EXTERNAL_COUPLING
PM Nominal Voltage	IVIUPCONVERTER_ATTR_PM_NOMINAL_VOLTAGE
PM Source	IVIUPCONVERTER_ATTR_PM_SOURCE
Source	

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Analog Modulation Source Count	`IVIUPCONVERTER_ATTR_ANALOG_MODULATION_SOURCE_COUNT`
External LO	
External LO Enabled	`IVIUPCONVERTER_ATTR_EXTERNAL_LO_ENABLED`
External LO Frequency	`IVIUPCONVERTER_ATTR_EXTERNAL_LO_FREQUENCY`
IF Input	
Active IF Input	`IVIUPCONVERTER_ATTR_ACTIVE_IF_INPUT`
Auto Corrections Enabled	`IVIUPCONVERTER_ATTR_AUTO_CORRECTIONS_ENABLED`
Bypass	`IVIUPCONVERTER_ATTR_BYPASS`
IF Input Attenuation	`IVIUPCONVERTER_ATTR_IF_INPUT_ATTENUATION`
IF Input Count	`IVIUPCONVERTER_ATTR_IF_INPUT_COUNT`
IF Input Coupling	`IVIUPCONVERTER_ATTR_IF_INPUT_COUPLING`
IF Input Frequency	`IVIUPCONVERTER_ATTR_IF_INPUT_FREQUENCY`
Inherent IVI Attributes	
Advanced Session Information	
Driver Setup	`IVIUPCONVERTER_ATTR_DRIVER_SETUP`
I/O Resource Descriptor	`IVIUPCONVERTER_ATTR_IO_RESOURCE_DESCRIPTOR`
Logical Name	`IVIUPCONVERTER_ATTR_LOGICAL_NAME`
Class Driver Identification	
Class Driver Class Spec Major Version	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_CLASS_SPEC_MAJOR_VERSION`
Class Driver Class Spec Minor Version	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_CLASS_SPEC_MINOR_VERSION`
Class Driver Description	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_DESCRIPTION`
Class Driver Prefix	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_PREFIX`
Class Driver Revision	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_REVISION`
Class Driver Vendor	`IVIUPCONVERTER_ATTR_CLASS_DRIVER_VENDOR`
Driver Capabilities	

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Class Group Capabilities	`IVIUPCONVERTER_ATTR_GROUP_CAPABILITIES`
Supported Instrument Models	`IVIUPCONVERTER_ATTR_SUPPORTED_INSTRUMENT_MODELS`
Driver Identification	
Specific Driver Class Spec Major Version	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJOR_VERSION`
Specific Driver Class Spec Minor Version	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINOR_VERSION`
Specific Driver Description	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_DESCRIPTION`
Specific Driver Prefix	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_PREFIX`
Specific Driver Revision	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_REVISION`
Specific Driver Vendor	`IVIUPCONVERTER_ATTR_SPECIFIC_DRIVER_VENDOR`
Instrument Identification	
Instrument Firmware Revision	`IVIUPCONVERTER_ATTR_INSTRUMENT_FIRMWARE_REVISION`
Instrument Manufacturer	`IVIUPCONVERTER_ATTR_INSTRUMENT_MANUFACTURER`
Instrument Model	`IVIUPCONVERTER_ATTR_INSTRUMENT_MODEL`
User Options	
Cache	`IVIUPCONVERTER_ATTR_CACHE`
Interchange Check	`IVIUPCONVERTER_ATTR_INTERCHANGE_CHECK`
Query Instrument Status	`IVIUPCONVERTER_ATTR_QUERY_INSTRUMENT_STATUS`
Range Check	`IVIUPCONVERTER_ATTR_RANGE_CHECK`
Record Value Coercions	`IVIUPCONVERTER_ATTR_RECORD_COERCIONS`
Simulate	`IVIUPCONVERTER_ATTR_SIMULATE`
IQ	
IQ Enabled	`IVIUPCONVERTER_ATTR_IQ_ENABLED`
IQ Impairment	
IQ Impairment Enabled	`IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_ENABLED`

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
IQ Impairment I-Offset	IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_I_OFFSET
IQ Impairment Q-Offset	IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_Q_OFFSET
IQ Impairment Ratio	IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_RATIO
IQ Impairment Skew	IVIUPCONVERTER_ATTR_IQ_IMPAIRMENT_SKEW
IQ Nominal Voltage	IVIUPCONVERTER_ATTR_IQ_NOMINAL_VOLTAGE
IQ Swap Enabled	IVIUPCONVERTER_ATTR_IQ_SWAP_ENABLED
Pulse Modulation	
Pulse Modulation Enabled	IVIUPCONVERTER_ATTR_PULSE_MODULATION_ENABLED
Pulse Modulation External Polarity	IVIUPCONVERTER_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY
Reference Oscillator	
Reference Oscillator External Frequency	IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY
Reference Oscillator Output Enabled	IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_OUTPUT_ENABLED
Reference Oscillator Source	IVIUPCONVERTER_ATTR_REFERENCE_OSCILLATOR_SOURCE
RF Output	
Active RF Output	IVIUPCONVERTER_ATTR_ACTIVE_RF_OUTPUT
Attenuator Hold Enabled	IVIUPCONVERTER_ATTR_ATTENUATOR_HOLD_ENABLED
Is Ready	IVIUPCONVERTER_ATTR_IS_READY
RF Output Bandwidth	IVIUPCONVERTER_ATTR_RF_OUTPUT_BANDWIDTH
RF Output Count	IVIUPCONVERTER_ATTR_RF_OUTPUT_COUNT
RF Output Enabled	IVIUPCONVERTER_ATTR_RF_OUTPUT_ENABLED
RF Output Frequency	IVIUPCONVERTER_ATTR_RF_OUTPUT_FREQUENCY
RF Output Gain	IVIUPCONVERTER_ATTR_RF_OUTPUT_GAIN
RF Output Level	IVIUPCONVERTER_ATTR_RF_OUTPUT_LEVEL
RF Output Ready Trigger	IVIUPCONVERTER_ATTR_RF_OUTPUT_READY_TRIGGER
Sweep	

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Frequency Step	
Frequency Step Dwell	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_DWELL
Frequency Step Scaling	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SCALING
Frequency Step Single Step Enabled	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED
Frequency Step Size	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_SIZE
Frequency Step Start	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_START
Frequency Step Stop	IVIUPCONVERTER_ATTR_FREQUENCY_STEP_STOP
Frequency Sweep	
Frequency Sweep Start	IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_START
Frequency Sweep Stop	IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_STOP
Frequency Sweep Time	IVIUPCONVERTER_ATTR_FREQUENCY_SWEEP_TIME
Is Sweeping	IVIUPCONVERTER_ATTR_IS_SWEEPING
List	
List Dwell	IVIUPCONVERTER_ATTR_LIST_DWELL
List Selected Name	IVIUPCONVERTER_ATTR_LIST_SELECTED_NAME
List Single Step Enabled	IVIUPCONVERTER_ATTR_LIST_SINGLE_STEP_ENABLED
Power Step	
Power Step Dwell	IVIUPCONVERTER_ATTR_POWER_STEP_DWELL
Power Step Single Step Enabled	IVIUPCONVERTER_ATTR_POWER_STEP_SINGLE_STEP_ENABLED
Power Step Size	IVIUPCONVERTER_ATTR_POWER_STEP_SIZE
Power Step Start	IVIUPCONVERTER_ATTR_POWER_STEP_START
Power Step Stop	IVIUPCONVERTER_ATTR_POWER_STEP_STOP
Gain Step	
Gain Step Dwell	IVIUPCONVERTER_ATTR_GAIN_STEP_DWELL
Gain Step Single Step Enabled	IVIUPCONVERTER_ATTR_GAIN_STEP_SINGLE_STEP_ENABLED

Table 33-7 IviUpconverter C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Gain Step Size	IVIUPCONVERTER_ATTR_GAIN_STEP_SIZE
Gain Step Start	IVIUPCONVERTER_ATTR_GAIN_STEP_START
Gain Step Stop	IVIUPCONVERTER_ATTR_GAIN_STEP_STOP
Power Sweep	
Power Sweep Start	IVIUPCONVERTER_ATTR_POWER_SWEEP_START
Power Sweep Stop	IVIUPCONVERTER_ATTR_POWER_SWEEP_STOP
Power Sweep Time	IVIUPCONVERTER_ATTR_POWER_SWEEP_TIME
Gain Sweep	
Gain Sweep Start	IVIUPCONVERTER_ATTR_GAIN_SWEEP_START
Gain Sweep Stop	IVIUPCONVERTER_ATTR_GAIN_SWEEP_STOP
Gain Sweep Time	IVIUPCONVERTER_ATTR_GAIN_SWEEP_TIME
Sweep Mode	IVIUPCONVERTER_ATTR_SWEEP_MODE
Sweep Trigger Source	IVIUPCONVERTER_ATTR_SWEEP_TRIGGER_SOURCE

A. Specific Driver Development Guidelines

A.1. Introduction

This section describes situations driver developers should be aware of when developing a specific instrument driver that complies with the IviUpconverter class.

A.2. Disabling Unused Extension Groups

Specific drivers are required to disable extension capability groups that an application program does not explicitly use. The specific driver can do so by setting the attributes of an extension capability group to the values that this section recommends. A specific driver can set these values for all extension capability groups when the `Prefix_init`, `Prefix_InitWithOptions`, or `Prefix_reset` functions execute. This assumes that the extension capability groups remain disabled until the application program explicitly uses them. For the large majority of instruments, this assumption is true.

Under certain conditions, a specific driver might have to implement a more complex approach. For some instruments, configuring a capability group might affect instrument settings that correspond to an unused extension capability group. If these instrument settings affect the behavior of the instrument, then this might result in an interchangeability problem. If this can occur, the specific driver must take appropriate action so that the instrument settings that correspond to the unused extension capability group do not affect the behavior of the instrument when the application program performs an operation that might be affected by those settings.

The remainder of this section recommends attribute values that effectively disable each extension capability group.

Disabling the IviUpconverterALC Extension Group

Attribute values that effectively disable the IviUpconverterALC extension group are shown in the following table.

Table A-1. Values for Disabling the IviUpconverterALC Group

Attribute	Value
ALC Enabled	False

Disabling the IviUpconverterAnalogModulationSource Extension Group

The IviUpconverterAnalogModulationSource extension group does not affect instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviUpconverterAnalogModulationSource extension group.

Disabling the IviUpconverterCalibration Extension Group

The IviUpconverterCalibration extension group does not affect instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviUpconverterCalibration extension group.

Disabling the IviUpconverterAttenuatorHold Extension Group

Attribute values that effectively disable the IviUpconverterAttenuatorHold extension group are shown in the following table.

Table A-2. Values for Disabling the IviUpconverterAttenuatorHold Group

Attribute	Value
Attenuator Hold Enabled	False

Disabling the IviUpconverterBypass Extension Group

Attribute values that effectively disable the IviUpconverterBypass extension group are shown in the following table.

Table A-3. Values for Disabling the IviUpconverterBypass Group

Attribute	Value
Bypass	False

Disabling the IviUpconverterFrequencyStep Extension Group

Attribute values that effectively disable the IviUpconverterFrequencyStep extension group are shown in the following table.

Table A-4. Values for Disabling the IviUpconverterFrequencyStep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterFrequencySweep Extension Group

Attribute values that effectively disable the IviUpconverterFrequencySweep extension group are shown in the following table.

Table A-5. Values for Disabling the IviUpconverterFrequencySweep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterIQImpairment Extension Group

Attribute values that effectively disable the IviUpconverterIQImpairment extension group are shown in the following table.

Table A-6. Values for Disabling the IviUpconverterIQImpairment Group

Attribute	Value
IQ Impairment Enabled	False

Disabling the IviUpconverterList Extension Group

Attribute values that effectively disable the IviUpconverterList extension group are shown in the following table.

Table A-7. Values for Disabling the IviUpconverterList Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterModulateAM Extension Group

Attribute values that effectively disable the IviUpconverterModulateAM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

Table A-8. Values for Disabling the IviUpconverterModulateAM Extension Group

Attribute	Value
AM Enabled	False

Disabling the IviUpconverterModulateFM Extension Group

Attribute values that effectively disable the IviUpconverterModulateFM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

Table A-9. Values for Disabling the IviUpconverterModulateFM Extension Group

Attribute	Value
FM Enabled	False

Disabling the IviUpconverterModulateIQ Extension Group

Attribute values that effectively disable the IviUpconverterModulateIQ extension group are shown in the following table.

Table A-10. Values for Disabling the IviUpconverterModulateIQ Group

Attribute	Value
IQ Enabled	False

Disabling the IviUpconverterModulatePM Extension Group

Attribute values that effectively disable the IviUpconverterModulatePM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

Table A-11. Values for Disabling the IviUpconverterModulatePM Extension Group

Attribute	Value
PM Enabled	False

Disabling the IviUpconverterModulatePulse Extension Group

Attribute values that effectively disable the IviUpconverterModulatePulse extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

Table A-12. Values for Disabling the IviUpconverterModulatePulseExtension Group

Attribute	Value
Pulse Modulation Enabled	False

Disabling the IviUpconverterOutputGain Extension Group

The IviUpconverterOutputGain extension group does not affect instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviUpconverterOutputGain extension group.

Disabling the IviUpconverterOutputPowerLevel Extension Group

The IviUpconverterOutputPowerLevel extension group does not affect instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviUpconverterOutputPowerLevel extension group.

Disabling the IviUpconverterOutputReadyTrigger Extension Group

Attribute values that effectively disable the IviUpconverterOutputReadyTrigger extension group are shown in the following table.

Table A-13. Values for Disabling the IviUpconverterOutputReadyTrigger Group

Attribute	Value
RF Output Ready Trigger	“None”

Disabling the IviUpconverterPowerStep Extension Group

Attribute values that effectively disable the IviUpconverterPowerStep extension group are shown in the following table.

Table A-14. Values for Disabling the IviUpconverterPowerStep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterGainStep Extension Group

Attribute values that effectively disable the IviUpconverterGainStep extension group are shown in the following table.

Table A-15. Values for Disabling the IviUpconverterGainStep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterPowerSweep Extension Group

Attribute values that effectively disable the IviUpconverterPowerSweep extension group are shown in the following table.

Table A-16. Values for Disabling the IviUpconverterPowerSweep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterGainSweep Extension Group

Attribute values that effectively disable the IviUpconverterGainSweep extension group are shown in the following table.

Table A-17. Values for Disabling the IviUpconverterGainSweep Group

Attribute	Value
Sweep Mode	None

Disabling the IviUpconverterReferenceOscillator Extension Group

Attribute values that effectively disable the IviUpconverterReferenceOscillator extension group are shown in the following table.

Table A18. Values for Disabling the IviUpconverterReferenceOscillator Extension Group

Attribute	Value
Reference Oscillator Source	Internal

Disabling the IviUpconverterSoftwareTrigger Extension Group

The IviUpconverterSoftwareTrigger extension group affects the instrument behavior only when the Sweep Trigger Source attribute is set to Software Trigger. Therefore, this specification does not recommend attribute values that disable the IvUpconverterSoftwareTrigger extension group..

Disabling the IviUpconverterSweep Extension Group

Attribute values that effectively disable the IviUpconverterSweep extension group are shown in the following table.

Table A-19. Values for Disabling the IviUpconverterSweep Group

Attribute	Value
Sweep Mode	None

A.3. Special Consideration for Query Instrument Status

Based on the value of Query Instr Status, the instrument may be queried by the specific driver to determine if it has encountered an error.

B. Interchangeability Checking Rules

B.1. Introduction

IVI drivers have a feature called interchangeability checking. Interchangeability checking returns a warning when it encounters a situation where the application program might not produce the same behavior when the user attempts to use a different instrument.

B.2. When to Perform Interchangeability Checking

Interchangeability checking occurs when all of the following conditions are met:

- The Interchange Check attribute is set to True
- The user calls one of the following functions:
  - Is Ready
  - Wait Until Ready

B.3. Interchangeability Checking Rules

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the base capability group. In addition, interchangeability checking is performed on extension capability groups for which the user has ever set any of the attributes of the group. If the user has never set any attributes of an extension capability group, interchangeability checking is not performed on that group.

In general interchangeability warnings are generated if the following conditions are encountered:

- An attribute that affects the behavior of the instrument is not in a state that the user specifies.
- The user sets a class driver defined attribute to an instrument-specific value.
- The user configures the value of an attribute that the class defines as read-only. In a few cases the class drivers define read-only attributes that specific drivers might implement as read/write.

The remainder of this section defines additional rules and exceptions for each capability group.

IviUpconverterBase Capability Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the External LO Enabled attribute is set to false, then the External LO Frequency attribute does not need to be in a user-specified state.

IviUpconverterOutputGain Capability Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterOutputGain capability group or in any extension group need be in a user-specified state.

IviUpconverterOutputPowerLevel Capability Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterOutputPowerLevel capability group or in any extension group need be in a user-specified state.

IviUpconverterALC Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the ALC Enabled attribute is set to False, attributes in the IviUpconverterALC extension group need not be in a user-specified state.

IviUpconverterAttenuatorHold Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterAttenuatorHold extension group.

IviUpconverterCalibration Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterCalibration extension group.

IviUpconverterAnalogModulationSource Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterAnalogModulationSource extension group.

IviUpconverterBypass Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBypass capability group or in any extension group need be in a user-specified state.
2. If the Bypass attribute is set to True, no attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

IviUpconverterFrequencyStep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterFrequencyStep capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Frequency Step, attributes in the IviUpconverterFrequencyStep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterFrequencyStep extension group need not be in a user-specified state.

IviUpconverterFrequencySweep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterFrequencySweep capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Frequency Sweep, attributes in the IviUpconverterFrequencySweep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterFrequencySweep extension group need not be in a user-specified state.

IviUpconverterIQImpairment Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterIQImpairment capability group or in any extension group need be in a user-specified state.

2. If the IQ Impairment Enabled attribute is set to False, attributes in the IviUpconverterIQImpairment extension group need not be in a user-specified state.

3. If the IviUpconverterModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterIQImpairment extension group need not be in a user-specified state.

IviUpconverterList Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterList capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to List, attributes in the IviUpconverterList extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterList extension group need not be in a user-specified state.

IviUpconverterModulateAM Extension Group

1. If the RF Output Enabled attribute is set to False, attributes in the IviUpconverterModulateAM extension group need not be in a user-specified state.

2. If the AM Enabled attribute is set to False, all other attributes in the IviUpconverterModulateAM extension group need not be in a user-specified state.

IviUpconverterModulateFM Extension Group

1. If the RF Output Enabled attribute is set to False, attributes in the IviUpconverterModulateFM extension group need not be in a user-specified state.

2. If the FM Enabled attribute is set to False, all other attributes in the IviUpconverterModulateFM extension group need not be in a user-specified state.

IviUpconverterModulateIQ Extension Group

1. If the RF Output Enabled attribute is set to False, attributes in the IviUpconverterModulateIQ extension group need not be in a user-specified state.

2. If the IQ Enabled attribute is set to False, attributes in the IviUpconverterModulateIQ extension group need not be in a user-specified state.

IviUpconverterModulatePM Extension Group

1. If the RF Output Enabled attribute is set to False, attributes in the IviUpconverterModulatePM extension group need not be in a user-specified state.

2. If the PM Enabled attribute is set to False, all other attributes in the IviUpconverterModulatePM extension group need not be in a user-specified state.

IviUpconverterModulatePulse Extension Group

1. If the RF Output Enabled attribute is set to False, attributes in the IviUpconverterModulatePulse extension group need not be in a user-specified state.

2. If the Pulse Modulation Enabled attribute is set to False, all other attributes in the IviUpconverterModulatePulse extension group need not be in a user-specified state.

IviUpconverterOutputReadyTrigger Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterOutputReadyTrigger extension group.

IviUpconverterPowerStep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Power Step, attributes in the IviUpconverterPowerStep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterPowerStep extension group need not be in a user-specified state.

IviUpconverterGainStep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Gain Step, attributes in the IviUpconverterGainStep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterGainStep extension group need not be in a user-specified state.

IviUpconverterPowerSweep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Power Sweep, attributes in the IviUpconverterPowerSweep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterPowerSweep extension group need not be in a user-specified state.

IviUpconverterGainSweep Extension Group

1. If the RF Output Enabled attribute is set to False, no other attributes in the IviUpconverterBase capability group or in any extension group need be in a user-specified state.

2. If the Sweep Mode attribute is not set to Gain Sweep, attributes in the IviUpconverterGainSweep extension group need not be in a user-specified state.

3. If the IviUpconverterSweep extension group does not need to be checked for interchangeability, then attributes in the IviUpconverterGainSweep extension group need not be in a user-specified state.

IviUpconverterReferenceOscillator Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterReferenceOscillator extension group.

IviUpconverterSoftwareTrigger Extension Group

No additional interchangeability rules or exceptions are defined for the IviUpconverterSoftwareTrigger extension group.

IviUpconverterSweep Extension Group

1. If the Sweep Mode attribute is set to None, all other attributes in the IviUpconverterSweep extension group need not be in a user-specified state.

2. If the RF Output Enabled attribute is set to False, all the attributes in the IviUpconverterSweep extension group need not be in a user-specified state.