

# IVI-4.8: IviSpecAn Class Specification

April 15, 2011 Edition Revision 2.0

# **Important Information**

The IVI-4.8: IviSpecAn 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.

1	Ove	erview	of the IviSpecAn Specification	9
	1.1	Introdu	action	9
	1.2	IviSpec	cAn Class Overview	9
	1.3		nces	
	1.4	Definit	tions of Terms and Acronyms	10
2	lviQ	enoc Ar	າ Class Capabilities	12
_				
	2.1 2.2		uctioncAn Group Names	
	2.3	_	ted Capability Names	
		2.3.1	Trace	
	2.4		Marker	
	2.4 2.5		an Attribute and Parameter Values	
			Namespace	
	2.6	.NETT	IviSpecAn Session Factory	14
3	Ger	neral R	Requirements	16
•	3.1		um Class Compliance	
	5.1	3 1 1	Disable	
	3.2	0.1.1	ility Group Compliance	
4	IviS		าBase Capability Group	
	4.1	IviSpec	cAnBase Capability Group Overview	17
		4.1.1	Sweep Coupling Overview	
	4.2	IviSpec	cAnBase Attributes	19
		4.2.1	Amplitude Units	20
		4.2.2	Attenuation	22
		4.2.3	Attenuation Auto	
		4.2.4	Detector Type	
		4.2.5	Detector Type Auto	
		4.2.6	Frequency Start	
		4.2.7	Frequency Stop	
		4.2.8	Frequency Offset.	
		4.2.9	Input Impedance	
		4.2.10	Number Of Sweeps	
		4.2.10	Reference Level	
		4.2.11	Reference Level Offset	
		4.2.12	Resolution Bandwidth	
		4.2.14	Resolution Bandwidth Auto	33

		4.2.15		
		4.2.16	Sweep Time	
		4.2.17	Sweep Time Auto	38
		4.2.18	Trace Count	39
		4.2.19	Trace Item (IVI-COM and IVI.NET Only)	40
		4.2.20	Trace Name (IVI-COM and IVI.NET Only)	41
		4.2.21	Trace Size (IVI-C and IVI-COM Only)	42
		4.2.22	Trace Type	43
		4.2.23	Vertical Scale	45
		4.2.24	Video Bandwidth	47
		4.2.25	Video Bandwidth Auto	48
	4.3	IviSpec	cAnBase Functions	49
		4.3.1	Abort	50
		4.3.2	Acquisition Status	51
		4.3.3	Configure Acquisition	53
		4.3.4	Configure Frequency Center Span	55
		4.3.5	Configure Frequency Offset (IVI-C Only)	
		4.3.6	Configure Frequency Start Stop	
		4.3.7	Configure Level	
		4.3.8	Configure Sweep Coupling	
		4.3.9	Configure Trace Type (IVI-C Only)	
		4.3.10	Fetch Y Trace	
		4.3.11	Get Trace Name (IVI-C Only).	
		4.3.12	Initiate	
		4.3.13	Query Trace Size (IVI-C Only)	
		4.3.14	* **	
		4.3.15	Create Spectrum (IVI.NET only)	
	4.4	IviSpec	cAn Behavior Model	
5	5.1 5.2	IviSpec IviSpec 5.2.1 5.2.2	AnMultitrace Extension Group	
		5.2.3	Exchange Traces	
	5.0	5.2.4	Subtract Traces	
	5.3		cAnMultitrace Behavior Model	
	5.4	ivispec	cAnMultitrace Compliance Notes	80
6	IviS	pecAn	nMarker Extension Group	81
	6.1		cAnMarker Extension Group Overview	
	6.2		cAnMarker Attributes	
		6.2.1	Active Marker	82
		6.2.2	Marker Amplitude	83
		6.2.3	Marker Count	84
		6.2.4	Marker Enabled	
		6.2.5	Marker Frequency Counter Enabled	86
		6.2.6	Marker Frequency Counter Resolution	
		6.2.7	Marker Name (IVI-COM Only)	
			÷ •	88
		6.2.7	Marker Name (IVI-COM Only)	

		6.2.11 Peak Excursion	
		6.2.12 Signal Track Enabled	94
	6.3	IviSpecAnMarker Functions	95
		6.3.1 Configure Marker Enabled	96
			97
			98
		6.3.4 Configure Signal Track Enabled (IVI-C C	Only)99
		6.3.5 Disable All Markers	
		6.3.6 Get Marker Name (IVI-C and IVI .NET C	Only)101
		6.3.7 Marker Search	
		6.3.8 Move Marker (IVI-C Only)	
		6.3.9 Query Marker	
		6.3.10 Set Active Marker (IVI-C Only)	
		6.3.11 Set Instrument From Marker	
	6.4	IviSpecAnMarker Behavior Model	
	6.5	IviSpecAnMarker Compliance Notes	111
_	1:0	noo An Trinner Frateries Orons	440
1		pecAnTrigger Extension Group.	
	7.1	IviSpecAnTrigger Extension Group Overview	
	7.2	IviSpecAnTrigger Attributes	
	7.3	IviSpecAnTrigger Functions	
	7.4	IviSpecAnTrigger Behavior Model	
	7.5	IviSpecAnTrigger Compliance Notes	
8	IviS	pecAnExternalTrigger Extension	n Group 118
	8.1	IviSpecAnExternalTrigger Extension Group Over	view 118
	8.2	IviSpecAnExternalTrigger Attributes	
		8.2.1 External Trigger Level	
	8.3	IviSpecAnExternalTrigger Functions	
		8.3.1 Configure External Trigger	
	8.4	IviSpecAnExternalTrigger Behavior Model	
	8.5	IviSpecAnExternalTrigger Compliance Notes	
۵	lvie	noc An Coffwaro Trigger Extensio	n Group 125
9		pecAnSoftwareTrigger Extensio	
	9.1	IviSpecAnSoftwareTrigger Overview	
	9.2	IviSpecAnSoftwareTrigger Functions	
	0.2		
	9.3	IviSpecAnSoftwareTrigger Behavior Model	
	9.4	IviSpecAnSoftwareTrigger Compliance Notes	123
10	IviS	pecAnVideoTrigger Extension G	roup 126
	10.1	IviSpecAnVideoTrigger Extension Group Overvie	·w126
	10.2	IviSpecAnVideoTrigger Attributes	
	10.3	IviSpecAnVideoTrigger Functions	

	10.4	IviSpecAnVideoTrigger Behavior Model	
	10.5	IviSpecAnVideoTrigger Compliance Notes	132
11	lviS	pecAnDisplay Extension Group	133
	11.1	IviSpecAnDisplay Extension Group Overview	
	11.2	IviSpecAnDisplay Attributes	
		11.2.1 Number Of Divisions	134
		11.2.2 Units Per Division	135
	11.3	IviSpecAnDisplay Behavior Model	135
	11.4	IviSpecAnDisplay Compliance Notes	135
12	IviS	pecAnMarkerType Extension Group	136
	12.1	IviSpecAnMarkerType Extension Group Overview	
	12.2	IviSpecAnMarkerType Attributes	
		12.2.1 Marker Type	
	12.3	IviSpecAnMarkerType Functions	
		12.3.1 Query Marker Type (IVI-C Only)	
	12.4	IviSpecAnMarkerType Behavior Model	
	12.5	IviSpecAnMarkerType Compliance Notes	141
13	lviS	pecAnDeltaMarker Extension Group	142
. •	13.1	IviSpecAnDeltaMarker Extension Group Overview	
	13.1	IviSpecAnDeltaMarker Attributes	
	13.2	13.2.1 Reference Marker Amplitude	
		13.2.2 Reference Marker Position	
	13.3	IviSpecAnDeltaMarker Functions	
	13.3	13.3.1 Make Marker Delta	
		13.3.2 Query Reference Marker	
	13.4	IviSpecAnDeltaMarker Behavior Model	
	13.5	IviSpecAnDeltaMarker Compliance Notes	
14	lviQ	pecAnExternalMixer Extension Group	150
17	14.1	IviSpecAnExternalMixer Extension Group Overview	
	14.1	14.1.1 Mixer Configuration	
		14.1.2 Conversion Loss	
	14.2	IviSpecAnExternalMixer Attributes	
	17.2	14.2.1 External Mixer Average Conversion Loss	
		14.2.2 External Mixer Bias	
		14.2.3 External Mixer Bias Enabled.	
		14.2.4 External Mixer Bias Limit	
		14.2.5 External Mixer Conversion Loss Table Enabled	
		14.2.6 External Mixer Enabled	
		14.2.7 External Mixer Harmonic	
		14.2.8 External Mixer Number Of Ports	
	14.3	IviSpecAnExternalMixer Functions	
	17.3	14.3.1 Configure Conversion Loss Table	
		14.3.2 Configure Conversion Loss Table Enabled (IVI-C Only)	
		14.3.3 Configure External Mixer	
		14.3.4 Configure External Mixer Bias	
		14.3.5 Configure External Mixer Bias Enabled (IVI-C Only)	
		17.5.5 Configure External wince Dias Eliabled (1 v 1-C Only)	100

		14.3.6 Configure External Mixer Enabled (IVI-C Only)	
		14.3.7 Configure External Mixer Number Of Ports (IVI-C Only)	168
	14.4	IviSpecAnExternalMixing Behavior Model	
	14.5	IviSpecAnExternalMixing Compliance Notes	169
15	IviS	pecAnPreselector Extension Group	170
	15.1	IviSpecAnPreselector Extension Group Overview	
	15.2	IviSpecAnPreselector Functions	170
		15.2.1 Peak Preselector	
	15.3	IviSpecAnPreselector Behavior Model	
	15.4	IviSpecAnPreselector Compliance Notes	172
16	IviS	pecAn Attribute ID Definitions	173
17	IviS	pecAn Attribute Value Definitions	
	17.1	IviSpecAn Attribute Value Extension Bases	178
18	IviS	pecAn Function Parameter Value Definitions	179
19	IviS	pecAn Error, Completion Code, and Exception Cla	SS
Defi	nition	ns 181	
	19.1	IVI.NET IviSpecAn Exceptions and Warnings	182
		19.1.1 MarkerNotEnabledException	
		19.1.2 NotDeltaMarkerException	184
20	IviS	pecAn Hierarchies	185
_	20.1	IviSpecAn .NET Hierarchy	
		20.1.1 IviSpecAn .NET Interfaces	
		20.1.2 .NET Interface Reference Properties	189
	20.2	IviSpecAn COM Hierarchy	
		20.2.1 IviSpecAn COM Interfaces	
		20.2.2 IviSpecAn COM Interface Reference Properties	
		20.2.3 IviSpecAn COM Category	
	20.3	IviSpecAn C Function Hierarchy	196
Spe	cific [	Oriver Development Guidelines	198
		uction	
		Domain Measurements	
		Level, and Test Speed Optimizations	
		ling Unused Extensions	
	Query	Instrument Status	200
Inte		geability Checking Guidelines	
		uction	
		to Perform Interchangeability Checking	
	Intercl	hangeability Checking Rules	201

# IviSpecAn Class Specification

# **IviSpecAn Revision History**

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

Table 1-1. IviSpecAn Class Specification Revisions

<b>Revision Number</b>	Date of Revision	Revision Notes	
Revision 1.0	April 2002	Specification approved by IVI Foundation	
Revision 1.0	April 2008	Editorial change to update the IVI Foundation contact information in the Important Information section to remove obsolete address information and refer only to IVI Foundation web site.	
Revision 1.0	April 2009	Editorial change to update repeated capabilities section to include both qualified and unqualified repeated capability names.	
Revision 2.0	June 9, 2010	Incorporated IVI .NET	
Revision 2.0	April 15, 2011	Editorial change to add .NET overloads for Fetch Y and Read Y methods and remove the reference to complex spectrums.	

#### **API Versions**

Architecture	Drivers that comply with version 2.0comply with all of the versions below
С	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 IviSpecAn Specification

#### 1.1 Introduction

This specification defines the IVI class for spectrum analyzers. The IviSpecAn class is designed to support the typical spectrum analyzers as well as common extended functionality found in more complex instruments. This section summarizes the *IviSpecAn Specification* itself and contains general information that the reader may need in order to understand, interpret, and implement aspects of this specification. These aspects include the following:

IviSpecAn Class Overview

Spectrum Analyzer Overview

References

Definitions of Terms and Acronyms

## 1.2 IviSpecAn Class Overview

This specification defines the IVI class for spectrum analyzers. The IviSpecAn class is designed to support the typical swept analyzer as well as common extended functionality found in instruments that are more complex. The IviSpecAn class conceptualizes a swept analyzer as an instrument that can measure the amplitude at discrete frequencies while sweeping a receiver through a range of frequencies. Typically, the measured quantity is a voltage or power level.

There is a second type of analyzer that is often referred to as a dynamic signal analyzer, an audio analyzer, or an FFT analyzer. While many of the end results are the same, the process of taking the measurements is quite different. Since they are so different, they are not covered in this specification.

The IviSpecAn class is divided into the base capability group and several extension groups. The base capability group is used to configure an analyzer for a typical sweep, initiate that sweep, and return an array of measured values. The IviSpecAnBase capability group is described in Section 4, *IviSpecAnBase Capability Group*.

In addition to the base capabilities, the IviSpecAn class defines extended capabilities for spectrum analyzers that have:

Simple mathematical operations on multiple traces

Marker and delta-marker functionality

External and Video trigger sources

Basic display function of specifying units per division

External mixer capability to expand the frequency range

Peak preselector to reduce noise and increase dynamic range

The IviSpecAn extended capabilities are arranged into a set of extension capability groups.

## 1.3 References

Several other documents and specifications are related to this specification. These other related documents are the following:

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 an acronyms that are specific to the IviSpecAn class.

Intermediate Frequency (IF) A signal at a known, fixed frequency, dependent on the analyzer,

> obtained by submitting the input signal to a series of frequency conversions in the mixer block. This instrument specific frequency is

chosen so as to enable easier analysis of the input signal.

Local Oscillator (LO) A device used to generate a signal which is beat against the signal of

interest to mix it to a different frequency.

Mixer A component that accepts, as its input, two signals with different

> frequencies and presents at its output a signal equal in frequency to the sum of the frequencies of the input signals, a signal equal in frequency to the difference between the frequencies of the input signals, and, if

they are not filtered out, the original input frequencies.

In a spectrum analyzer, filters are then applied in order to isolate the

desired frequency component in the output signal (IF signal).

Preselectors A network of filters and preamplifiers that are built into one unit for

> reducing noise and increasing dynamic range of a spectrum analyzer. Preselectors are often separate instruments, but they are instruments that only work with spectrum analyzers. Some analyzers have internal

preselectors.

RF Radio Frequency

Signal Tracking There are two prevalent methods of signal tracking in the industry.

> 1. A search for largest signal on screen is made, and the spectrum analyzer center frequency is tuned to this signal.

2. The marker is placed on a signal (anywhere on screen) and when marker track is enabled, the frequency of the peak of the selected signal is used for tuning the spectrum analyzer. This does not need to be the largest signal on screen, and this method tracks the specified signal in the presence of a larger signal.

This specification implements the first method of signal tracking.

# 2 IviSpecAn Class Capabilities

## 2.1 Introduction

The IviSpecAn specification divides generic analyzer 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 for each capability group.

## 2.2 IviSpecAn Group Names

The capability group names for the IviSpecAn 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 Group Capabilities attribute.

Table 2-1. IviSpecAn Group Names

Group Name	Description
IviSpecAnBase	Base: Complies with the IviSpecAn base capabilities.
IviSpecAnMultitrace	Extension: IviSpecAn with the ability to do simple mathematical operations on multiple traces.
IviSpecAnMarker	Extension: IviSpecAn with the ability to utilize internal markers to perform various measurements.
IviSpecAnTrigger	Extension: IviSpecAn with the ability to utilize different trigger sources.
IviSpecAnExternalTrigger	Extension: IviSpecAn with the ability to utilize external trigger level and slope.
IviSpecAnSendSoftwareTrigg er	Extension: IviSpecAn with the ability to receive software triggers.
IviSpecAnVideoTrigger	Extension: IviSpecAn with the ability to utilize video trigger level and slope.
IviSpecAnDisplay	Extension: IviSpecAn with the ability to configure basic display characteristics.
IviSpecAnMarkerType	Extension: IviSpecAn with the ability to specify different marker types.
IviSpecAnDeltaMarker	Extension: IviSpecAn with the ability to utilize delta markers to perform various relative measurements.
IviSpecAnExternalMixer	Extension: IviSpecAn with the ability to expand the frequency range.
IviSpecAnPreselector	Extension: IviSpecAn with the ability to utilize peak preselector function to reduce noise and increase dynamic range.

## 2.3 Repeated Capability Names

The IviSpecAn 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*.

Trace

Marker

#### 2.3.1 Trace

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

When using the IviSpecAnMarker extension group, the value for the Marker Trace attribute corresponds to the Trace capability.

## 2.3.2 Marker

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

The marker capability is used by the IviSpecAnMarker, IviSpecAnMarkerType, and IviSpecAnDeltaMarker extension groups and will be available only if those groups are implemented.

When using the Marker capability, it is necessary to set the Active Marker attribute before performing marker operations. All subsequent marker related attributes and functions operate exclusively on the active marker.

#### 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 IviSpecAn class is Ivi.SpecAn.

## 2.6 .NET IviSpecAn Session Factory

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

#### .NET Method Prototype

#### **Parameters**

Inputs	Inputs Description	
name	A session name or a logical name that points to a session that uses an IVI.NET IviSpecAn 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.	
options	A string that allows the user to specify the initial values of certain inherent attributes. The default is an empty string.	

Description	Base Type
interface pointer to the fivispeer in interface of the driver	IIviSpecAn
nt	1

### .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 IviSpecAn instrument class API from the logical name "My LogicalName", use the following:

```
IIviSpecAn specAn = IviSpecAn.Create("MyLogicalName");
```

In this case, the II be supplied from	D of the instrument will n the configuration store ar	not be verified, the insend/or driver defaults.	strument will not be r	eset, and options will

# 3 General Requirements

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

## 3.1 Minimum Class Compliance

To be compliant with the IviSpecAn Class Specification, an IVI specific driver shall conform to the requirements for an IVI class-compliant specific driver as specified in IVI-3.1: *Driver Architecture Specification*, implement the inherent capabilities defined by IVI-3.2: *Inherent Capabilities Specification*, and implement the IviSpecAn capability group.

#### 3.1.1 Disable

Refer to IVI-3.2: *Inherent Capabilities Specification* for the prototype of this function. The IviSpecAn 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 IviSpecAnBase Capability Group

## 4.1 IviSpecAnBase Capability Group Overview

The IviSpecAnBase capability group supports spectrum analyzers that configure and take a frequency sweep. A frequency sweep is thought of as adjusting the frequency of a tuner from the start frequency to the stop frequency in a defined amount of time (For time domain measurements see Section 0, *Time Domain Measurements*). While the tuner is being adjusted, power levels (or voltage levels) are being measured for the frequencies. The result is an array of amplitude versus frequency data. In addition to configuring the start and stop frequency, a user can also set a frequency offset. This affects the setting of the spectrum analyzer's absolute frequencies, such as start, stop, center, and marker. It does not affect values, which are the difference of frequencies, such as span and delta marker.

The IviSpecAnBase capability group also includes functions for configuring the analyzer as well as initiating and retrieving measurements. This includes configuring the sweep, range, and resolution. A typical spectrum analyzer does not have several input channels. It can, however, have several traces. Multiple traces are covered in the IviSpecAnMultitrace extension group. The trigger functions are covered in Trigger, External Trigger, and Video Trigger extension groups.

In addition, most spectrum analyzers have a coupled function mode, which links the resolution bandwidth (RBW), video bandwidth (VBW), and sweep time together. This behavior is described in Section 4.1.1, *Sweep Coupling Overview*.

The diagram shown in Figure 4-1 details the tuning, filtering, and detection process in a typical spectrum analyzer.

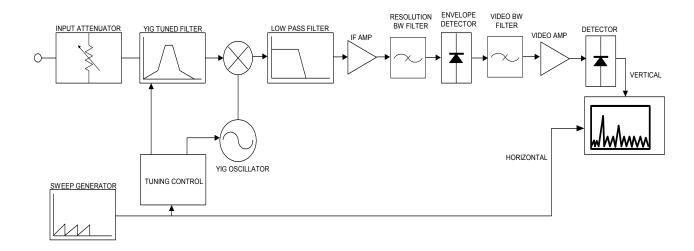


Figure 4-1. Typical Spectrum Analyzer Diagram

The diagram shown in Figure 4-2 details how the detector detects the signal and outputs the signal on multiple traces.

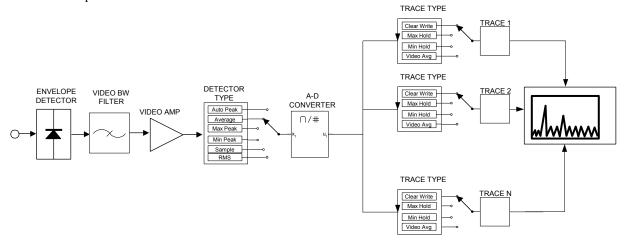


Figure 4-2. Detector Type and Trace Type Attributes

## 4.1.1 Sweep Coupling Overview

Many spectrum analyzers are capable of coupling the resolution bandwidth, video bandwidth, and sweep time together. The instrument makes decisions based on the various settings to ensure the accuracy of the measurement. If the user elects to choose the settings, it is possible to place constraints on the system that make accurate measurements impossible. Most analyzers respond to this condition with a message indicating that the measurement is uncalibrated.

When all the settings are coupled, the instrument makes decisions along the following hierarchy. The RBW and VBW are typically locked together and set for 1% of the span. For example, if the span were set for 100 MHz, the instrument chooses 1 MHz for the RBW and VBW. To measure very low signal levels, the RBW may need to be more narrowed, which slows the sweep speed. Lastly, the sweep time is dependent on the RBW and the VBW. As a rule of thumb:

As can be seen from the above equation, the RBW has a drastic effect on sweep speed. These are issues to keep in mind when developing automated tests.

When considering interchangeability and measurement accuracy, it is important to consider the way different instruments couple settings together. The valid combination of RBW, VBW, and sweep time on one spectrum analyzer may not be a valid combination on another spectrum analyzer. Awareness of these differences ensures that the results obtained in one system correlate with the results from another system.

## 4.2 IviSpecAnBase Attributes

The IviSpecAnBase capability group defines the following attributes:

Amplitude Units

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.

Attenuation Auto

.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.

Detector Type

Detector Type Auto

.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.

Frequency Start

.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.

Frequency Stop

.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.

Frequency Offset

.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.

Input Impedance

.NET Exceptions

Number Of Sweeps

.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.

Reference Level

.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.

Reference Level Offset

.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.

Resolution Bandwidth

.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.

Resolution Bandwidth Auto

.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.

Sweep Mode Continuous

.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.

Sweep Time

.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.

Sweep Time Auto

.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.

Trace Count

Trace Item (IVI-COM and IVI.NET Only)

Trace Name (IVI-COM and IVI.NET Only)

Trace Size (IVI-C and IVI-COM Only)

Trace Type

Vertical Scale

Video Bandwidth

.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.

Video Bandwidth Auto

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

## 4.2.1 Amplitude Units

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

#### .NET Property Name

Level.AmplitudeUnits

#### .NET Enumeration Name

AmplitudeUnits

## **COM Property Name**

Level.AmplitudeUnits

#### **COM Enumeration Name**

 ${\tt IviSpecAnAmplitudeUnitsEnum}$ 

#### **C** Constant Name

IVISPECAN\_ATTR\_AMPLITUDE\_UNITS

#### Description

Specifies the amplitude units for input, output and display amplitude.

#### **Defined Values**

Name	Description
	*

		Language	Identifier
dBm	S	ets the spectrum a	analyzer to measure in decibels relative to 1 milliwatt.
		С	IVISPECAN_VAL_AMPLITUDE_UNITS_DBM
		.NET	AmplitudeUnits.dBm
		COM	IviSpecAnAmplitudeUnitsDBM
dBmV	S	ets the spectrum a	analyzer to measure in decibels relative to 1 millivolt.
		С	IVISPECAN_VAL_AMPLITUDE_UNITS_DBMV
		.NET	AmplitudeUnits.dBmV
		COM	IviSpecAnAmplitudeUnitsDBMV
dBuV	Sets the spectrum		analyzer to measure in decibels relative to 1 microvolt.
		C	IVISPECAN_VAL_AMPLITUDE_UNITS_DBUV
		.NET	AmplitudeUnits.dBuV
		COM	IviSpecAnAmplitudeUnitsDBUV
Volt	S	ets the spectrum a	analyzer to measure in volts.
		С	IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT
		.NET	AmplitudeUnits.Volt
		COM	IviSpecAnAmplitudeUnitsVolt
Watt	Sets the spectrum a		analyzer to measure in watts.
		С	IVISPECAN_VAL_AMPLITUDE_UNITS_WATT
		.NET	AmplitudeUnits.Watt
		COM	IviSpecAnAmplitudeUnitsWatt

#### .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

## 4.2.2 Attenuation

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

## .NET Property Name

Level.Attenuation

## **COM Property Name**

Level.Attenuation

#### **C Constant Name**

IVISPECAN\_ATTR\_ATTENUATION

## Description

Specifies the input attenuation (in positive dB).

## .NET Exceptions

## 4.2.3 Attenuation Auto

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

## .NET Property Name

Level.AttenuationAuto

## **COM Property Name**

Level.AttenuationAuto

#### **C Constant Name**

IVISPECAN\_ATTR\_ATTENUATION\_AUTO

## Description

If set to True, attenuation is automatically selected. If set to False, attenuation is manually selected.

## .NET Exceptions

# 4.2.4 Detector Type

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

## .NET Property Name

Acquisition.DetectorType

#### .NET Enumeration Name

DetectorType

## **COM Property Name**

Acquisition.DetectorType

#### **COM Enumeration Name**

 ${\tt IviSpecAnDetectorTypeEnum}$ 

#### **C** Constant Name

IVISPECAN ATTR DETECTOR TYPE

## Description

Specifies the detection method used to capture and process the signal. This governs the data acquisition for a particular sweep, but does not have any control over how multiple sweeps are processed.

#### **Defined Values**

Name	Descrip	otion		
		Language	Identifier	
Auto Peak		Allows the detector to capture better readings by using both positive and negative peak values when noise is present.		
		С	IVISPECAN_VAL_DETECTOR_TYPE_AUTO_PEAK	
		.NET	DetectorType.AutoPeak	
		COM	IviSpecAnDetectorTypeAutoPeak	
Average	Averag display	verage value of samples taken within the bin for a dedicated point on the isplay.		
		C IVISPECAN_VAL_DETECTOR_TYPE_AVERA		
		.NET	DetectorType.Average	
		COM IviSpecAnDetectorTypeAverage		
Maximum Peak		nins the maximum video signal between the last display point and the ent display point.		
		C	IVISPECAN_VAL_DETECTOR_TYPE_MAX_PEAK	

		.NET	DetectorType.MaxPeak
		COM	IviSpecAnDetectorTypeMaxPeak
Minimum Peak		s the minimum v display point.	video signal between the last display point and the
		C	IVISPECAN_VAL_DETECTOR_TYPE_MIN_PEAK
		.NET	DetectorType.MinPeak
		COM	IviSpecAnDetectorTypeMinPeak
Sample	Pick on	ne point within a	bin.
		С	IVISPECAN_VAL_DETECTOR_TYPE_SAMPLE
		.NET	DetectorType.Sample
		COM	IviSpecAnDetectorTypeSample
RMS	RMS va	S value of samples taken within the bin for a dedicated point on the blay.	
		С	IVISPECAN_VAL_DETECTOR_TYPE_RMS
		.NET	DetectorType.Rms
		COM	IviSpecAnDetectorTypeRMS

#### .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 4.2.5 Detector Type Auto

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Acquisition

## .NET Property Name

Acquisition.DetectorTypeAuto

## **COM Property Name**

Acquisition.DetectorTypeAuto

#### **C** Constant Name

IVISPECAN\_ATTR\_DETECTOR\_TYPE\_AUTO

## **Description**

If set to True, the detector type is automatically selected. The relationship between Trace Type and Detector Type is not defined by the specification when the Detector Type Auto is set to True. If set to False, the detector type is manually selected.

## .NET Exceptions

# 4.2.6 Frequency Start

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

#### .NET Property Name

Frequency.Start

## **COM Property Name**

Frequency.Start

## **C** Constant Name

IVISPECAN ATTR FREQUENCY START

## Description

Specifies the left edge of the frequency domain in Hertz. This is used in conjunction with the Frequency Stop attribute to define the frequency domain. If the Frequency Start attribute value is equal to the Frequency Stop attribute value then the spectrum analyzer's horizontal attributes are in time-domain.

## .NET Exceptions

# 4.2.7 Frequency Stop

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

#### .NET Property Name

Frequency.Stop

#### **COM Property Name**

Frequency.Stop

## **C** Constant Name

IVISPECAN ATTR FREQUENCY STOP

## **Description**

Specifies the right edge of the frequency domain in Hertz. This is used in conjunction with the Frequency Start attribute to define the frequency domain. If the Frequency Start attribute value is equal to the Frequency Stop attribute value then the spectrum analyzer's horizontal attributes are in time-domain.

## .NET Exceptions

## 4.2.8 Frequency Offset

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Offset

#### .NET Property Name

Frequency.Offset

#### **COM Property Name**

Frequency.Offset

#### **C Constant Name**

IVISPECAN ATTR FREQUENCY OFFSET

## **Description**

Specifies an offset value, in Hertz, that is added to the frequency readout. The offset is used to compensate for external frequency conversion. This changes the driver's Frequency Start and Frequency Stop attributes. The equations relating the affected values are:

Frequency Start = Actual Start Frequency + Frequency Offset

Frequency Stop = Actual Stop Frequency + Frequency Offset

Marker Position = Actual Marker Frequency + Frequency Offset

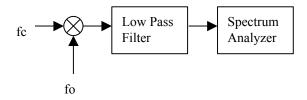


Figure 4-3. Simplified Down Converter

Figure 4-3 shows a very simplified block diagram of a down converter. Typically, a user will program the spectrum analyzer in terms of the original signal,  $f_c$ , not the input signal of the spectrum analyzer. Setting the Frequency Offset attribute to  $f_0$  allows for the resulting data to be relative to the original signal.

#### .NET Exceptions

# 4.2.9 Input Impedance

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

## .NET Property Name

Level.InputImpedance

## **COM Property Name**

Level.InputImpedance

#### **C Constant Name**

IVISPECAN\_ATTR\_INPUT\_IMPEDANCE

## **Description**

Specifies the value of input impedance, in ohms, expected at the active input port. This is typically 50 ohms or 75 ohms.

## .NET Exceptions

# 4.2.10 Number Of Sweeps

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViInt32	R/W	N/A	None	Configure Acquisition

## .NET Property Name

Acquisition.NumberOfSweeps

## **COM Property Name**

Acquisition.NumberOfSweeps

#### **C** Constant Name

IVISPECAN ATTR NUMBER OF SWEEPS

## Description

This attribute defines the number of sweeps. This attribute value has no effect if the Trace Type attribute is set to the value Clear Write.

## .NET Exceptions

## 4.2.11 Reference Level

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

## .NET Property Name

Level.Reference

## **COM Property Name**

Level.Reference

#### **C Constant Name**

IVISPECAN ATTR REFERENCE LEVEL

## Description

The calibrated vertical position of the captured data used as a reference for amplitude measurements. This is typically set to a value slightly higher than the highest expected signal level. The units are determined by the Amplitude Units attribute.

## .NET Exceptions

## 4.2.12 Reference Level Offset

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

## .NET Property Name

Level.ReferenceOffset

## **COM Property Name**

Level.ReferenceOffset

#### **C Constant Name**

IVISPECAN\_ATTR\_REFERENCE\_LEVEL\_OFFSET

## Description

Specifies an offset for the Reference Level attribute. This value is used to adjust the reference level for external signal gain or loss. A positive value corresponds to a gain while a negative number corresponds to a loss. The value is in dB.

## .NET Exceptions

## 4.2.13 Resolution Bandwidth

Data Type	Access	Applies to	Coercion	High Level Functions	
ViReal64	R/W	N/A	Up	Configure Sweep Coupling	

## .NET Property Name

SweepCoupling.ResolutionBandwidth

## **COM Property Name**

SweepCoupling.ResolutionBandwidth

#### **C** Constant Name

IVISPECAN ATTR RESOLUTION BANDWIDTH

## Description

Specifies the width of the IF filter in Hertz. For more information see Section 4.1.1, Sweep Coupling Overview.

## .NET Exceptions

## 4.2.14 Resolution Bandwidth Auto

Data Type	Acces	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Sweep Coupling

#### .NET Property Name

SweepCoupling.ResolutionBandwidthAuto

## **COM Property Name**

SweepCoupling.ResolutionBandwidthAuto

## **C** Constant Name

IVISPECAN ATTR RESOLUTION BANDWIDTH AUTO

## **Description**

If set to True, the resolution bandwidth is automatically selected. If set to False, the resolution bandwidth is manually selected.

## .NET Exceptions

# 4.2.15 Sweep Mode Continuous

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Acquisition

## .NET Property Name

Acquisition.SweepModeContinuous

## **COM Property Name**

Acquisition.SweepModeContinuous

## **C** Constant Name

IVISPECAN\_ATTR\_SWEEP\_MODE\_CONTINUOUS

# Description

If set to True, the sweep mode is continuous If set to False, the sweep mode is not continuous..

## .NET Exceptions

# 4.2.16 Sweep Time

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64(C/COM)	R/W	N/A	Up	Configure Sweep Coupling
<pre>Ivi.Driver.PrecisionTime Span (.NET)</pre>				

### .NET Property Name

SweepCoupling.SweepTime

## **COM Property Name**

SweepCoupling.SweepTime

## **C** Constant Name

IVISPECAN ATTR SWEEP TIME

## **Description**

Specifies the length of time to sweep from the left edge to the right edge of the current domain. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan class.

## .NET Exceptions

# 4.2.17 Sweep Time Auto

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Sweep Coupling

## .NET Property Name

SweepCoupling.SweepTimeAuto

## **COM Property Name**

SweepCoupling.SweepTimeAuto

## **C** Constant Name

IVISPECAN ATTR SWEEP TIME AUTO

# Description

If set to True, the sweep time is automatically selected If set to False, the sweep time is manually selected..

## .NET Exceptions

## 4.2.18 Trace Count

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

## .NET Property Name

Traces.Count

This property is inherited from IIviRepeatedCapabilityCollection.

## **COM Property Name**

Traces.Count

### **C** Constant Name

IVISPECAN\_ATTR\_TRACE\_COUNT

## **Description**

Specifies the number of Traces available.

## .NET Exceptions

## 4.2.19 Trace Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviSpecAn Trace	RO	Traces	None	N/A

### .NET Property Name

```
IIviSpecAnTrace Traces[String name];
```

This indexer is inherited from IIviRepeatedCapabilityCollection. The name parameter uniquely identifies a particular trace in the traces collection.

## **COM Property Name**

#### **C** Constant Name

N/A

## Description

Trace Item uniquely identifies a trace in the traces collection. It returns an interface pointer which can be used to control the attributes and other functionality of that trace.

The Item property takes a trace name. If the user passes an invalid value for the name parameter, the property returns an error.

Valid names include physical repeated capability identifiers and virtual repeated capability identifiers.

## .NET Exceptions

## 4.2.20 Trace Name (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	Traces	None	None

### .NET Property Name

Traces[].Name

This property is inherited from IIviRepeatedCapabilityIdentification.

#### **COM Property Name**

#### **C Constant Name**

N/A. (Use the Get Trace Name function.)

## **Description**

Returns the physical repeated capability identifier defined by the specific driver for the trace that corresponds to the index that the user specifies. If the driver defines a qualified trace name, this property returns the qualified name.

For C and COM, valid values for the Index parameter are between one and the value of the Trace Count attribute, inclusive. If the user passes an invalid value for the Index parameter, the value of this attribute is an empty string.

### .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**

For an instrument with only one Trace, i.e. the Trace Count attribute is one, the driver may return an empty string.

# 4.2.21 Trace Size (IVI-C and IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions	
ViInt32	RO	Trace	None	Query Trace Size	

## .NET Property Name

N/A

Size should be determined from the ISpectrum interface of the spectrum returned by the driver.

## **COM Property Name**

Traces.Item().Size

## **C** Constant Name

IVISPECAN\_ATTR\_TRACE\_SIZE

# Description

Returns the number of points in the trace array.

# 4.2.22 Trace Type

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Trace	None	Configure Trace Type

## .NET Property Name

Traces[].Type

## .NET Enumeration Name

TraceType

## **COM Property Name**

Traces.Item().Type

### **COM Enumeration Name**

IviSpecAnTraceTypeEnum

## **C** Constant Name

IVISPECAN\_ATTR\_TRACE\_TYPE

# Description

Specifies the representation of the acquired data.

### **Defined Values**

Name	Description				
		Language	Identifier		
Clear Write	Sets the spectrum analyzer to clear previous sweep data off the display before performing a sweep. Subsequent sweeps may or may not clear the display first, but the data array at the end of the sweep is entirely new.				
			IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE		
			TraceType.ClearWrite		
		COM	IviSpecAnTraceTypeClearWrite		
Maximum Hold	Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, which ever is higher.				
		С	IVISPECAN_VAL_TRACE_TYPE_MAX_HOLD		
		.NET	TraceType.MaxHold		
COM IviSpecAnTraceTy		IviSpecAnTraceTypeMaxHold			
Minimum Hold	Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, which ever is lower.				
		С	IVISPECAN_VAL_TRACE_TYPE_MIN_HOLD		

Name	De	scription					
		Language	Identifier				
		.NET	TraceType.MinHold				
		COM	IviSpecAnTraceTypeMinHold				
Video Average	Sets the spectrum analyzer to maintain a running average of the swept data.						
		С	IVISPECAN_VAL_TRACE_TYPE_VIDEO_AVERAGE				
	.NET		TraceType.VideoAverage				
	COM		IviSpecAnTraceTypeVideoAverage				
View	Disables acquisition into this trace but displays the existing trace data.						
	C IVISPECAN_VAL_TRACE_TYPE_VIEW						
	.NET TraceType.View		TraceType.View				
		COM	IviSpecAnTraceTypeView				
Store	Disables acquisition and disables the display of the existing trace data.						
	C		IVISPECAN_VAL_TRACE_TYPE_STORE				
		.NET	TraceType.Store				
		COM	IviSpecAnTraceTypeStore				

## .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.

## 4.2.23 Vertical Scale

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

## .NET Property Name

Acquisition.VerticalScale

#### .NET Enumeration Name

VerticalScale

## **COM Property Name**

Acquisition.VerticalScale

#### **COM Enumeration Name**

IviSpecAnVerticalScaleEnum

### **C** Constant Name

IVISPECAN\_ATTR\_VERTICAL\_SCALE

# Description

Specifies the vertical scale of the measurement hardware (use of log amplifiers versus linear amplifiers).

### **Defined Values**

Name	De	escription		
		Language		Identifier
Linear	Specifies the ver		rtic	al scale in linear units.
		С	Ι	VISPECAN_VAL_VERTICAL_SCALE_LINEAR
		.NET	V	erticalScale.Linear
		COM	Ι	viSpecAnVerticalScaleLinear
Logarithmic	Specifies the ver		rtic	al scale in logarithmic units.
		С		VISPECAN_VAL_VERTICAL_SCALE_LOGARI
		.NET	V	erticalScale.Logarithmic
		COM	I	viSpecAnVerticalScaleLogarithmic

## .NET Exceptions

## **Compliance Notes**

- 1. If an IVI class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

## 4.2.24 Video Bandwidth

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	Up	Configure Sweep Coupling

## .NET Property Name

SweepCoupling.VideoBandwidth

## **COM Property Name**

SweepCoupling.VideoBandwidth

### **C** Constant Name

IVISPECAN\_ATTR\_VIDEO\_BANDWIDTH

# Description

Specifies the video bandwidth of the post-detection filter in Hertz.

## .NET Exceptions

## 4.2.25 Video Bandwidth Auto

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Sweep Coupling

## .NET Property Name

SweepCoupling.VideoBandwidthAuto

## **COM Property Name**

SweepCoupling.VideoBandwidthAuto

### **C** Constant Name

IVISPECAN\_ATTR\_VIDEO\_BANDWIDTH\_AUTO

## Description

If set to True, the video bandwidth is automatically selected. If set to False, the video bandwidth is manually selected.

## .NET Exceptions

# 4.3 IviSpecAnBase Functions

The IviSpecAnBase capability group defines the following functions:

Abort

Acquisition Status

Configure Acquisition

Configure Frequency Center Span

Configure Frequency Offset (IVI-C Only)

Configure Frequency Start Stop

Configure Level

Configure Sweep Coupling

Configure Trace Type (IVI-C Only)

Fetch Y Trace

Get Trace Name (IVI-C Only)

Initiate

Query Trace Size (IVI-C Only)

Read Y Trace

Create Spectrum (IVI.NET Only)

This section describes the behavior and requirements of each function.

## 4.3.1 Abort

## Description

This function aborts a previously initiated measurement and returns the spectrum analyzer to the idle state.

This function does not check instrument status.

### .NET Method Prototype

```
void Traces.Abort();
```

## **COM Method Prototype**

```
HRESULT Traces. Abort ();
```

## **C Function Prototype**

```
ViStatus IviSpecAn Abort (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

# 4.3.2 Acquisition Status

## Description

This function determines and returns the status of an acquisition.

## .NET Method Prototype

AcquisitionStatus Traces.AcquisitionStatus ();

## **COM Method Prototype**

## **C Function Prototype**

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Outputs	Description	Base Type
Status (C/COM)	Returns the acquisition status.	ViInt32
Return value (.NET)	Returns the acquisition status.	ViInt32

## **Defined Values for Status Parameter**

Name	Description		
		Language	Identifier
Acquisition Complete	The spectrum analyzer has completed the acquisition.		
		С	IVISPECAN_VAL_ACQUISITION_STATUS_COMPLETE
		.NET	AcquisitionStatus.Complete
		COM	IviSpecAnAcquisitionStatusComplete
Acquisition In Progress	The spectrum analyzer is still acquiring data.		
		С	IVISPECAN_VAL_ACQUISITION_STATUS _IN_PROGRESS
	.NET		AcquisitionStatus.InProgress
		COM	IviSpecAnAcquisitionStatusInProgress
Acquisition Status Unknown	The spectrum analyzer cannot determine the status of the acquisition.		
		С	IVISPECAN_VAL_ACQUISITION_STATUS_UNKNOWN

.NET	AcquisitionStatus.Unknown
COM	IviSpecAnAcquisitionStatusUnknown

## 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.

### **Compliance Notes**

- 1. If an IVI class driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

## 4.3.3 Configure Acquisition

## Description

This function configures the acquisition attributes of the spectrum analyzer.

## .NET Method Prototype

#### **COM Method Prototype**

## **C** Function Prototype

```
ViStatus IviSpecAn_ConfigureAcquisition (ViSession Vi,
ViBoolean SweepModeContinuous,
ViInt32 NumberOfSweeps,
ViBoolean DetectorTypeAuto,
ViInt32 DetectorType,
ViInt32 VerticalScale);
```

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
SweepModeContinuous	Enables or disables continuous sweeping. The driver uses this value to set the Sweep Mode Continuous attribute. See the attribute description for more details.	ViBoolean
NumberOfSweeps	Specifies the number of sweeps to take. The driver uses this value to set the Number Of Sweeps attribute. See the attribute description for more details.	ViInt32
DetectorTypeAuto	Enables or Disables the auto detector. The driver uses this value to set the Detector Type Auto attribute. See the attribute description for more details.	ViBoolean

DetectorType	Specifies the method of capturing and processing signal data. The driver uses this value to set the Detector Type attribute. See the attribute description for more details.	ViInt32
VerticalScale	Specifies the vertical scale. The driver uses this value to set the Vertical Scale attribute. See the attribute description for more details.	ViInt32

# Return Values (C/COM)

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

# .NET Exceptions

## 4.3.4 Configure Frequency Center Span

### Description

This function configures the frequency range defining the center frequency and the frequency span. If the span corresponds to zero Hertz, then the spectrum analyzer operates in time-domain mode. Otherwise, the spectrum analyzer operates in frequency-domain mode,

This function modifies the Frequency Start and Frequency Stop attributes as follows:

```
Frequency Start = CenterFrequency - Span / 2
Frequency Stop = CenterFrequency + Span / 2
```

### .NET Method Prototype

## **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
CenterFrequency	Specifies the center frequency of the frequency sweep. The units are Hertz.	ViReal64
Span	Specifies the frequency span of the frequency sweep. 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

# 4.3.5 Configure Frequency Offset (IVI-C Only)

## Description

This function configures the Frequency Offset attribute of the spectrum analyzer. This function affects the setting of the spectrum analyzer's absolute frequencies, such as start, stop, center, and marker. It does not affect values such as span and delta marker, which are the difference of frequencies.

## .NET Method Prototype

N/A
(Use the Frequency Offset property.)

## **COM Method Prototype**

N/A
(Use the Frequency Offset property.)

## **C Function Prototype**

#### **Parameters**

Inputs	Inputs Description	
Vi	Instrument handle.	ViSession
FrequencyOffset	Specifies the frequency offset. The driver uses this value to set the Frequency Offset attribute. See the attribute description for more details. The units are Hertz.	ViReal64

## Return Values (C)

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

## 4.3.6 Configure Frequency Start Stop

## Description

This function configures the frequency range defining its start frequency and its stop frequency. If the start frequency is equal to the stop frequency, then the spectrum analyzer operates in time-domain mode. Otherwise, the spectrum analyzer operates in frequency-domain mode.

## .NET Method Prototype

### **COM Method Prototype**

```
HRESULT Frequency.ConfigureStartStop ([in] DOUBLE StartFrequency, [in] DOUBLE StopFrequency);
```

### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
StartFrequency	Specifies the start frequency of the frequency sweep (in Hertz). The driver uses this value to set the Frequency Start attribute. See the attribute description for more details.	ViReal64
StopFrequency	Specifies the stop frequency of the frequency sweep (in Hertz). The driver uses this value to set the Frequency 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

## 4.3.7 Configure Level

#### Description

This function configures the vertical attributes of the spectrum analyzer. This corresponds to the Amplitude Units, Input Attenuation, Input Impedance, Reference Level, and Reference Level Offset attributes.

## .NET Method Prototype

#### **COM Method Prototype**

#### **C** Function Prototype

```
ViStatus IviSpecAn_ConfigureLevel (ViSession Vi,
ViInt32 AmplitudeUnits,
ViReal64 InputImpedance,
ViReal64 ReferenceLevel,
ViReal64 ReferenceLevelOffset,
ViBoolean AttenuationAuto,
ViReal64 Attenuation);
```

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
AmplitudeUnits	Specifies the amplitude units for input, output and display. The driver uses this value to set the Amplitude Units attribute. See the attribute description for more details.	ViInt32
InputImpedance	Specifies the input impedance. The driver uses this value to set the Input Impedance attribute. See the attribute description for more details.	ViReal64

ReferenceLevel	Specifies the amplitude value of the reference level. The driver uses this value to set the Reference Level attribute. See the attribute description for more details.	ViReal64
ReferenceLevelOffset	Specifies the offset value to the reference level. The driver uses this value to set the Reference Level Offset attribute. See the attribute description for more details.	ViReal64
AttenuationAuto	Enables or disables auto attenuation. The driver uses this value to set the Attenuation Auto attribute. See the attribute description for more details.	ViBoolean
Attenuation	Specifies the attenuation level. If AttenuationAuto is True then this parameter is ignored. The driver uses this value to set the Attenuation 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

## 4.3.8 Configure Sweep Coupling

## **Description**

This function configures the coupling and sweeping attributes. For additional sweep coupling information refer to Section 4.1.1, *Sweep Coupling Overview*.

### .NET Method Prototype

```
void SweepCoupling.Configure (Boolean resolutionBandwidthAuto,
      Boolean videoBandwidthAuto,
      Boolean sweepTimeAuto,
void SweepCoupling.Configure (Boolean resolutionBandwidthAuto,
      Boolean videoBandwidthAuto,
      Ivi.Driver.PrecisionTimeSpan sweepTime);
                                            // sets SweepTimeAuto
                                             // to Auto.Off
void SweepCoupling.Configure (Boolean resolutionBandwidthAuto,
      Double videoBandwidth,
                          // sets VideoBandwidthAuto to Auto.Off
      Boolean sweepTimeAuto,
void SweepCoupling.Configure (Boolean resolutionBandwidthAuto,
      Double videoBandwidth, // sets VideoBandwidthAuto to Auto.Off
      // to Auto.Off
void SweepCoupling.Configure (
      Double resolutionBandwidth, // sets ResolutionBandwidthAuto to Auto.Off
      Boolean videoBandwidthAuto,
      Boolean sweepTimeAuto,
void SweepCoupling.Configure (
      Double resolutionBandwidth, // sets ResolutionBandwidthAuto to Auto.Off
      Boolean videoBandwidthAuto,
      // to Auto.Off
void SweepCoupling.Configure (
      Double resolutionBandwidth, // sets ResolutionBandwidthAuto to Auto.Off
      Double videoBandwidth, // sets VideoBandwidthAuto to Auto.Off
      Boolean sweepTimeAuto,
void SweepCoupling.Configure (
      Double resolutionBandwidth, // sets ResolutionBandwidthAuto to Auto.Off
      Double videoBandwidth, // sets VideoBandwidthAuto to Auto.Off
      // to Auto.Off
```

### **COM Method Prototype**

HRESULT SweepCoupling.Configure ([in] VARIANT BOOL ResolutionBandwidthAuto,

[in] DOUBLE ResolutionBandwidth,

[in] VARIANT BOOL VideoBandwidthAuto,

[in] DOUBLE VideoBandwidth,

[in] VARIANT BOOL SweepTimeAuto,

[in] DOUBLE SweepTime);

## **C Function Prototype**

ViStatus IviSpecAn ConfigureSweepCoupling (ViSession Vi,

ViBoolean ResolutionBandwidthAuto, ViReal64 ResolutionBandwidth, ViBoolean VideoBandwidthAuto, ViReal64 VideoBandwidth, ViBoolean SweepTimeAuto, ViReal64 SweepTime);

#### **Parameters**

Inputs	Description	Base Type	
Vi	Instrument handle.	ViSession	
ResolutionBandwidth Auto	Enables or disables resolution bandwidth auto coupling. The driver uses this value to set the Resolution Bandwidth Auto attribute. See the attribute description for more details.	ViBoolean	
ResolutionBandwidth	Specifies the measurement resolution bandwidth in Hertz. This value is ignored when ResolutionBandwidthAuto is True. The driver uses this value to set the Resolution Bandwidth attribute. See the attribute description for more details.	ViReal64	
VideoBandwidthAuto	Enables or disables video bandwidth auto coupling. The driver uses this value to set the Video Bandwidth Auto attribute. See the attribute description for more details.	ViBoolean	
VideoBandwidth	Specifies the video bandwidth of the post-detection filter in Hertz. This value is ignored when VideoBandwidthAuto is True. The driver uses this value to set the Video Bandwidth attribute. See the attribute description for more details.	ViReal64	
SweepTimeAuto	Enables or disables sweep time auto coupling. The driver uses this value to set the Sweep Time Auto attribute. See the attribute description for more details.	ViBoolean	

SweepTime	Specifies the length of time to sweep from the left edge to the right edge of the current	ViReal64 (C/COM) PrecisionTimeSpan
	domain. For C and COM, the units are seconds. For .NET, the units are implied by	(.NET)
	PrecisionTimeSpan.This value is ignored	
	when SweepTimeAuto is True. The driver uses this value to set the Sweep Time	
	attribute. See the attribute description for more details.	

## Return Values (C/COM)

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

## .NET Exceptions

# 4.3.9 Configure Trace Type (IVI-C Only)

# Description

This function configures the Trace Type attribute.

## .NET Method Prototype

N/A

Use the Trace Type property.

## **COM Method Prototype**

N/A

Use the Trace Type property.

## **C Function Prototype**

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
TraceName	Specifies the trace name.	ViConstString
TraceType	Specifies the type of trace. The driver uses this value to set the Trace Type attribute. See the attribute description for more details.	ViInt32

## Return Values (C)

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

#### 4.3.10 Fetch Y Trace

#### Description

This function returns the trace the spectrum analyzer acquires. The trace is from a previously initiated acquisition. The user calls the Initiate function to start an acquisition. The user calls the Acquisition Status function to determine when the acquisition is complete.

The user may call the Read Y Trace function instead of the Initiate function. This function starts an acquisition, waits for the acquisition to complete, and returns the trace in one function call.

The Amplitude array returns data that represents the amplitude of the signals obtained by sweeping from the start frequency to the stop frequency (in frequency domain, in time domain the amplitude array is ordered from beginning of sweep to end). The Amplitude Units attribute determines the units of the points in the Amplitude array.

This function does not check the instrument status. The user calls the Error Query function at the conclusion of the sequence to check the instrument status.

The behavior is different for IVI-C/IVI-COM and IVI.NET as follows:

IVI-C/IVI-COM: After this function executes, each element in the amplitude array parameter is either a amplitude or a value indicating that the spectrum analyzer could not sample an amplitude.

IVI.NET: For .NET the return value of ISpectrum<T> is a spectrum object. Refer to Section 4, Common Properties and Methods of Waveform and Spectrum Interfaces, and Section 7, ISpectrum<T> Interface, of IVI-3.18: IVI.NET Utility Classes and Interfaces Specification, for the definition of the ISpectrum object and information regarding its use. In particular, refer to Section 4.2, How to use Waveform and Spectrum Types, in IVI-3.18: IVI.NET Utility Classes and Interfaces Specification, for more information about how to implement these methods.

If the frequency span corresponds to zero Hertz, the spectrum analyzer operates in time domain mode. Therefore the return value of FetchY is a waveform object..

## .NET Method Prototype

```
ISpectrum<Double> Traces[].FetchY (ISpectrum<Double> spectrum);
IWaveform<Double> Traces[].FetchY (IWaveform<Double> waveform);
```

### **COM Method Prototype**

```
HRESULT Traces.Item().FetchY ([in,out] SAFEARRAY(DOUBLE) *Amplitude);
```

#### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
TraceName	Specifies the trace to return.	ViConstString
ArrayLength	Specifies the number of array points requested.	ViInt32
spectrum	This input parameter can be used to provide a	Ivi.Driver.
(.NET)	previously allocated spectrum.	ISpectrum <double></double>
waveform	This input parameter can be used to provide a	Ivi.Driver.
(.NET)	previously allocated waveform.	IWaveform <double></double>

Outputs	Description	Base Type
ActualPoints	Specified the number of points actually returned in the array.	ViInt32
Amplitude[] (C/COM)	Specifies a user allocated (IVI-C) or driver- allocated (IVI-COM) buffer into which the trace amplitudes are stored.	ViReal64
Return value (.NET)	Spectrum containing the measurement	<pre>Ivi.Driver.ISpectrum<dou ble=""></dou></pre>

## Return Values (C/COM)

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

<b>Completion Codes</b>	Description
Over Range	This warning indicates that the signal exceeded the input range.

### .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. The *IVI-3-18: IVI.NET Utility Classes and Interfaces Specification* defines additional spectrum-related exceptions that may be thrown by this method.

Note that .NET does not return an Over Range error. In IVI.NET, the ISpectrum interface includes a property that indicates whether the spectrum contains an out of range value.

# 4.3.11 Get Trace Name (IVI-C Only)

## Description

This function returns the specific driver defined trace name that corresponds to the one-based index that the user specifies. If the driver defines a qualified trace name, this property 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 Trace Count attribute, the function returns an empty string in the Name parameter and returns the Invalid Value error.

### .NET Method Prototype

N/A

(Use Trace Name property.)

## **COM Method Prototype**

N/A

(Use Trace Name property.)

## **C Function Prototype**

## **Parameters**

Inputs	Description	Base Type
Vi	Unique identifier for an IVI session	ViSession
Index	A one-based index that defines which name to return.	ViInt32
NameBufferSize	Specifies the number of bytes in the ViChar array referenced by the Name parameter.	ViInt32

Outputs	Description	Base Type
Name	Specifies the buffer into which the function returns the name that corresponds to the index the user specifies.	ViChar[]
	The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	

# Return Values (C)

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

## **Compliance Notes**

For an instrument with only one Trace, i.e. the Trace Count attribute is one, the driver may return an empty string.

For additional rules for C functions with ViChar Array Output Parameters, see Section 3.1.2.1, *Additional Compliance Rules for C Functions with ViChar Array Output Parameters* of IVI-3.2: *Inherent Capabilities Specification*.

### 4.3.12 Initiate

## Description

This function initiates an acquisition. After calling this function, the spectrum analyzer leaves the idle state.

This function does not check the instrument status. The user calls the Acquisition Status function to determine when the acquisition is complete.

## .NET Method Prototype

```
void Traces.Initiate ();
```

#### **COM Method Prototype**

```
HRESULT Traces. Initiate ();
```

## **C Function Prototype**

ViStatus IviSpecAn Initiate (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. The table below specifies additional class-defined status codes for this function.

<b>Completion Codes</b>	Description
Measure Uncalibrated	This warning indicates that the spectrum analyzer is in an uncalibrated state.

## .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.

The table below specifies additional class-defined warning events for this method.

Warning	Description
Measure Uncalibrated	This warning indicates that the spectrum analyzer is in an uncalibrated state.

# 4.3.13 Query Trace Size (IVI-C Only)

## Description

This function queries the read-only Trace Size attribute.

## .NET Method Prototype

N/A

Size should be determined from the ISpectrum interface of the spectrum returned by the driver.

## **COM Method Prototype**

N/A

(Use the Trace Size property.)

## **C Function Prototype**

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
TraceName	Specifies the Trace name.	ViConstString

Outputs	Description	Base Type
TraceSize	Returns the size of the Trace.	ViInt32

## Return Values (C)

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

#### 4.3.14 Read Y Trace

## Description

This function initiates a signal acquisition based on the present instrument configuration. It then waits for the acquisition to complete, and returns the trace as an array of amplitude values. The amplitude array returns data that represent the amplitude of the signals obtained by sweeping from the start frequency to the stop frequency (in frequency domain, in time domain the amplitude array is ordered from beginning of sweep to end). The Amplitude Units attribute determines the units of the points in the amplitude array. This function resets the sweep count.

If the spectrum analyzer did not complete the acquisition within the time period the user specified with the MaxTime parameter, the function returns the Max Time Exceeded error.

The behavior is different for IVI-C/IVI-COM and IVI.NET as follows:

IVI-C/IVI-COM: After this function executes, each element in the amplitude array parameter is either a amplitude or a value indicating that the spectrum analyzer could not sample an amplitude.

IVI.NET: For .NET the return value of ISpectrum<T> is a spectrum object. Refer to Section 4, Common Properties and Methods of Waveform and Spectrum Interfaces, and Section 7, ISpectrum<T> Interface, of IVI-3.18: IVI.NET Utility Classes and Interfaces Specification, for the definition of the ISpectrum object and information regarding its use. In particular, refer to Section 4.2, How to use Waveform and Spectrum Types, in IVI-3.18: IVI.NET Utility Classes and Interfaces Specification, for more information about how to implement these methods.

If the frequency span corresponds to zero Hertz, the spectrum analyzer operates in time domain mode. Therefore the return value of ReadY is a waveform object..

### .NET Method Prototype

### **COM Method Prototype**

#### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

TraceName	Specifies the trace to return.	ViConstString
MaxTimeMilliseconds (C/COM)	Specifies the maximum length of time allowed for the function to complete in milliseconds.	ViInt32
maximumTime (.NET)	Specifies the maximum length of time allowed	Ivi.Driver.
	for the function to complete.	PrecisionTimeSpan
ArrayLength	Specifies the number of points in the Amplitude	ViInt32
	array .	
spectrum (.NET)	This input parameter can be used to provide a	Ivi.Driver.
	previously allocated spectrum.	ISpectrum <double></double>
waveform (.NET)	This input parameter can be used to provide a	Ivi.Driver.
	previously allocated waveform.	IWaveform <double></double>

Outputs	Description	Base Type
ActualPoints	Specifies the number of points actually returned in the Amplitude array.	ViInt32
Amplitude[] (C/COM)	Specifies a user allocated (IVI-C) or driverallocated (IVI-COM) buffer into which the trace amplitudes are stored.	ViReal64
Return value (.NET)	The measured spectrum	<pre>Ivi.Driver. ISpectrum<double></double></pre>

## **Defined Values for MaxTime Parameter**

Name	Description		
		Language	Identifier
Max Time Immediate	Sets timeout to immediate. The function returns immediately. If no valid measurement value exists, the function returns the Max Time Exceeded error.		
		С	IVISPECAN_VAL_MAX_TIME_IMMEDIATE
		COM	IviSpecAnMaxTimeImmediate
Max Time Infinite	Sets timeout to infinite. The function waits indefinitely for the measurement to complete.		
		С	IVISPECAN_VAL_MAX_TIME_INFINITE
		COM	IviSpecAnMaxTimeInfinite

# **Defined Values for the MaximumTime Parameter (.NET)**

Name	Description			
		Language	Identifier	
Zero	The function returns immediately. If no valid measurement value exists, the function returns an error.			
		.NET		PrecisionTimeSpan.Zero
MaxValue	The function waits indefinitely for the measurement to complete.			
		.NET		PrecisionTimeSpan.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
Over Range Warning	This warning indicates that the signal exceeded the input range.
Measure Uncalibrated	This warning indicates that the data was captured while the spectrum analyzer was in an uncalibrated state.
Max Time Exceeded	This error indicates that the maximum time was 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. The *IVI-3-18: IVI.NET Utility Classes and Interfaces Specification* defines additional spectrum-related exceptions that may be thrown by this method.

Note that .NET does not return an Over Range error. In IVI.NET, the ISpectrum interface includes a property that indicates whether the spectrum contains an out of range value.

Note that the .NET MaxTimeExceededException is defined in IVI-3.2: Inherent Capabilities Specification.

The table below specifies additional class-defined warning events for this method.

Warning	Description
Measure Uncalibrated	This warning indicates that the spectrum analyzer is in an uncalibrated state.

# 4.3.15 Create Spectrum (IVI.NET only)

# Description

This function creates a spectrum object that can be passed to Read and Fetch routines. This is used for performance optimization to control when memory is allocated.

If size is zero, the driver shall allocate the spectrum memory with a size based on the current driver configuration.

# .NET Method Prototype

ISpectrum<Double> Traces.CreateSpectrum (Int32 size);

# **COM Method Prototype**

N/A

# **C Function Prototype**

N/A

#### **Parameters**

Inputs	Description	Base Type
Size	The number of points of type Double in the spectrum	Int32

Outputs	Description	Base Type
Return value (.NET)	The newly allocated Spectrum	Ivi.Driver.
		ISpectrum <double></double>

# .NET Exceptions

# 4.4 IviSpecAn Behavior Model

The following state diagram shows relationships between IviSpecAn Fundamental Capabilities and analyzer behavior.

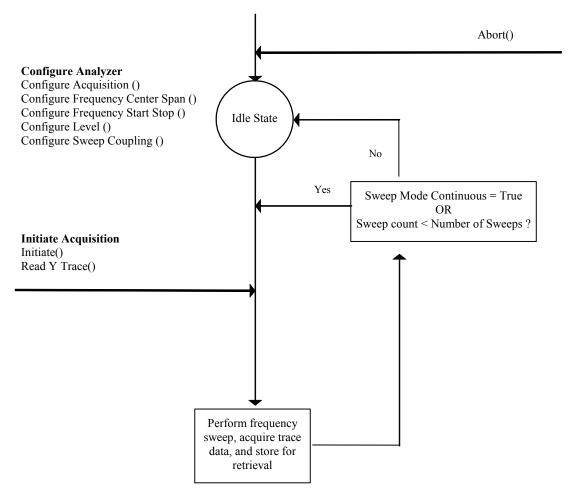


Figure 4-4. IviSpecAn Behavior Model

The main state in the IviSpecAn Class is the Idle state. The analyzer enters the Idle state as the result of being "powered-on", successfully completing a measurement, or by being aborted from a previous measurement by the user with the Abort function. Typically, the user configures the analyzer while it is in the Idle state. IviSpecAn attributes can be configured individually with the Set Attribute function (IVI-C) or using one of the higher-level functions.

The Read Y Trace and Initiate functions cause the analyzer to leave the Idle state. The Read Y Trace function does not return until the measurement process is complete and the analyzer has returned to the Idle state. The Initiate function returns as soon as the analyzer leaves the Idle state.

After the sweep is taken, the analyzer returns to the Idle state. The user can use the Acquisition Status function to determine if the acquisition is complete or is still in progress.

The Fetch Y Trace function is used to return a waveform from a previously initiated measurement.

# 5 IviSpecAnMultitrace Extension Group

# 5.1 IviSpecAnMultitrace Extension Group Overview

The IviSpecAnMultitrace extension group defines extensions for analyzers capable of performing simple mathematical functions on one or more traces.

# 5.2 IviSpecAnMultitrace Functions

The IviSpecAnMultitrace capability group defines the following extended functions:

Add Traces

Copy Trace

**Exchange Traces** 

**Subtract Traces** 

This section describes the behavior and requirements of each function.

# 5.2.1 Add Traces

#### Description

This function modifies a trace to be the point by point sum of two other traces. Any data in the destination trace is deleted.

DestinationTrace = Trace1 + Trace2

# .NET Method Prototype

### **COM Method Prototype**

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
DestinationTrace	Specifies the name of the result.	ViConstString
Tracel	Specifies the name of first trace operand.	ViConstString
Trace2	Specifies the name of the second trace operand.	ViConstString

# Return Values (C/COM)

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

# .NET Exceptions

# 5.2.2 Copy Trace

### Description

This function copies the data array from one trace into another trace. Any data in the Destination Trace is deleted.

#### .NET Method Prototype

# **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
DestinationTrace	Specifies the name of the trace into which the array is copied.	ViConstString
SourceTrace	Specifies the name of the trace to be copied.	ViConstString

#### Return Values (C/COM)

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

# .NET Exceptions

# 5.2.3 Exchange Traces

### Description

This function exchanges the data arrays of two traces.

## .NET Method Prototype

# **COM Method Prototype**

# **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Trace1	Specifies the name of the first trace to be exchanged.	ViConstString
Trace2	Specifies the name of the second trace to be exchanged.	ViConstString

# Return Values (C/COM)

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

# .NET Exceptions

#### 5.2.4 Subtract Traces

### Description

This function modifies a trace to be the point by point difference between two traces. Any data in the destination trace is deleted.

DestinationTrace = Trace1 - Trace2

# .NET Method Prototype

### **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
DestinationTrace	Specifies the name of the result trace.	ViConstString
Trace1	Specifies the name of the first trace operand.	ViConstString
Trace2	Specified the name of the second trace operand.	ViConstString

#### Return Values (C/COM)

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

### .NET Exceptions

# 5.3 IviSpecAnMultitrace Behavior Model

The IviSpecAnMultitrace extension group follows the same behavior model as the IviSpecAnBase capability group described in Section 0, *IviSpecAnBase Behavior Model*.

# 5.4 IviSpecAnMultitrace Compliance Notes

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

# 6 IviSpecAnMarker Extension Group

# 6.1 IviSpecAnMarker Extension Group Overview

The IviSpecAnMarker extension group supports spectrum analyzers that have markers. Markers are applied to traces and used for a wide range of operations. Some operations are simple, such as reading an amplitude value at an X-axis position, while others operations are complex, such as signal tracking.

# 6.2 IviSpecAnMarker Attributes

The IviSpecAnMarker capability group defines the following attributes:

Active Marker

Marker Amplitude

Marker Count

Marker Enabled

Marker Frequency Counter Enabled

Marker Frequency Counter Resolution

Marker Name (IVI-COM Only)

Marker Position

Marker Threshold

Marker Trace

Peak Excursion

Signal Track Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 6.2.1 Active Marker

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select Marker

# .NET Property Name

Marker.ActiveMarker

# **COM Property Name**

Marker.ActiveMarker

#### **C Constant Name**

IVISPECAN\_ATTR\_ACTIVE\_MARKER

# Description

Specifies the marker which is currently active. The values for this attribute correspond to the Marker repeated capability. If the driver defines a qualified Marker name, this attribute returns the qualified name.

# .NET Exceptions

# 6.2.2 Marker Amplitude

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViReal64	RO	Marker	None	Query Marker

# .NET Property Name

Marker.Amplitude

#### **COM Property Name**

Marker.Amplitude

#### **C Constant Name**

IVISPECAN\_ATTR\_MARKER\_AMPLITUDE

# Description

Returns the amplitude of the active marker. The units are specified by the Amplitude Units attribute, except when the Marker Type attribute is set to Delta. Then the units are dB. If the Marker Enabled attribute is set to False, any attempt to read this attribute returns the Marker Not Enabled error.

# .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.

The table below specifies additional class-defined exceptions for this method.

<b>Exception Class</b>	Description
MarkerNotEnabledException	Marker not enabled.

# 6.2.3 Marker Count

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

# .NET Property Name

Marker.Count

# **COM Property Name**

Marker.Count

#### **C** Constant Name

IVISPECAN\_ATTR\_MARKER\_COUNT

# Description

Specifies the number of markers available for a particular instrument.

# .NET Exceptions

# 6.2.4 Marker Enabled

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViBoolean	R/W	Marker	None	Configure Marker Enabled

# .NET Property Name

Marker.Enabled

# **COM Property Name**

Marker.Enabled

#### **C** Constant Name

IVISPECAN\_ATTR\_MARKER\_ENABLED

# Description

If set to True, the active marker is enabled. When False, the active marker is disabled.

# .NET Exceptions

# 6.2.5 Marker Frequency Counter Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Marker	None	Configure Marker Frequency Counter

# .NET Property Name

Marker.FrequencyCounter.Enabled

#### **COM Property Name**

Marker.FrequencyCounter.Enabled

#### **C Constant Name**

IVISPECAN\_ATTR\_MARKER\_FREQUENCY\_COUNTER\_ENABLED

# Description

Enables/disables the marker frequency counter for greater marker measurement accuracy. If set to True, the marker frequency counter is enabled. If set to False, the marker frequency counter is disabled. This attribute returns the Marker Not Enabled error if the Marker Enabled attribute is set to False.

# .NET Exceptions

# 6.2.6 Marker Frequency Counter Resolution

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Marker	Down	Configure Marker Frequency Counter

# .NET Property Name

Marker.FrequencyCounter.Resolution

# **COM Property Name**

Marker.FrequencyCounter.Resolution

#### **C Constant Name**

IVISPECAN\_ATTR\_MARKER\_FREQUENCY\_COUNTER\_RESOLUTION

# Description

Specifies the resolution of the frequency counter in Hertz. The measurement gate time is the reciprocal of the specified resolution.

# .NET Exceptions

# 6.2.7 Marker Name (IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	Markers	None	None

# .NET Property Name

N/A

(Use the Get Marker Name method.)

### **COM Property Name**

#### **C Constant Name**

N/A
(Use the Get Marker Name function.)

# Description

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

#### **Parameters**

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the trace name.	ViChar[]

# 6.2.8 Marker Position

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Marker	None	Move Marker Query Marker

### .NET Property Name

Marker.Position

# **COM Property Name**

Marker.Position

#### **C Constant Name**

IVISPECAN\_ATTR\_MARKER\_POSITION

# Description

Specifies the frequency in Hertz or time position in seconds of the active marker (depending on the mode in which the analyzer is operating, frequency or time-domain). This attribute returns the Marker Not Enabled error if the active marker is not enabled.

# .NET Exceptions

# 6.2.9 Marker Threshold

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViReal64	R/W	Marker	None	Configure Marker Search

# .NET Property Name

Marker.Threshold

# **COM Property Name**

Marker. Threshold

#### **C** Constant Name

IVISPECAN\_ATTR\_MARKER\_THRESHOLD

# Description

Specifies the lower limit of the search domain vertical range for the Marker Search function.

# .NET Exceptions

# 6.2.10 Marker Trace

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Marker	None	Configure Marker Enabled

# .NET Property Name

Marker.Trace

# **COM Property Name**

Marker.Trace

#### **C** Constant Name

IVISPECAN\_ATTR\_MARKER\_TRACE

# Description

Specifies the Trace for the active marker.

# .NET Exceptions

# 6.2.11 Peak Excursion

Dat	a Type	Access	Applies to	Coercion	High Level Functions
ViRe	eal64	R/W	Marker	None	Configure Marker Search

# .NET Property Name

Marker.PeakExcursion

# **COM Property Name**

Marker.PeakExcursion

#### **COM Enumeration Name**

N/A

#### **C Constant Name**

IVISPECAN\_ATTR\_PEAK\_EXCURSION

# Description

Specifies the minimum amplitude variation of the signal in dB that the Marker Search function can identify as a peak.

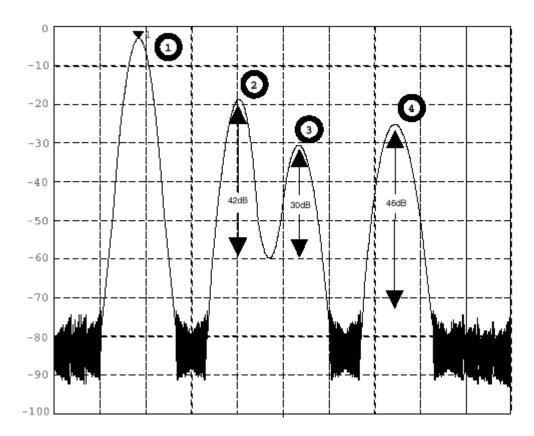


Figure 6-1. Peak Measurement Example

Signal 1 constitutes the absolute peak of the signal (the Marker Search function when the SearchType parameter is set to Highest, positions the active marker to the signal 1 position). Signal 2, 3 and 4 show a maximum relative level change of respectively, 42 dB, 30 dB, and 46 dB.

If the Peak Excursion attribute is set to 40 dB, the Marker Search function will successively position the active marker on signal 2 and signal 4 positions, when the SearchType parameter is set to Next Peak or Next Peak Right. Signal 3 will be ignored because it corresponds to a signal amplitude variation of only 30 dB, which is less than the value set for the Peak Excursion attribute.

# .NET Exceptions

# 6.2.12 Signal Track Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Marker	None	Configure Signal Track Enabled

#### .NET Property Name

Marker.SignalTrackEnabled

# **COM Property Name**

Marker.SignalTrackEnabled

#### **C Constant Name**

IVISPECAN\_ATTR\_SIGNAL\_TRACK\_ENABLED

# Description

If set to True, the spectrum analyzer centers the signal after each sweep. This process invalidates the Frequency Start and Frequency Stop attributes. If set to False, the spectrum analyzer does not center the signal after each sweep.

Operations on this attribute return the Marker Not Enabled error if the active marker is not enabled.

Note: Signal tracking can only be enabled on one marker at any given time. The driver is responsible for enforcing this policy.

# .NET Exceptions

# 6.3 IviSpecAnMarker Functions

The IviSpecAnMarker extension group defines the following functions:

Configure Marker Enabled

Configure Marker Frequency Counter

Configure Marker Search

Configure Signal Track Enabled (IVI-C Only)

Disable All Markers

Get Marker Name (IVI-C and IVI.NET Only)

Marker Search

Move Marker (IVI-C Only)

Query Marker

Set Active Marker (IVI-C Only)

Set Instrument From Marker

This section describes the behavior and requirements of each function.

# 6.3.1 Configure Marker Enabled

# Description

This function enables the active marker on the specified Trace.

## .NET Method Prototype

# **COM Method Prototype**

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
MarkerEnabled	Enables or disables the active marker. The driver uses this value to set the Marker Enabled attribute. See the attribute description for more details.	ViBoolean
MarkerTraceName	Specifies the trace name. The driver uses this value to set the Marker Trace attribute. See the attribute description for more details.	ViConstString

# Return Values (C/COM)

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

#### .NET Exceptions

# 6.3.2 Configure Marker Frequency Counter

### Description

This function sets the marker frequency counter resolution and enables or disables the marker frequency counter.

#### .NET Method Prototype

# **COM Method Prototype**

```
HRESULT Marker.FrequencyCounter.Configure ([in] VARIANT_BOOL Enabled, [in] DOUBLE Resolution);
```

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Enabled	Enables or disables the marker frequency counter. The driver uses this value to set the Marker Frequency Counter Enabled attribute. See the attribute description for more details.	ViBoolean
Resolution	Specifies the frequency counter resolution in Hertz. This value is ignored when Enabled is False. The driver uses this value to set the Marker Frequency Counter Resolution 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

# 6.3.3 Configure Marker Search

# Description

This function configures the Peak Excursion and Marker Threshold attribute values.

## .NET Method Prototype

### **COM Method Prototype**

```
HRESULT Marker.ConfigureSearch ([in] DOUBLE PeakExcursion, [in] DOUBLE MarkerThreshold);
```

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
PeakExcursion	Minimum amplitude variation of the signal that the marker can recognize as a peak in dB. The driver uses this value to set the Peak Excursion attribute. See the attribute description for more details.	ViReal64
MarkerThreshold	Minimum amplitude below which a peak will not be detected. The driver uses this value to set the Marker Threshold 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

# 6.3.4 Configure Signal Track Enabled (IVI-C Only)

# Description

If set to True, the active marker is enabled. When False, the active marker is disabled. For additional information about signal-tracking, see Section 1.4, *Definition of Terms and Acronyms*, and Section 6.2.12, *Signal Track Enabled*.

#### .NET Method Prototype

N/A

(Use the Signal Track Enabled property.)

### **COM Method Prototype**

N/A

(Use the Signal Track Enabled property.)

# **C Function Prototype**

ViStatus IviSpecAn\_ConfigureSignalTrackEnabled (ViSession Vi, ViBoolean SignalTrackEnabled);

# **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
SignalTrackEnabled	If set to True, the active marker is enabled. When False, the active marker is disabled. The driver uses this value to set the Signal Track 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.

# 6.3.5 Disable All Markers

# Description

This function disables all markers.

# .NET Method Prototype

```
void Marker.DisableAll ();
```

# **COM Method Prototype**

```
HRESULT Marker.DisableAll ();
```

# **C** Function Prototype

ViStatus IviSpecAn DisableAllMarkers (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

# 6.3.6 Get Marker Name (IVI-C and IVI .NET Only)

### Description

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

#### .NET Method Prototype

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

### **COM Method Prototype**

N/A

(Use the Marker Name property)

# **C Function ProtoType**

#### **Parameters**

Inputs	Description	Base Type
Vi	Unique identifier for an IVI session	ViSession
Index	A one-based index that defines which name to return.	ViInt32
NameBufferSize	Specifies the number of bytes in the ViChar array referenced by the Name parameter.	ViInt32

Outputs	Description	Base Type
Name (C/COM)	Specifies the buffer into which the function returns the name that corresponds to the index the user specifies.  The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	ViChar[]
Return value (.NET)	The name that corresponds to the index the user specifies.	String

#### Return Values (C)

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

# .NET Exceptions

# **Compliance Notes**

For an instrument with only one Marker, i.e. the Marker Count attribute is one, the driver may return an empty string.

For additional rules for C functions with ViChar Array Output Parameters, see Section 3.1.2.1, *Additional Compliance Rules for C Functions with ViChar Array Output Parameters* of IVI-3.2: *Inherent Capabilities Specification*.

# 6.3.7 Marker Search

# Description

This function specifies the type of marker search and performs the search. This function returns the Marker Not Enabled error if the Marker Enabled attribute is set to False.

# .NET Method Prototype

void Marker.Search (MarkerSearch searchType);

# **COM Method Prototype**

HRESULT Marker.Search ([in] IviSpecAnMarkerSearchEnum SearchType);

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
SearchType	Specifies the type of marker search.	ViInt32

# **Defined Values for SearchType Parameter**

Name	De	Description		
		Language	Identifier	
Highest	Sets marker search for the highest amplitude.			
		C	IVISPECAN_VAL_MARKER_SEARCH_HIGHEST	
		.NET	MarkerSearch.Highest	
		COM	IviSpecAnMarkerSearchHighest	
Minimum	Sets marker search for the minimum amplitude.			
		С	IVISPECAN_VAL_MARKER_SEARCH_MINIMUM	
	.NET MarkerSearch.Minimum		MarkerSearch.Minimum	
		COM	IviSpecAnMarkerSearchMinimum	
Next Peak	Set	ts marker searc	ch for the next highest peak.	
	C IVISPECAN_VAL_MARKER_SEARCH_NEXT_F		IVISPECAN_VAL_MARKER_SEARCH_NEXT_PE AK	
	.NET MarkerSearch.NextPeak		MarkerSearch.NextPeak	
		COM	IviSpecAnMarkerSearchNextPeak	
Next Peak Left	Sets marker search for the next peak left of the marker position.			

•			<del>-</del>
		С	IVISPECAN_VAL_MARKER_SEARCH_NEXT_PE AK_LEFT
		.NET	MarkerSearch.NextPeakLeft
		COM	IviSpecAnMarkerSearchNextPeakLeft
Next Peak Right	Sets marker search for the next peak right of the marker position.		
	C IVISPECAN_VAL_MARKER_SEARCH_NEX AK_RIGHT		IVISPECAN_VAL_MARKER_SEARCH_NEXT_PE AK_RIGHT
	.NET MarkerSearch.NextPeakRight		MarkerSearch.NextPeakRight
		COM	IviSpecAnMarkerSearchNextPeakRight

### 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.

### **Compliance Notes**

- If an IVI class driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 6.3.8 Move Marker (IVI-C Only)

# Description

This function specifies the frequency in Hertz or time position in seconds of the specified horizontal position.

# .NET Method Prototype

N/A

(Use the Marker Position property.)

# **COM Method Prototype**

N/A

(Use the Marker Position property.)

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
MarkerPosition	Horizontal position (Hertz or seconds). The driver uses this value to set the Marker Position 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.

# 6.3.9 Query Marker

# Description

This function returns the horizontal position and the amplitude level of the active marker.

## .NET Method Prototype

```
public struct MarkerInfo {
    public Double Position {};
    public Double Amplitude {};
}
MarkerInfo Marker.Query ();
```

# **COM Method Prototype**

# **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Outputs	Description	Base Type
MarkerPosition (C/COM)	The frequency in Hertz or time position in seconds of the active marker (depending on the mode in which the analyzer is operating, frequency or time-domain). See the Marker Position attribute description for more details.	ViReal64
MarkerAmplitude (C/COM)	The amplitude of the active marker. The units are specified by the Amplitude Units attribute, except when the Marker Type attribute is set to Delta. Then the units are dB. See the Marker Amplitude attribute description for more details.	ViReal64
Return value (.NET)	Struct that contains the marker position and amplitude	MarkerInfo

# Return Values (C/COM)

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

# .NET Exceptions

# 6.3.10 Set Active Marker (IVI-C Only)

### Description

This function selects one of the available markers, and makes it the active marker.

### .NET Method Prototype

```
N/A (Use the Active Marker property.)
```

## **COM Method Prototype**

```
_{\mathrm{N/A}} (Use the Active Marker property.)
```

## **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ActiveMarker	Marker to be selected. The driver uses this value to set the Active Marker 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.

#### 6.3.11 Set Instrument From Marker

#### Description

This function uses the Marker Position or Marker Amplitude attributes to configure the spectrum analyzer setting specified by the InstrumentSetting parameter.

This function may set the Frequency Start, Frequency Stop, or Reference Level attributes.

If the Marker Enabled attribute is set to False, this function returns the Marker Not Enabled error. If the Marker Type attribute is not Delta and the InstrumentSetting parameter is Frequency Span, the function returns the Delta Marker Not Enabled error.

#### .NET Method Prototype

```
void Marker.SetInstrumentFromMarker (InstrumentSetting instrumentSetting);
```

#### **COM Method Prototype**

#### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
InstrumentSetting	Specifies the instrument setting to be set from the marker position.	ViInt32

### 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.

#### **Defined Values for InstrumentSetting Parameter**

Name	Description				
	Language Identifier				
Frequency Center	Sets the center frequency with the Marker Position attribute.				

Name	Description			
		Language	Identifier	
		С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_ CENTER	
		.NET	InstrumentSetting.FrequencyCenter	
		COM	IviSpecAnInstrumentSettingFrequencyCenter	
Frequency Span		1 2 1	an with the Marker Position attribute. This operation is only ype attribute is set to Delta.	
		С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_ SPAN	
		.NET	InstrumentSetting.FrequencySpan	
		COM	IviSpecAnInstrumentSettingFrequencySpan	
Frequency Start	Sets t	he Frequency S	tart attribute with the Marker Position attribute.	
		С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_ START	
		.NET	InstrumentSetting.FrequencyStart	
		COM	IviSpecAnInstrumentSettingFrequencyStart	
Frequency Stop	Sets t	he Frequency S	top attribute with the Marker Position attribute.	
		С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_ STOP	
		.NET	InstrumentSetting.FrequencyStop	
		COM	IviSpecAnInstrumentSettingFrequencyStop	
Reference Level	el Sets the Reference Level attribute with the Marker Amplitude attribute.			
		С	IVISPECAN_VAL_INSTRUMENT_SETTING_REFERENCE_ LEVEL	
		.NET	InstrumentSetting.ReferenceLevel	
		COM	IviSpecAnInstrumentSettingReferenceLevel	

### **Compliance Notes**

- 1. If an IVI class driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

## 6.4 IviSpecAnMarker Behavior Model

The marker attributes may be used at any time during the course of the operation of the spectrum analyzer. When signal tracking is enabled, the effects are the same as calling the Set Instrument From Marker function with the InstrumentSetting parameter set to FrequencyCenter: it keeps the signal peak at the center of the trace. The adjustment of attributes necessary to center the peak is done in the calculation block.

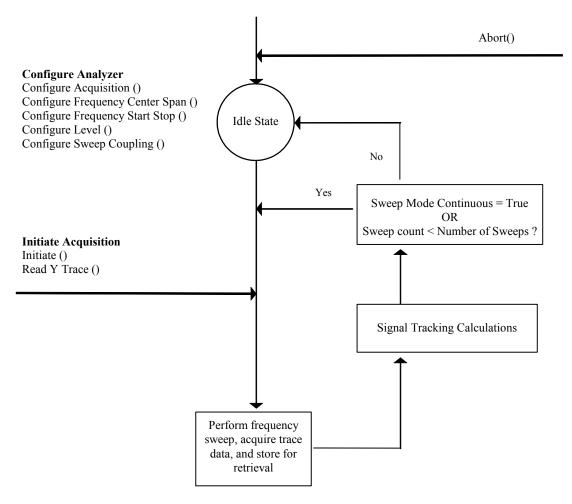


Figure 6-2: IviSpecAnMarker Behavior Model

# 6.5 IviSpecAnMarker Compliance Notes

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

# 7 IviSpecAnTrigger Extension Group

# 7.1 IviSpecAnTrigger Extension Group Overview

This extension group specifies the source of the trigger signal that causes the analyzer to leave the *Wait-For-Trigger* state.

# 7.2 IviSpecAnTrigger Attributes

The IviSpecAnTrigger capability group defines the following attributes:

Trigger Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

## 7.2.1 Trigger Source

Data Type Access		Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Trigger Source (IVI-C Only)
ViString (.NET)	R/W	N/A	None	None

#### .NET Property Name

Trigger.Source

#### **COM Property Name**

Trigger.Source

#### **COM Enumeration Name**

IviSpecAnTriggerSourceEnum

#### **C** Constant Name

IVISPECAN ATTR TRIGGER SOURCE

#### Description

Specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state.

#### **Defined Values**

In IVI.NET the Trigger Source is a string. If an IVI driver supports an Trigger Source and the Trigger Source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that Trigger Source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new Trigger Source strings for Trigger Source that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

If a driver implements the Video trigger, it shall accept the string "Video" for the video trigger.

Name	Description			
		Language	Identifier	
External	The spectrum analyzer waits until it receives a trigger on the external trigger connector.			
	C		IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNAL	
		COM	IviSpecAnTriggerSourceExternal	
Immediate	The spectrum analyzer does not wait for a trigger of any kind.			
		С	IVISPECAN_VAL_TRIGGER_SOURCE_IMMEDIATE	
		COM	IviSpecAnTriggerSourceImmediate	

Software	exe	The spectrum analyzer waits until the Send Software Trigger function executes. For more information, refer to Section 2, <i>Software Triggering Capability</i> of the <i>Standard Cross-Class Capabilities Specification</i> .				
		C IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE				
		COM	IviSpecAnTriggerSourceSoftware			
AC Line	The spectrum analyzer waits until it receives a trigger on the AC line.					
		C IVISPECAN_VAL_TRIGGER_SOURCE_AC_LINE				
		COM	IviSpecAnTriggerSourceACLine			
Video	Th	The spectrum analyzer waits until it receives a video level.				
		C IVISPECAN_VAL_TRIGGER_SOURCE_VIDEO				
		COM IviSpecAnTriggerSourceVideo				

### .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 7.3 IviSpecAnTrigger Functions

The IviSpecAnTrigger Extension group includes the following functions:

Configure Trigger Source (IVI-C Only)

This section describes the behavior and requirements of each function.

# 7.3.1 Configure Trigger Source (IVI-C Only)

### Description

This function specifies the trigger source that causes the spectrum analyzer to leave the *Wait-for-Trigger* state.

## .NET Method Prototype

N/A

(Use the Trigger Source property.)

### **COM Method Prototype**

N/A

(Use the Trigger Source property.)

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
TriggerSource	Specifies the trigger source that causes the analyzer to leave the <i>Wait-For-Trigger</i> state. The driver uses this value to set the Trigger Source attribute. See the attribute description for more details.	ViInt32

### Return Values (C)

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

# 7.4 IviSpecAnTrigger Behavior Model

The IviSpecAnTrigger extension group follows the same behavior model as the IviSpecAnBase capability group described in Section 4.4, *IviSpecAn Behavior Model*, with the addition of a Wait For Trigger state, as shown below.

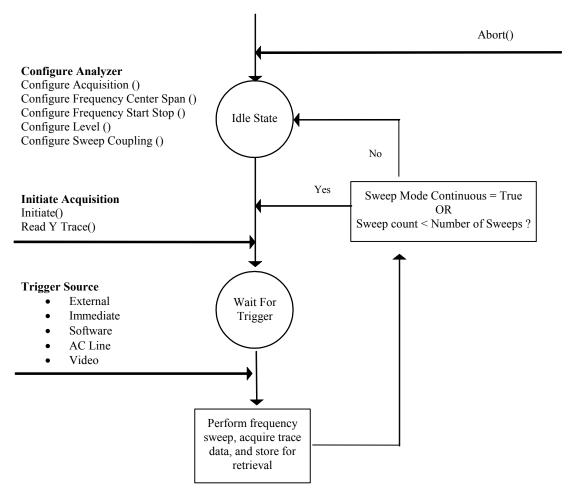


Figure 7-1: IviSpecAnTrigger Behavior Model

# 7.5 IviSpecAnTrigger Compliance Notes

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

# 8 IviSpecAnExternalTrigger Extension Group

# 8.1 IviSpecAnExternalTrigger Extension Group Overview

This extension group specifies the external trigger level and external trigger slope when the Trigger Source Attribute is set to external, which causes the analyzer to leave the *Wait-For-Trigger* state.

## 8.2 IviSpecAnExternalTrigger Attributes

The IviSpecAnExternalTrigger capability group defines the following attributes:

External Trigger Level

External Trigger Slope

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 8.2.1 External Trigger Level

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure External Trigger

## **Property Name**

Trigger.External.Level

### **COM Property Name**

Trigger.External.Level

#### **COM Enumeration Name**

N/A

#### **C Constant Name**

IVISPECAN\_ATTR\_EXTERNAL\_TRIGGER\_LEVEL

### Description

Specifies the level, in Volts, that the external trigger signal shall reach to trigger the acquisition..

### .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 External Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure External Trigger

### .NET Property Name

Trigger.External.Slope

#### .NET Enumeration Name

Slope

# **COM Property Name**

Trigger.External.Slope

#### **COM Enumeration Name**

IviSpecAnExternalTriggerSlopeEnum

### **C** Constant Name

IVISPECAN ATTR EXTERNAL TRIGGER SLOPE

### **Description**

Specifies which slope of the external trigger signal triggers the acquisition.

#### **Defined Values**

Name	De	Description		
		Language	Identifier	
Positive	Spo	pecifies positive slope.		
		C	IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_POSITIVE	
		.NET Slope.Positive		
		COM	IviSpecAnExternalTriggerSlopePositive	
Negative	Spo	Specifies negative slope.		
		C IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_NEGATI		
		.NET	Slope.Negative	
		COM	IviSpecAnExternalTriggerSlopeNegative	

# .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 8.3 IviSpecAnExternalTrigger Functions

The IviSpecAnExternalTrigger Extension group includes the following functions:

Configure External Trigger

This section describes the behavior and requirements of each function.

## 8.3.1 Configure External Trigger

#### **Description**

This function specifies at which level and slope of the external trigger signal, acquisition is triggered. This is applicable when the Trigger Source attribute is set to External.

### .NET Method Prototype

### **COM Method Prototype**

```
HRESULT Trigger.External.Configure ([in] DOUBLE ExternalTriggerLevel,
[in] IviSpecAnExternalTriggerSlopeEnum ExternalTriggerSlope);
```

#### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ExternalTriggerLevel	Specifies the level, in volts, that the external trigger signal shall reach to trigger the acquisition. The driver uses this value to set the External Trigger Level attribute. See the attribute description for more details.	ViReal64
ExternalTriggerSlope	Specifies which slope of the external trigger signal triggers the acquisition. The driver uses this value to set the External Trigger Slope attribute. See the attribute description for more details.	ViInt32

#### 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.

# 8.4 IviSpecAnExternalTrigger Behavior Model

The IviSpecAnExternalTrigger extension group follows the same behavior model as the IviSpecAnTrigger extension group described in Section 7.4, *IviSpecAnTrigger Behavior Model*.

# 8.5 IviSpecAnExternalTrigger Compliance Notes

For a specific driver to comply with the IviSpecAnExternalTrigger extension it shall comply with the IviSpecAnTrigger Extension group, support the External value for the Trigger Source attribute, and implement all the attributes and functions listed in this section.

# 9 IviSpecAnSoftwareTrigger Extension Group

### 9.1 IviSpecAnSoftwareTrigger Overview

The IviSpecAnSoftwareTrigger Extension Group supports spectrum analyzers that can acquire traces based on a software trigger signal. The user can send a software trigger to cause signal output to occur.

This extension affects instrument behavior when the Trigger Source attribute is set to Software.

## 9.2 IviSpecAnSoftwareTrigger Functions

The IviSpecAnSoftwareTrigger extension defines the following functions:

Send Software Trigger

This section describes the behavior and requirements of this function.

## 9.2.1 Send Software Trigger

Refer to *IVI-3.3: Standard Cross-Class Capabilities Specification* for the prototype and complete description of this function.

## 9.3 IviSpecAnSoftwareTrigger Behavior Model

The IviSpecAnSoftwareTrigger extension group follows the same behavior model as the IviSpecAnTrigger extension group described in Section 7.4, *IviSpecAnTrigger Behavior Model*.

# 9.4 IviSpecAnSoftwareTrigger Compliance Notes

For a specific driver to comply with the IviSpecAnSoftwareTrigger extension it shall comply with the IviSpecAnTrigger Extension group, support the External value for the Trigger Source attribute, and implement all the attributes and functions listed in this section.

# 10 IviSpecAnVideoTrigger Extension Group

# 10.1 IviSpecAnVideoTrigger Extension Group Overview

This extension group specifies the video trigger level and video trigger slope when the Trigger Source attribute is set to Video, which causes the analyzer to leave the *Wait-For-Trigger* state.

## 10.2 IviSpecAnVideoTrigger Attributes

The IviSpecAnVideoTrigger capability group defines the following attributes:

Video Trigger Level

Video Trigger Slope

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 10.2.1 Video Trigger Level

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViReal64	R/W	N/A	None	Configure Video Trigger

### .NET Property Name

Trigger.Video.Level

### **COM Property Name**

Trigger.Video.Level

#### **C Constant Name**

IVISPECAN\_ATTR\_VIDEO\_TRIGGER\_LEVEL

### Description

Specifies the level that the video signal shall reach to trigger the acquisition. The units are specified by the Amplitude Units 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.

# 10.2.2 Video Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Video Trigger

### .NET Property Name

Trigger.Video.Slope

#### .NET Enumeration Name

Slope

### **COM Property Name**

Trigger.Video.Slope

#### **COM Enumeration Name**

IviSpecAnVideoTriggerSlopeEnum

### **C** Constant Name

IVISPECAN\_ATTR\_VIDEO\_TRIGGER\_SLOPE

### **Description**

Specifies which slope of the video signal triggers the acquisition.

#### **Defined Values**

Name	De	scription	
		Language	Identifier
Positive	Specifies positive		e slope.
		С	IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POSITIVE
		.NET	Slope.Positive
		COM	IviSpecAnVideoTriggerSlopePositive
Negative	Spe	ecifies negativ	ve slope.
		С	IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_NEGATIVE
		.NET	Slope.Negative
		COM	IviSpecAnVideoTriggerSlopeNegative

### .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 10.3 IviSpecAnVideoTrigger Functions

The IviSpecAnVideoTrigger Extension group includes the following functions:

Configure Video Trigger

This section describes the behavior and requirements of each function.

## 10.3.1 Configure Video Trigger

#### Description

This function specifies at which level and slope of the video signal, acquisition is triggered. This is applicable when the Trigger Source attribute is set to Video.

### .NET Method Prototype

### **COM Method Prototype**

```
HRESULT Trigger.Video.Configure ([in] DOUBLE VideoTriggerLevel,
[in] IviSpecAnVideoTriggerSlopeEnum VideoTriggerSlope);
```

#### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
VideoTriggerLevel	Specifies the level that the video signal shall reach to trigger the acquisition. The driver uses this value to set the Video Trigger Level attribute. See the attribute description for more details.	ViReal64
VideoTriggerSlope	Specifies which slope of the video signal triggers the acquisition. The driver uses this value to set the Video Trigger Slope attribute. See the attribute description for more details.	ViInt32

#### 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.

# 10.4 IviSpecAnVideoTrigger Behavior Model

The IviSpecAnVideoTrigger extension group follows the same behavior model as the IviSpecAnTrigger extension group described in Section 7.4, *IviSpecAnTrigger Behavior Model*.

# 10.5 IviSpecAnVideoTrigger Compliance Notes

For a specific driver to comply with the IviSpecAnVideoTrigger extension group it shall comply with the IviSpecAnTrigger Extension group, support the value Video for the Trigger Source attribute, and implement all the attributes and functions listed in this section.

# 11 IviSpecAnDisplay Extension Group

# 11.1 IviSpecAnDisplay Extension Group Overview

The IviSpecAnDisplay extension group controls the display related attributes.

# 11.2 IviSpecAnDisplay Attributes

The IviSpecAnDisplay capability group defines the following attributes:

Number Of Divisions

Units Per Division

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

## 11.2.1 Number Of Divisions

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

### .NET Property Name

Display.NumberOfDivisions

### **COM Property Name**

Display.NumberOfDivisions

#### **C** Constant Name

IVISPECAN\_ATTR\_NUMBER\_OF\_DIVISIONS

## Description

Specifies the number of vertical screen divisions.

### .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 Units Per Division

Data 7	Гуре	Access	Applies to	Coercion	High Level Functions
ViReal	.64	R/W	N/A	None	N/A

#### .NET Property Name

Display.UnitsPerDivision

#### **COM Property Name**

Display.UnitsPerDivision

#### **C** Constant Name

IVISPECAN ATTR UNITS PER DIVISION

#### Description

Specifies the number of vertical units in one screen division. This attribute is typically used in conjunction with the Reference Level attribute to set the vertical range of the spectrum analyzer.

#### .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.

# 11.3 IviSpecAnDisplay Behavior Model

The IviSpecAnDisplay extension group follows the same behavior model as the IviSpecAnBase capability group described in Section 0, *IviSpecAnBase Behavior Model*.

## 11.4 IviSpecAnDisplay Compliance Notes

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

# 12 IviSpecAnMarkerType Extension Group

# 12.1 IviSpecAnMarkerType Extension Group Overview

This section supports analyzers that have multiple marker types.

# 12.2 IviSpecAnMarkerType Attributes

The IviSpecAnMarkerType capability group defines the following attribute:

Marker Type

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 12.2.1 Marker Type

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Marker	None	Query Marker Type

### .NET Property Name

Marker. Type

#### .NET Enumeration Name

MarkerType

### **COM Property Name**

Marker.Type

### **COM Enumeration Name**

IviSpecAnMarkerTypeEnum

### **C** Constant Name

IVISPECAN\_ATTR\_MARKER\_TYPE

### **Description**

Specifies the marker type of the active marker.

#### **Defined Values**

Name	Description		
		Language	Identifier
Normal	Re	gular marker ı	used to make absolute measurements.
		C	IVISPECAN_VAL_MARKER_TYPE_NORMAL
		.NET	MarkerType.Normal
		COM	IviSpecAnMarkerTypeNormal
Delta	Marker used in conjunction with the reference marker to mal relative measurements.		
		С	IVISPECAN_VAL_MARKER_TYPE_DELTA
		.NET	MarkerType.Delta
		COM	IviSpecAnMarkerTypeDelta

## .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 class driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.
- 2. If an IVI specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

# 12.3 IviSpecAnMarkerType Functions

The IviSpecAnMarkerType Extension group includes the following functions:

Query Marker Type (IVI-C Only)

This section describes the behavior and requirements of each function.

# 12.3.1 Query Marker Type (IVI-C Only)

## Description

This function returns the type of the active marker.

## .NET Method Prototype

```
N/A (Use the Marker Type property.)
```

## **COM Method Prototype**

```
{\mathbb N}/{\mathbb A} (Use the Marker Type property.)
```

## **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Outputs	Description	Base Type
MarkerType	Returns the type of the active marker.	ViInt32

## Return Values (C)

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

## 12.4 IviSpecAnMarkerType Behavior Model

The IviSpecAnMarkerType extension group follows the same behavior model as the IviSpecAnMarker capability group described in Section 0, *Compliance Notes* 

If an IVI class driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.

3. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

IviSpecAnMarker Behavior Model.

## 12.5 IviSpecAnMarkerType Compliance Notes

For a specific driver to comply with the IviSpecAnMarkerType extension it shall be compliant with the IviSpecAnMarker Extension group and it shall implement all of the attributes and functions listed in this section.

# 13 IviSpecAnDeltaMarker Extension Group

# 13.1 IviSpecAnDeltaMarker Extension Group Overview

This section supports analyzers that have delta-marker capabilities. A delta marker has the same properties as a normal marker except that its position and amplitude are relative to a fixed reference point. This reference point is defined when the marker is converted from a normal marker to a delta marker.

### 13.2 IviSpecAnDeltaMarker Attributes

The IviSpecAnDelta capability group defines the following attributes:

Reference Marker Amplitude

.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.

The table below specifies additional class-defined exceptions for this method.

<b>Exception Class</b>	Description
NotDeltaMarkerException	The marker type is not Delta.

Reference Marker Position

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 13.2.1 Reference Marker Amplitude

	Data Type	Access	Applies to	Coercion	High Level Functions
ĺ	ViReal64	RO	Marker	None	Query Reference Marker

### .NET Property Name

Marker.ReferenceAmplitude

### **COM Property Name**

Marker.ReferenceAmplitude

#### **C Constant Name**

IVISPECAN\_ATTR\_REFERENCE\_MARKER\_AMPLITUDE

### Description

Specifies the reference marker amplitude when the active marker is a delta marker. The unit is given by the Amplitude Units attribute.

If the Marker Type attribute is not Delta, this property returns the Not Delta Marker error.

### .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.

The table below specifies additional class-defined exceptions for this method.

<b>Exception Class</b>	Description
NotDeltaMarkerException	The marker type is not Delta.

### 13.2.2 Reference Marker Position

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Marker	None	Query Reference Marker

### .NET Property Name

Marker.ReferencePosition

### **COM Property Name**

Marker.ReferencePosition

#### **C Constant Name**

IVISPECAN\_ATTR\_REFERENCE\_MARKER\_POSITION

### Description

Specifies the position of the reference marker, when the active marker is a delta marker. The units are Hertz for frequency domain measurements, and seconds for time domain measurements.

If the Marker Type attribute is not Delta, this property returns the Not Delta Marker error.

### .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.

The table below specifies additional class-defined exceptions for this method.

<b>Exception Class</b>	Description
NotDeltaMarkerException	The marker type is not Delta.

# 13.3 IviSpecAnDeltaMarker Functions

The IviSpecAnDeltaMarker Extension group includes the following functions:

Make Marker Delta

.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.

Query Reference Marker

This section describes the behavior and requirements of each function.

#### 13.3.1 Make Marker Delta

### Description

This function specifies whether the active marker is a delta marker.

When the DeltaMarker parameter is True, the Marker Type attribute is set to Delta. The Reference Marker Amplitude and Reference Marker Position attributes are set with the active marker current Marker Amplitude and Marker Position attribute respectively.

While the marker remains a delta marker, the values of the Marker Amplitude and Marker Position attributes will be relative to the Reference Marker Amplitude and Reference Marker Position attributes, respectively.

When the DeltaMarker parameter is False, the active marker is changed to a normal marker. The Marker Type attribute is set to Normal. The Marker Amplitude and Marker Position attributes are returned to a normal state and are no longer relative to the Reference Marker Amplitude and Reference Marker Position attributes.

### .NET Method Prototype

```
void Marker.MakeDelta (Boolean deltaMarker);
```

### **COM Method Prototype**

```
HRESULT Marker.MakeDelta ([in] VARIANT BOOL DeltaMarker);
```

#### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
DeltaMarker	Enables or disables delta marker mode. The driver uses this value to set the Marker Type attribute. See the attribute description for more details.	ViBoolean

### Return Values (C/COM)

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

#### .NET Exceptions

# 13.3.2 Query Reference Marker

### Description

This function returns the amplitude and position of the reference marker. If the Marker Type attribute is not Delta, this function returns the Not Delta Marker error.

### .NET Method Prototype

```
public struct MarkerInfo {
    public Double Position {};
    public Double Amplitude {};
}

MarkerInfo Marker.QueryReference ();
```

### **COM Method Prototype**

```
HRESULT Marker.QueryReference ([in, out] DOUBLE *ReferenceAmplitude [in, out] DOUBLE *ReferencePosition);
```

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession

Outputs	Description	Base Type
ReferenceMarkerAmpl itude (C/COM)	Returns the Reference Marker Amplitude attribute value. See the attribute description for more details.	ViReal64
ReferenceMarkerPosi tion (C/COM)	Returns the Reference Marker Position attribute value. See the attribute description for more details.	ViReal64
Return value (.NET)	Returns the Reference Marker Amplitude and Position attribute values. See the attribute descriptions for more details.	MarkerInfo

### Return Values (C/COM)

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

### .NET Exceptions

The table below specifies additional class-defined exceptions for this method.

<b>Exception Class</b>	Description
NotDeltaMarkerException	The marker type is not Delta.

# 13.4 IviSpecAnDeltaMarker Behavior Model

The IviSpecAnDeltaMarker extension group follows the same behavior model as the IviSpecAnMarker capability group described in Section 0, *Compliance Notes* 

If an IVI class driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Vendor Defined Class Value Extension Base and less than IviSpecAn Specific Driver Value Extension Base. See Section 17.1 IviSpecAn Attribute Value Extension Bases for more information.

4. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviSpecAn Specific Driver Value Extension Base. See Section 17.1 *IviSpecAn Attribute Value Extension Bases* for more information.

IviSpecAnMarker Behavior Model.

### 13.5 IviSpecAnDeltaMarker Compliance Notes

For a specific driver to comply with the IviSpecAnDeltaMarker extension it shall be compliant with the IviSpecAnMarkerType Extension group and it shall implement all the attributes and functions listed in this section.

# 14 IviSpecAnExternalMixer Extension Group

# 14.1 IviSpecAnExternalMixer Extension Group Overview

Many spectrum analyzers have outputs and inputs that allow external equipment to use the IF or mixer signal that the spectrum analyzer uses. In this case, external equipment can be used to mix signals to convert them to measurable frequencies. This allows the use of an analyzer to measure values that are outside of the normal frequency range of the equipment. When using an external mixer, many of the settings of the analyzer have to be carefully converted to allow the user to know what is meant by the values read. Specifically, the frequency, the harmonic number, mixer configuration, and conversion loss must be configured carefully to be able to use the external mixing successfully.

The frequency of the input signal can be expressed as a function of the local oscillator (LO) frequency and the selected harmonic of the 1st LO is as follows:

$$f_{in} = n * f_{LO} +/- f_{IF}$$

Where: f<sub>in</sub> frequency of input signal

n order of harmonic used for conversion

 $f_{LO}$  frequency of 1st LO

f<sub>IF</sub> intermediate frequency

The Harmonic number defines the order n of the harmonic used for conversion. Both even and odd harmonics can be used. The selected harmonic, together with the setting range of the 1st LO, determines the limits of the settable frequency range. The following applies:

Lower frequency limit:  $f_{min} = n * f_{LO,min} - f_{IF}$ 

Upper frequency limit:  $f_{max} = n * f_{LO,max} + f_{IF}$ 

Where:  $f_{LO min}$  lower frequency limit of LO

f<sub>LO,max</sub> upper frequency limit of LO

The following sections describe the mixer configuration and the conversion loss table configuration.

# 14.1.1 Mixer Configuration

The external mixers are typically configured either as two-port or three port mixers. Single-diode mixers generally require a DC voltage which is applied via the LO line. This DC voltage is to be tuned to the minimum conversion loss versus frequency. Some instruments can define a limit for the BIAS current.

The two-port mixer connects the 'LO OUT / IF IN' output of the analyzer to the LO/IF port of the external mixer. The diplexer is contained in the analyzer and the IF signal can be tapped from the line which is used to feed the LO signal to the mixer. The signal to be measured is fed to the RF input of the external mixer.

On the other hand, the three-port mixer connects the 'LO OUT / IF IN' output of the analyzer to the LO port of the external mixer. The 'IF IN' input of the analyzer is connected to the IF port of the external mixer. The signal to be measured is fed to the RF input of the external mixer.

#### 14.1.2 Conversion Loss

The maximum settable reference level depends on the external mixer's conversion loss which is defined by average conversion loss or by using the conversion loss table. For example, if an IF signal with a level of -20 dBm is applied to the LO OUT / IF IN or IF IN input of the spectrum analyzer, full screen level is attained. Consequently, the maximum settable reference level is -20 dBm at a set conversion loss of 0 dB. If a conversion loss > 0 dB is entered, the maximum settable reference level increases in the same proportion. If the maximum possible reference level is set on the analyzer, this level is reduced if a smaller conversion loss is entered.

In addition to the dynamic range of the spectrum analyzer the 1 dB compression point of the mixer has to be taken into account. The levels of the input signals lie well below this value to avoid generation of harmonics of these signals in the mixer. These are converted by the LO signals harmonics of higher order and appear in the displayed spectrum.

Some instruments allow the definition of conversion loss tables. The Conversion loss table allows the conversion loss of the mixer in the selected band to be taken into account as a function of frequency. Correction values for frequencies between the individual reference values are obtained by interpolation (Linear interpolation). Outside the frequency range covered by the table the conversion loss is assumed to be the same as that for the reference value marking the table limit.

### 14.2 IviSpecAnExternalMixer Attributes

The IviSpecAnExternalMixer capability group defines the following attributes:

External Mixer Average Conversion Loss

External Mixer Bias

External Mixer Bias Enabled

External Mixer Bias Limit

External Mixer Conversion Loss Table Enabled

External Mixer Enabled

External Mixer Harmonic

External Mixer Number Of Ports

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviSpecAn Attribute ID Definitions*.

# 14.2.1 External Mixer Average Conversion Loss

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure External Mixer

### .NET Property Name

ExternalMixer.AverageConversionLoss

### **COM Property Name**

ExternalMixer.AverageConversionLoss

#### **C** Constant Name

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_AVERAGE\_CONVERSION\_LOSS

### Description

Specifies the average conversion loss.

### .NET Exceptions

# 14.2.2 External Mixer Bias

	Data Type	Access	Applies to	Coercion	High Level Functions
Ī	ViReal64	R/W	N/A	None	Configure External Mixer Bias

# .NET Property Name

ExternalMixer.Bias.Level

### **COM Property Name**

ExternalMixer.Bias.Level

#### **C** Constant Name

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_BIAS

# Description

Specifies the bias current in Amps.

# .NET Exceptions

### 14.2.3 External Mixer Bias Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure External Mixer Bias Enabled

### .NET Property Name

ExternalMixer.Bias.Enabled

### **COM Property Name**

ExternalMixer.Bias.Enabled

### **C** Constant Name

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_BIAS\_ENABLED

### **Description**

If set to True, the external mixer bias is enabled. If set to False, the external mixer bias is disabled.

### .NET Exceptions

### 14.2.4 External Mixer Bias Limit

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure External Mixer Bias

# .NET Property Name

ExternalMixer.Bias.Limit

### **COM Property Name**

ExternalMixer.Bias.Limit

#### **C** Constant Name

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_BIAS\_LIMIT

# Description

Specifies the bias current limit in Amps.

# .NET Exceptions

# 14.2.5 External Mixer Conversion Loss Table Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Conversion Loss Table Enabled

### .NET Property Name

ExternalMixer.ConversionLossTable.Enabled

### **COM Property Name**

ExternalMixer.ConversionLossTable.Enabled

#### **C Constant Name**

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_CONVERSION\_LOSS\_TABLE\_ENABLED

### Description

If set to True, the conversion loss table is enabled. If set to False, the conversion loss table is disabled.

### .NET Exceptions

# 14.2.6 External Mixer Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure External Mixer Enabled

### .NET Property Name

ExternalMixer.Enabled

### **COM Property Name**

ExternalMixer.Enabled

#### **C Constant Name**

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_ENABLED

# Description

If set to True, the external mixer is enabled. If set to False, the external mixer is disabled.

### .NET Exceptions

# 14.2.7 External Mixer Harmonic

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure External Mixer

# .NET Property Name

ExternalMixer.Harmonic

### **COM Property Name**

ExternalMixer.Harmonic

### **C** Constant Name

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_HARMONIC

# Description

Specifies the order n of the harmonic used for conversion.

# .NET Exceptions

# 14.2.8 External Mixer Number Of Ports

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure External Mixer Number of Ports

# .NET Property Name

ExternalMixer.NumberOfPorts

### **COM Property Name**

ExternalMixer.NumberOfPorts

#### **C Constant Name**

IVISPECAN\_ATTR\_EXTERNAL\_MIXER\_NUMBER\_OF\_PORTS

# Description

Specifies the number of mixer ports.

### .NET Exceptions

# 14.3 IviSpecAnExternalMixer Functions

The IviSpecAnExternalMixer Extension group includes the following functions:

Configure Conversion Loss Table

Configure Conversion Loss Table Enabled (IVI-C Only)

Configure External Mixer

Configure External Mixer Bias

Configure External Mixer Bias Enabled (IVI-C Only)

Configure External Mixer Enabled (IVI-C Only)

Configure External Mixer Number Of Ports (IVI-C Only)

This section describes the behavior and requirements of each function.

# 14.3.1 Configure Conversion Loss Table

### Description

This function configures the conversion loss table by specifying a series of frequency and a power loss pairs.

### .NET Method Prototype

### **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Count	Specifies the number of frequency and conversion loss pairs.	
Frequency[]	Specifies the frequency values for the pairs. The array must have at least as many elements as the value in the Count parameter (IVI-C only).	ViReal64
ConversionLoss[]	Specifies the conversion loss values for the pairs. The array must have at least as many elements as the value in the Count parameter (IVI-C only).	ViReal64

### Return Values (C/COM)

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

### .NET Exceptions

# 14.3.2 Configure Conversion Loss Table Enabled (IVI-C Only)

### Description

This function enables or disables the conversion loss table.

### .NET Method Prototype

N/A

(Use the External Mixer Conversion Loss Table Enabled property.)

# **COM Method Prototype**

N/A

(Use the External Mixer Conversion Loss Table Enabled property.)

### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ConversionLossTableEn abled	Enables or disables the conversion loss table. The driver uses this value to set the External Mixer Conversion Loss Table Enabled attribute. See the attribute description for more details.	ViBoolean

### **Return Values (C)**

# 14.3.3 Configure External Mixer

### Description

This function specifies the mixer harmonic and average conversion loss.

### .NET Method Prototype

### **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Harmonic	Specifies the harmonic number. The driver uses this value to set the External Mixer Harmonic attribute. See the attribute description for more details.	ViInt32
AverageConversionLoss	Specifies the average conversion loss. The driver uses this value to set the External Mixer Average Conversion Loss 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

# 14.3.4 Configure External Mixer Bias

### Description

This function configures the external mixer bias and the external mixer bias limit.

### .NET Method Prototype

### **COM Method Prototype**

### **C Function Prototype**

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Bias	Specifies the bias current. The driver uses this value to set the External Mixer Bias attribute. See the attribute description for more details.	ViReal64
BiasLimit	Specifies the bias current limit. The driver uses this value to set the External Mixer Bias Limit 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

# 14.3.5 Configure External Mixer Bias Enabled (IVI-C Only)

# Description

This function enables or disables the external mixer's bias.

### .NET Method Prototype

N/A

(Use the External Mixer Bias Enabled property.)

# **COM Method Prototype**

N/A

(Use the External Mixer Bias Enabled property.)

### **C** Function Prototype

ViStatus IviSpecAn\_ConfigureExternalMixerBiasEnabled (ViSession Vi, ViBoolean BiasEnabled);

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
BiasEnabled	Enables or disables the external mixer's bias. The driver uses this value to set the External Mixer Bias Enabled attribute. See the attribute description for more details.	ViBoolean

### **Return Values (C)**

# 14.3.6 Configure External Mixer Enabled (IVI-C Only)

### Description

This function enables or disables the external mixer.

# .NET Method Prototype

N/A

(Use the External Mixer Enabled property.)

# **COM Method Prototype**

N/A

(Use the External Mixer Enabled property.)

### **C** Function Prototype

#### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ExternalMixerEna bled	Enables or disables the external mixer. The driver uses this value to set the External Mixer Enabled attribute. See the attribute description for more details.	ViBoolean

# Return Values (C)

# 14.3.7 Configure External Mixer Number Of Ports (IVI-C Only)

# Description

This function specifies the number of external mixer ports.

### .NET Method Prototype

N/A

(Use the External Mixer Number Of Ports property.)

# **COM Method Prototype**

N/A

(Use the External Mixer Number Of Ports property.)

### **C** Function Prototype

ViStatus IviSpecAn\_ConfigureExternalMixerNumberOfPorts (ViSession Vi, ViInt32 NumberOfPorts);

### **Parameters**

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
NumberOfPorts	Specifies the number of ports used by the external mixer. The driver uses this value to set the External Mixer Number Of Ports attribute. See the attribute description for more details.	ViInt32

### **Return Values (C)**

# 14.4 IviSpecAnExternalMixing Behavior Model

The IviSpecAnExternalMixing extension group follows the same behavior model as the IviSpecAnBase capability group described in Section 0, *IviSpecAnBase Behavior Model*.

# 14.5 IviSpecAnExternalMixing Compliance Notes

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

# 15 IviSpecAnPreselector Extension Group

# 15.1 IviSpecAnPreselector Extension Group Overview

The IviSpecAnPreselector extension controls preselectors. Preselectors are a network of filters and preamplifiers that are built into one unit for reducing noise and increasing dynamic range of an analyzer. Preselectors are often separate instruments, but they are instruments that only work with spectrum analyzers. Some analyzers have internal preselectors.

# 15.2 IviSpecAnPreselector Functions

The IviSpecAnPreselector Extension group includes the following functions:

Peak Preselector

This section describes the behavior and requirements of each function.

### 15.2.1 Peak Preselector

### Description

This function adjusts the preselector to obtain the maximum readings for the current start and stop frequency. This function may affect the marker configuration.

### .NET Method Prototype

```
void Preselector.Peak ();
```

### **COM Method Prototype**

```
HRESULT Preselector.Peak ();
```

### **C Function Prototype**

ViStatus IviSpecAn PeakPreselector (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

# 15.3 IviSpecAnPreselector Behavior Model

The IviSpecAnPreselector extension group follows the same behavior model as the IviSpecAnBase capability group described in Section 0, *IviSpecAnBase Behavior Model*.

# 15.4 IviSpecAnPreselector Compliance Notes

For a specific driver to comply with the IviSpecAnPreselector extension it shall be compliant with the IviSpecAnBase capability group and it shall implement the function listed in this section.

# 16 IviSpecAn Attribute ID Definitions

The following table defines the ID value for all IviSpecAn class attributes.

Table 16-1. IviSpecAn Attributes ID Values

Attribute Name	ID Definition
IVISPECAN_ATTR_AMPLITUDE_UNITS	IVI_CLASS_ATTR_BASE + 1
IVISPECAN_ATTR_ATTENUATION	IVI_CLASS_ATTR_BASE + 2
IVISPECAN_ATTR_ATTENUATION_AUTO	IVI_CLASS_ATTR_BASE + 3
IVISPECAN_ATTR_DETECTOR_TYPE	IVI_CLASS_ATTR_BASE + 4
IVISPECAN_ATTR_DETECTOR_TYPE_AUTO	IVI_CLASS_ATTR_BASE + 5
IVISPECAN_ATTR_FREQUENCY_START	IVI_CLASS_ATTR_BASE + 6
IVISPECAN_ATTR_FREQUENCY_STOP	IVI_CLASS_ATTR_BASE + 7
IVISPECAN_ATTR_FREQUENCY_OFFSET	IVI_CLASS_ATTR_BASE + 8
IVISPECAN_ATTR_INPUT_IMPEDANCE	IVI_CLASS_ATTR_BASE + 9
IVISPECAN_ATTR_NUMBER_OF_SWEEPS	IVI_CLASS_ATTR_BASE + 10
IVISPECAN_ATTR_REFERENCE_LEVEL	IVI_CLASS_ATTR_BASE + 11
IVISPECAN_ATTR_REFERENCE_LEVEL_OFFSET	IVI_CLASS_ATTR_BASE + 12
IVISPECAN_ATTR_RESOLUTION_BANDWIDTH	IVI_CLASS_ATTR_BASE + 13
IVISPECAN_ATTR_RESOLUTION_BANDWIDTH_AUTO	IVI_CLASS_ATTR_BASE + 14
IVISPECAN_ATTR_SWEEP_MODE_CONTINUOUS	IVI_CLASS_ATTR_BASE + 15
IVISPECAN_ATTR_SWEEP_TIME	IVI_CLASS_ATTR_BASE + 16
IVISPECAN_ATTR_SWEEP_TIME_AUTO	IVI_CLASS_ATTR_BASE + 17
IVISPECAN_ATTR_TRACE_COUNT	IVI_CLASS_ATTR_BASE + 18
IVISPECAN_ATTR_TRACE_SIZE	IVI_CLASS_ATTR_BASE + 19
IVISPECAN_ATTR_TRACE_TYPE	IVI_CLASS_ATTR_BASE + 20
IVISPECAN_ATTR_VERTICAL_SCALE	IVI_CLASS_ATTR_BASE + 21
IVISPECAN_ATTR_VIDEO_BANDWIDTH	IVI_CLASS_ATTR_BASE + 22
IVISPECAN_ATTR_VIDEO_BANDWIDTH_AUTO	IVI_CLASS_ATTR_BASE + 23
IVISPECAN ATTR ACTIVE MARKER	IVI CLASS ATTR BASE + 201
	IVI_CLASS_ATTR_BASE + 202
IVISPECAN_ATTR_MARKER_COUNT	IVI_CLASS_ATTR_BASE + 203
IVISPECAN_ATTR_MARKER_ENABLED	IVI_CLASS_ATTR_BASE + 204
IVISPECAN_ATTR_MARKER_FREQUENCY_COUNTER_ENABLED	IVI_CLASS_ATTR_BASE + 205
IVISPECAN_ATTR_MARKER_FREQUENCY_COUNTER_RESOLUTI ON	IVI_CLASS_ATTR_BASE + 206

Table 16-1. IviSpecAn Attributes ID Values

Attribute Name	ID Definition
IVISPECAN_ATTR_MARKER_POSITION	IVI_CLASS_ATTR_BASE + 207
IVISPECAN_ATTR_MARKER_THRESHOLD	IVI_CLASS_ATTR_BASE + 208
IVISPECAN_ATTR_MARKER_TRACE	IVI_CLASS_ATTR_BASE + 209
IVISPECAN_ATTR_PEAK_EXCURSION	IVI_CLASS_ATTR_BASE + 210
IVISPECAN_ATTR_SIGNAL_TRACK_ENABLED	IVI_CLASS_ATTR_BASE + 211
IVISPECAN_ATTR_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 301
IVISPECAN_ATTR_EXTERNAL_TRIGGER_LEVEL	IVI_CLASS_ATTR_BASE + 401
IVISPECAN_ATTR_EXTERNAL_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 402
IVISPECAN_ATTR_VIDEO_TRIGGER_LEVEL	IVI_CLASS_ATTR_BASE + 501
IVISPECAN_ATTR_VIDEO_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 502
IVISPECAN_ATTR_UNITS_PER_DIVISION  IVISPECAN_ATTR_NUMBER_OF_DIVISIONS	IVI_CLASS_ATTR_BASE + 601 IVI_CLASS_ATTR_BASE + 602
TVIOLECTIV_TITIN_NOTEDIAL_OT_DIVIDIONO	TVI_CEMOS_MITK_EMOE + 002
IVISPECAN_ATTR_MARKER_TYPE	IVI_CLASS_ATTR_BASE + 701
IVISPECAN_ATTR_REFERENCE_MARKER_AMPLITUDE	IVI_CLASS_ATTR_BASE + 801
IVISPECAN_ATTR_REFERENCE_MARKER_POSITION	IVI_CLASS_ATTR_BASE + 802
IVISPECAN_ATTR_EXTERNAL_MIXER_AVERAGE_CONVERSION _LOSS	IVI_CLASS_ATTR_BASE + 901
IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS	IVI_CLASS_ATTR_BASE + 902
IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS_ENABLED	IVI_CLASS_ATTR_BASE + 903
IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS_LIMIT	IVI_CLASS_ATTR_BASE + 904
IVISPECAN_ATTR_EXTERNAL_MIXER_CONVERSION_LOSS_TA BLE_ENABLED	IVI_CLASS_ATTR_BASE + 905
IVISPECAN_ATTR_EXTERNAL_MIXER_ENABLED	IVI_CLASS_ATTR_BASE + 906
IVISPECAN_ATTR_EXTERNAL_MIXER_HARMONIC	IVI_CLASS_ATTR_BASE + 907
IVISPECAN_ATTR_EXTERNAL_MIXER_NUMBER_OF_PORTS	IVI_CLASS_ATTR_BASE + 908

# 17 IviSpecAn Attribute Value Definitions

This section specifies the actual value for each defined attribute value. Each value can be extended using the extension bases defined in Section 17.1, *IviSpecAn Attribute Value Extension Bases*.

# **Amplitude Units**

Value Name	Language	Identifier	Actual Value
dBm	С	IVISPECAN_VAL_AMPLITUDE_UNITS_DBM	1
	.NET	AmplitudeUnits.dBm	0
	COM	IviSpecAnAmplitudeUnitsDBM	1
dBmV	С	IVISPECAN_VAL_AMPLITUDE_UNITS_DBMV	2
	.NET	AmplitudeUnits.dBmV	1
	COM	IviSpecAnAmplitudeUnitsDBMV	2
dBuV	С	IVISPECAN_VAL_AMPLITUDE_UNITS_DBUV	3
	.NET	AmplitudeUnits.dBuV	2
	COM	IviSpecAnAmplitudeUnitsDBUV	3
Volt	С	IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT	4
	.NET	AmplitudeUnits.Volt	3
	COM	IviSpecAnAmplitudeUnitsVolt	4
Watt	С	IVISPECAN_VAL_AMPLITUDE_UNITS_WATT	5
	.NET	AmplitudeUnits.Watt	4
	COM	IviSpecAnAmplitudeUnitsWatt	5

# **Detector Type**

Value Name	Language	Identifier	Actual Value
Auto Peak	С	IVISPECAN_VAL_DETECTOR_TYPE_AUTO_PEAK	1
	.NET	DetectorType.AutoPeak	0
	COM	IviSpecAnDetectorTypeAutoPeak	1
Average	С	IVISPECAN_VAL_DETECTOR_TYPE_AVERAGE	2
	.NET	DetectorType.Average	1
	COM	IviSpecAnDetectorTypeAverage	2
Maximum Peak	С	IVISPECAN_VAL_DETECTOR_TYPE_MAX_PEAK	3
	.NET	DetectorType.MaxPeak	2
	COM	IviSpecAnDetectorTypeMaxPeak	3

Value Name	Language	Identifier	Actual Value
Minimum Peak	С	IVISPECAN_VAL_DETECTOR_TYPE_MIN_PEAK	4
	.NET	DetectorType.MinPeak	3
	COM	IviSpecAnDetectorTypeMinPeak	4
Sample	С	IVISPECAN_VAL_DETECTOR_TYPE_SAMPLE	5
	.NET	DetectorType.Sample	4
	COM	IviSpecAnDetectorTypeSample	5
RMS	С	IVISPECAN_VAL_DETECTOR_TYPE_RMS	6
	.NET	DetectorType.Rms	5
	COM	IviSpecAnDetectorTypeRMS	6

# External Trigger Slope (IVI-C and IVI-COM Only)

Value Name	Language	Identifier	Actual Value
Positive	.NET	Slope.Positive	0
	С	IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_POSIT IVE	1
	COM	IviSpecAnExternalTriggerSlopePositive	1
Negative	.NET	Slope.Negative	1
	С	IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_NEGAT IVE	2
	COM	IviSpecAnExternalTriggerSlopeNegative	2

# Marker Type

Value Name	Language	Identifier	Actual Value
Normal	С	IVISPECAN_VAL_MARKER_TYPE_NORMAL	1
	.NET	MarkerType.Normal	0
	COM	IviSpecAnMarkerTypeNormal	1
Delta	С	IVISPECAN_VAL_MARKER_TYPE_DELTA	2
	.NET	MarkerType.Delta	1
	COM	IviSpecAnMarkerTypeDelta	2

# Trace Type

Value Name	Language	Identifier	Actual Value
Clear Write	С	IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE	1
	.NET	TraceType.ClearWrite	0
	COM	IviSpecAnTraceTypeClearWrite	1
Maximum Hold	С	IVISPECAN_VAL_TRACE_TYPE_MAX_HOLD	2
	.NET	TraceType.MaxHold	1
	COM	IviSpecAnTraceTypeMaxHold	2
Minimum Hold	С	IVISPECAN_VAL_TRACE_TYPE_MIN_HOLD	3
	.NET	TraceType.MinHold	2
	COM	IviSpecAnTraceTypeMinHold	3
Video Average	С	IVISPECAN_VAL_TRACE_TYPE_VIDEO_AVERAGE	4
	.NET	TraceType.VideoAverage	3
	COM	IviSpecAnTraceTypeVideoAverage	4
View	С	IVISPECAN_VAL_TRACE_TYPE_VIEW	5
	.NET	TraceType.View	4
	COM	IviSpecAnTraceTypeView	5
Store	С	IVISPECAN_VAL_TRACE_TYPE_STORE	6
	.NET	TraceType.Store	5
	COM	IviSpecAnTraceTypeStore	6

# Trigger Source (IVI-C & IVI-COM Only)

Value Name	Language	Identifier	Actual Value
External	С	IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviSpecAnTriggerSourceExternal	1
Immediate	С	IVISPECAN_VAL_TRIGGER_SOURCE_IMMEDIATE	2
	COM	IviSpecAnTriggerSourceImmediate	2
Software	С	IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE	3
	COM	IviSpecAnTriggerSourceSoftware	3
AC Line	С	IVISPECAN_VAL_TRIGGER_SOURCE_AC_LINE	4
	COM	IviSpecAnTriggerSourceACLine	4
Video	С	IVISPECAN_VAL_TRIGGER_SOURCE_VIDEO	5
	COM	IviSpecAnTriggerSourceVideo	5

# **Vertical Scale**

Value Name	Language	Identifier	Actual Value
Linear	С	IVISPECAN_VAL_VERTICAL_SCALE_LINEAR	1
	.NET	VerticalScale.Linear	0
	COM	IviSpecAnVerticalScaleLinear	1
Logarithmic	С	IVISPECAN_VAL_VERTICAL_SCALE_LOGARITHMIC	2
	.NET	VerticalScale.Logarithmic	1
	COM	IviSpecAnVerticalScaleLogarithmic	2

### Video Trigger Slope (IVI-C and IVI-COM Only)

Value Name	Language	Identifier	Actual Value
Positive	.NET	Slope.Positive	0
	С	IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POSITIVE	1
	COM	IviSpecAnVideoTriggerSlopePositive	1
Negative	.NET	Slope.Negative	1
	С	IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_NEGATIVE	2
	COM	IviSpecAnVideoTriggerSlopeNegative	2

# 17.1 IviSpecAn Attribute Value Extension Bases

The following extension bases are defined to prevent IviSpecAn Class Specification, vendor defined, and instrument specific attribute values from conflicting. This specification reserves values less than the Vendor Defined Class Value Extension Base for future expansion. The numerical range between the Vendor Defined Class Value Extension Base and the Specific Driver Value Extension Base is reserved for vendor defined values in class drivers. All values greater than the Specific Driver Value Extension Base are reserved for instrument specific values.

Value Name	Language	Identifier	Actual Value
Vendor Defined	С	IVISPECAN_VENDOR_CLASS_VAL_EXT_BASE	500
Class Value Extension Base	COM	N/A	N/A
Specific Driver Value Extension Base	С	IVISPECAN_INSTR_SPECIFIC_VAL_EXT_BASE	1000
	COM	N/A	N/A

# 18 IviSpecAn Function Parameter Value Definitions

This section specifies the actual values for each function parameter that defines values. Each value can be extended using the extension bases defined in Section 17.1, *IviSpecAn Attribute Value Extension Bases*, unless otherwise noted.

### **Acquisition Status**

Parameter: Status

Value Name	Language	Identifier	Actual Value
Acquisition	С	IVISPECAN_VAL_ACQUISITION_STATUS_COMPLETE	1
Complete	.NET	AcquisitionStatus.Complete	1
	COM	IviSpecAnAcquisitionStatusComplete	1
Acquisition In	С	IVISPECAN_VAL_ACQUISITION_STATUS_IN_PROGRESS	0
Progress	.NET	AcquisitionStatus.InProgress	0
	COM	IviSpecAnAcquisitionStatusInProgress	0
Acquisition	С	IVISPECAN_VAL_ACQUISITION_STATUS_UNKNOWN	-1
Status Unknown	.NET	AcquisitionStatus.Unknown	2
Clikilowii	COM	IviSpecAnAcquisitionStatusUnknown	-1

### **Marker Search**

Parameter: SearchType

Value Name	Language	Identifier	Actual Value
Highest	С	IVISPECAN_VAL_MARKER_SEARCH_HIGHEST	1
	.NET	MarkerSearch.Highest	0
	COM	IviSpecAnMarkerSearchHighest	1
Minimum	С	IVISPECAN_VAL_MARKER_SEARCH_MINIMUM	2
	.NET	MarkerSearch.Minimum	1
	COM	IviSpecAnMarkerSearchMinimum	2
Next Peak	С	IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK	3
	.NET	MarkerSearch.NextPeak	2
	COM	IviSpecAnMarkerSearchNextPeak	3
Next Peak Left	С	IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK_LEF T	4
	.NET	MarkerSearch.NextPeakLeft	3
	COM	IviSpecAnMarkerSearchNextPeakLeft	4
Next Peak Right	С	IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK_RIG HT	5

.NET	MarkerSearch.NextPeakRight	4
COM	IviSpecAnMarkerSearchNextPeakRight	5

## **Read Y Trace**

Parameter: maxTime

Value Name	Language	Identifier	Actual Value
Max Time Immediate	C	IVISPECAN_VAL_MAX_TIME_IMMEDIATE	0x0
	COM	IviSpecAnMaxTimeImmediate	0x0
Max Time Infinite	С	IVISPECAN_VAL_MAX_TIME_INFINITE	0xFFFFFFFFUL
	COM	IviSpecAnMaxTimeInfinite	0xFFFFFFFFUL

The definitions for this parameter may not be extended using the extension bases defined in Section 16.2, *IviSpecAn Attribute Value Extension Bases*. No extension mechanism is defined for this parameter.

#### **Set Instrument From Marker**

Parameter: InstrumentSetting

Value Name	Language	Identifier	Actual Value
Frequency Center	С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENC Y_CENTER	1
	.NET	InstrumentSetting.FrequencyCenter	0
	COM	IviSpecAnInstrumentSettingFrequencyCenter	1
Frequency Span	С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENC Y_SPAN	2
	.NET	InstrumentSetting.FrequencySpan	1
	COM	IviSpecAnInstrumentSettingFrequencySpan	2
Frequency Start	С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENC Y_START	3
	.NET	InstrumentSetting.FrequencyStart	2
	COM	IviSpecAnInstrumentSettingFrequencyStart	3
Frequency Stop	С	IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENC Y_STOP	4
	.NET	InstrumentSetting.FrequencyStop	3
	COM	IviSpecAnInstrumentSettingFrequencyStop	4
Reference Level	С	IVISPECAN_VAL_INSTRUMENT_SETTING_REFERENC E_LEVEL	5
	.NET	InstrumentSetting.ReferenceLevel	4
	COM	IviSpecAnInstrumentSettingReferenceLevel	5

# 19 IviSpecAn Error, Completion Code, and Exception Class Definitions

The table below specifies the actual value for each status code that the IviSpecAn class specification defines.

Table 19-1. IviSpecAn Completion Codes

Error Name	Description			
		Language	Identifier	Value(hex)
Measure	Uncalibrated measurement			
Uncalibrated		С	IVISPECAN_WARN_MEASURE_UNCALIBRATED	0x3FFA2001
		.NET		CD7C4BB8- 66CE-4dbe- B680- 02179E8A45C8
		COM	S_IVISPECAN_MEASURE_UNCALIBRATED	0x00042001
Over Range	M	easurement	overrange	
		С	IVISPECAN_WARN_OVER_RANGE	0x3FFA2002
		.NET		A8EC3638- D902-43f6- 84B4- 38597547C41F
		COM	S_IVISPECAN_OVER_RANGE	0x00042002
Marker Not	Marker Not Enabled			
Enabled		C	IVISPECAN_ERROR_MARKER_NOT_ENABLED	0xBFFA2001
		.NET	MarkerNotEnabledException	N/A
		COM	E_IVISPECAN_MARKER_NOT_ENABLED	0x80042001
Not Delta	No	ot Delta Mai	rker	
Marker		C	IVISPECAN_ERROR_NOT_DELTA_MARKER	0xBFFA2002
		.NET	NotDeltaMarkerException	N/A
		COM	E_IVISPECAN_NOT_DELTA_MARKER	0x80042002
Trigger Not	Tł	ne trigger so	urce is not set to software trigger.	
Software		.NET	<pre>Ivi.Driver.TriggerNotSoftwareException</pre>	IVI defined exception (see IVI 3.2)
		C	IVIDMM_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001
		COM	E_IVIDMM_TRIGGER_NOT_SOFTWARE	0x80041001
Max Time	M	ax Time Ex	ceeded	
Exceeded		C	IVISPECAN_ERROR_MAX_TIME_EXCEEDED	0xBFFA2003

Table 19-1. IviSpecAn Completion Codes

	.NET	Ivi.Driver.MaxTimeExceededException	IVI Defined Exception, see IVI 3.2
	COM	E_IVISPECAN_MAX_TIME_EXCEEDED	0x80042003

Table 18-2 defines the format of the message string associated with the errors. In C, these strings are returned by the Error Message function. In COM, these strings are the description contained in the ErrorInfo object. In .NET these strings are the *Message* property of the exception class thrown by the method or property.

**Note:** In the description string table entries listed below, **%s** is always used to represent the component name

Table 19-2. IviSpecAn Error Message Strings

Name	Message String
Measure Uncalibrated	"%s: The instrument was in an uncalibrated state when the measurement was taken."
Over Range	"%s:The measurement taken was over the instrument's range."
Marker Not Enabled	"%s: The Active Marker is not enabled."
Not Delta Marker	"%s: The Active Marker is not a delta marker."
Max Time Exceeded	"%s: The maximum waiting time for this operation was exceeded."

# 19.1 IVI.NET IviSpecAn Exceptions and Warnings

This section defines the list of IVI.NET exceptions and warnings that are specific to the IviSpecAn 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.SpecAn namespace.

- MarkerNotEnabledException
- NotDeltaMarkerException

# 19.1.1 MarkerNotEnabledException

# Description

This exception is used when the driver tries to perform an operation on the active marker and it is not enabled.

#### **Constructors**

## **Message String**

The active marker is not enabled.

# Usage

If driver developers use constructors that take a message string, they are responsible for message string localization.

# 19.1.2 NotDeltaMarkerException

# Description

This exception is used when the driver tries to perform a delta marker operation on the active marker, but the active marker is not a delta marker.

#### **Constructors**

# **Message String**

```
The active marker is not a delta marker. Marker name: <markerName>
```

#### Usage

If driver developers use constructors that take a message string, they are responsible for message string localization.

# 20 IviSpecAn Hierarchies

# 20.1 IviSpecAn .NET Hierarchy

Table 20-1. IviSpecAn .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Туре
Frequency		
ConfigureStartStop	Configure Frequency Start Stop	M
ConfigureCenterSpan	Configure Frequency Center Span	M
Start	Frequency Start	P
Stop	Frequency Stop	P
Offset	Frequency Offset	P
SweepCoupling		
Configure	Configure Sweep Coupling	M
ResolutionBandwidthAuto	Resolution Bandwith Auto	P
ResolutionBandwidth	Resolution Bandwidth	P
VideoBandwidthAuto	Video Bandwidth Auto	P
VideoBandwidth	Video Bandwidth	P
SweepTimeAuto	Sweep Time Auto	P
SweepTime	Sweep Time	P
Level		
Configure	Configure Level	M
AmplitudeUnits	Amplitude Units	P
InputImpedance	Input Impedance	P
Reference	Reference Level	P
ReferenceOffset	Reference Level Offset	P
AttenuationAuto	Attenuation Auto	P
Attenuation	Attenuation	P
Acquisition		
Configure	Configure Acquisition	M
SweepModeContinuous	Sweep Mode Continuous	P
NumberOfSweeps	Number Of Sweeps	P
DetectorTypeAuto	Detector Type Auto	P
DetectorType	Detector Type	P
VerticalScale	Vertical Scale	P
Display		
UnitsPerDivision	Units Per Division	P
NumberOfDivisions	Number Of Divisions	P

Table 20-1. IviSpecAn .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
ExternalMixer		
Configure	Configure External Mixer	M
Harmonic	External Mixer Harmonic	P
AverageConversionLoss	External Mixer Average Conversion Loss	P
Enabled	External Mixer Enabled	P
NumberOfPorts	External Mixer Number Of Ports	P
Bias		
Configure	Configure External Mixer Bias	M
Level	Bias	P
Limit	Bias Limit	P
Enabled	Bias Enabled	P
ConversionLossTable		P
Configure	Configure Conversion Loss Table	M
Enabled	Conversion Loss Table Enabled	P
Marker		
ConfigureEnabled	Configure Marker Enabled	M
ConfigureSearch	Configure Marker Search	M
Count	Marker Count	P
DisableAll	Disable All Markers	M
ActiveMarker	Active Marker	P
Amplitude	Marker Amplitude	P
Enabled	Marker Enabled	P
MakeDelta	Make Marker Delta	M
Туре	Marker Type	P
Search	Marker Search	M
PeakExcursion	Peak Excursion	P
Position	Marker Position	P
Query	Query Marker	M
QueryReference	QueryReferenceMarker	M
ReferenceAmplitude	Reference Marker Amplitude	P
ReferencePosition	Reference Marker Position	P
SetInstrumentFromMarker	Set Instrument From Marker	M
SignalTrackEnabled	Signal Track Enabled	P
Threshold	Marker Threshold	P
Trace	Marker Trace	P
Name	Marker Name	P

Table 20-1. IviSpecAn .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Туре
FrequencyCounter		
Configure	Configure Marker Frequency Counter	M
Enabled	Marker Frequency Counter Enabled	P
Resolution	Marker Frequency Counter Resolution	P
Preselector		
Peak	Peak Preselector	M
Traces		
Abort	Abort	M
AcquisitionStatus	Acquisition Status	M
Initiate	Initiate	M
Count	Trace Count	P
CreateSpectrum	Create Spectrum	M
Item	Trace Item	P
Name	Trace Name	P
SendSoftwareTrigger	Send Software Trigger	M
Traces[]		
FetchY	Fetch Y Trace	M
ReadY	Read Y Trace	M
Size	Trace Size	P
Туре	Trace Type	P
Math		
Add	Add Traces	M
Сору	Copy Trace	M
Exchange	Exchange Traces	M
Subtract	Subtract Traces	M
Trigger		
Source	Trigger Source	P
External		
Configure	Configure External Trigger	M
Level	External Trigger Level	P
Slope	External Trigger Slope	P
Video		
Configure	Configure Video Trigger	M
Level	Vider Trigger Level	P
Slope	Video Trigger Slope	P

# 20.1.1 IviSpecAn .NET Interfaces

In addition to implementing IVI Inherent capabilities interfaces, IviSpecAn-interfaces contain interface reference propertues for accessing the following IviSpecAn interfaces:

- IIviSpecAnAcquisition
- IiviSpecAnDisplay
- IIviSpecAnExternalMixer
- IiviSpecAnFrequency
- IiviSpecAnLevel
- IiviSpecAnMarker
- IiviSpecAnPreselector
- IiviSpecAnSweepCoupling
- IiviSpecAnTraces
- IiviSpecAnTrigger

The IiviSpecAnExternalMixer interface contains interface reference properties for accessing the following interfaces:

- IIviSpecAnExternalMixerBias
- IIviSpecAnExternalMixerConversionLossTable

The IiviSpecAnMarker interface contains interface reference properties for accessing the following interfacs:

IiviSpecAnMarkerFrequencyCounter

The IiviSpecAnTrigger interface contains interface reference properties for accessing the following interfaces:

- IiviSpecAnTriggerExternal
- IiviSpecAnTriggerVideo

The IiviSpecAnTraces is a collection, that contains elements of type IiviSpecAnTrace. Each of the IiviSpecAnTrace interfaces, also contains an interface reference property for accessing the following interface:

• IIviSpecAnTracesMath

# 20.1.2 .NET Interface Reference Properties

Interface reference properties are used to navigate the IviSpecAn COM hierarchy. This section describes the interface reference properties that the IviSpecAn defines.

Data Type	COM Property Name
IIviSpecAnAcquisition	RO
IIviSpecAnDisplay	Display
IIviSpecAnExternalMixer	ExternalMixer
IIviSpecAnExternalMixerBias	ExternalMixer.Bias
IIviSpecAnExternalMixerConversionL ossTable	ExternalMixer.ConversionLossTable
IIviSpecAnFrequency	Frequency
IIviSpecAnLevel	Level
IIviSpecAnMarker	Marker
IIviSpecAnMarkerFrequencyCounter	Marker.FrequencyCounter
IIviSpecAnPreselector	Preselector
IIviSpecAnSweepCoupling	SweepCoupling
IIviSpecAnTrace	Traces[]
IIviSpecAnTraces	Traces
IIviSpecAnTracesMath	Traces.Math
IIviSpecAnTrigger	Trigger
IIviSpecAnTriggerExternal	Trigger.External
IIviSpecAnTriggerVideo	Trigger.Video

# 20.2 IviSpecAn COM Hierarchy

Table 20-2. IviSpecAn COM Hierarchy

COM Interface Hierarchy	Generic Name	Туре
Frequency		
ConfigureStartStop	Configure Frequency Start Stop	M
ConfigureCenterSpan	Configure Frequency Center Span	M
Start	Frequency Start	P
Stop	Frequency Stop	P
Offset	Frequency Offset	P

Table 20-2. IviSpecAn COM Hierarchy

COM Interface Hierarchy	Generic Name	Туре
SweepCoupling		
Configure	Configure Sweep Coupling	M
ResolutionBandwidthAuto	Resolution Bandwith Auto	P
ResolutionBandwidth	Resolution Bandwidth	P
VideoBandwidthAuto	Video Bandwidth Auto	P
VideoBandwidth	Video Bandwidth	P
SweepTimeAuto	Sweep Time Auto	P
SweepTime	Sweep Time	P
Level		
Configure	Configure Level	M
AmplitudeUnits	Amplitude Units	P
InputImpedance	Input Impedance	P
Reference	Reference Level	P
ReferenceOffset	Reference Level Offset	P
AttenuationAuto	Attenuation Auto	P
Attenuation	Attenuation	P
Acquisition		
Configure	Configure Acquisition	M
SweepModeContinuous	Sweep Mode Continuous	P
NumberOfSweeps	Number Of Sweeps	P
DetectorTypeAuto	Detector Type Auto	P
DetectorType	Detector Type	P
VerticalScale	Vertical Scale	P
Display		
UnitsPerDivision	Units Per Division	P
NumberOfDivisions	Number Of Divisions	P
ExternalMixer		
Configure	Configure External Mixer	M
Harmonic	External Mixer Harmonic	P
AverageConversionLoss	External Mixer Average Conversion Loss	P
Enabled	External Mixer Enabled	P
NumberOfPorts	External Mixer Number Of Ports	P
Bias		
Configure	Configure External Mixer Bias	M
Level	Bias	P
Limit	Bias Limit	P
Enabled	Bias Enabled	P

Table 20-2. IviSpecAn COM Hierarchy

<b>COM Interface Hierarchy</b>	Generic Name	Туре
ConversionLossTable		P
Configure	Configure Conversion Loss Table	M
Enabled	Conversion Loss Table Enabled	P
Marker		
ConfigureEnabled	Configure Marker Enabled	M
ConfigureSearch	Configure Marker Search	M
Count	Marker Count	P
DisableAll	Disable All Markers	M
ActiveMarker	Active Marker	P
Amplitude	Marker Amplitude	P
Enabled	Marker Enabled	P
MakeDelta	Make Marker Delta	M
Туре	Marker Type	P
Search	Marker Search	M
PeakExcursion	Peak Excursion	P
Position	Marker Position	P
Query	Query Marker	M
QueryReference	QueryReferenceMarker	M
ReferenceAmplitude	Reference Marker Amplitude	P
ReferencePosition	Reference Marker Position	P
SetInstrumentFromMarker	Set Instrument From Marker	M
SignalTrackEnabled	Signal Track Enabled	P
Threshold	Marker Threshold	P
Trace	Marker Trace	P
Name	Marker Name	P
FrequencyCounter		
Configure	Configure Marker Frequency Counter	M
Enabled	Marker Frequency Counter Enabled	P
Resolution	Marker Frequency Counter Resolution	P
Preselector		
Peak	Peak Preselector	M
Traces		
Abort	Abort	M
AcquisitionStatus	Acquisition Status	M
Initiate	Initiate	M
Count	Trace Count	P
Item	Trace Item	P

Table 20-2. IviSpecAn COM Hierarchy

COM Interface Hierarchy	Generic Name	Туре
Name	Trace Name	P
SendSoftwareTrigger	Send Software Trigger	M
Item		
FetchY	Fetch Y Trace	M
ReadY	Read Y Trace	M
Size	Trace Size	P
Туре	Trace Type	P
Math		
Add	Add Traces	M
Сору	Copy Trace	M
Exchange	Exchange Traces	M
Subtract	Subtract Traces	M
Trigger		
Source	Trigger Source	P
External		
Configure	Configure External Trigger	M
Level	External Trigger Level	P
Slope	External Trigger Slope	P
Video		
Configure	Configure Video Trigger	M
Level	Vider Trigger Level	P
Slope	Video Trigger Slope	P

# 20.2.1 IviSpecAn COM Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviSpecAn interfaces contain interface reference properties for accessing the following IviSpecAn interfaces:

```
IIviSpecAnAcquisition
IIviSpecAnDisplay
IIviSpecAnExternalMixer
IIviSpecAnFrequency
IIviSpecAnLevel
IIviSpecAnMarker
IIviSpecAnPreselector
IIviSpecAnSweepCoupling
IIviSpecAnTrace
IIviSpecAnTraces
IIviSpecAnTrigger
```

The IIviSpecAnExternalMixer interface contains interface reference properties for accessing the following additional IviSpecAn external mixer interfaces:

```
IIviSpecAnExternalMixerBias
IIviSpecAnExternalMixerConversionLossTable
```

The IIviSpecAnMarker interface contains interface reference properties for accessing the following additional IviSpecAn marker interface:

```
IIviSpecAnMarkerFrequencyCounter
```

The IIviSpecAnTrigger interface contains interface reference properties for accessing the following additional IviSpecAn trigger interfaces:

```
IIviSpecAnTriggerExternal
IIviSpecAnTriggerVideo
```

The IIviSpecAnTraces interface contains methods and properties for accessing a collection of objects that implement the IIviScopeTrace interface. The IIviSpecAnTraces interface also contains interface reference properties for accessing the following additional IviSpecAn trace interface:

```
IIviSpecAnTracesMath
```

The following table lists the interfaces that this specification defines and their GUIDs.

Table 20-3. lviSpecAn Interface GUIDs

Interface	GUID
IIviSpecAn	{ 47ed52d8-a398-11d4-ba58-000064657374 }
IIviSpecAnAcquisition	{ 47ed52c4-a398-11d4-ba58-000064657374 }
IIviSpecAnDisplay	{ 47ed52ca-a398-11d4-ba58-000064657374 }
IIviSpecAnExternalMixer	{ 47ed52cb-a398-11d4-ba58-000064657374 }
IIviSpecAnExternalMixerBias	{ 47ed52d3-a398-11d4-ba58-000064657374 }
IIviSpecAnExternalMixerConversio nLossTable	{ 47ed52d4-a398-11d4-ba58-000064657374 }
IIviSpecAnFrequency	{ 47ed52c2-a398-11d4-ba58-000064657374 }
IIviSpecAnLevel	{ 47ed52c6-a398-11d4-ba58-000064657374 }
IIviSpecAnMarker	{ 47ed52cc-a398-11d4-ba58-000064657374 }
IIviSpecAnMarkerFrequencyCounter	{ 47ed52d0-a398-11d4-ba58-000064657374 }
IIviSpecAnPreselector	{ 47ed52cd-a398-11d4-ba58-000064657374 }
IIviSpecAnSweepCoupling	{ 47ed52c3-a398-11d4-ba58-000064657374 }
IIviSpecAnTrace	{ 47ed52c8-a398-11d4-ba58-000064657374 }
IIviSpecAnTraces	{ 47ed52c7-a398-11d4-ba58-000064657374 }
IIviSpecAnTracesMath	{ 47ed52c9-a398-11d4-ba58-000064657374 }
IIviSpecAnTrigger	{ 47ed52c5-a398-11d4-ba58-000064657374 }
IIviSpecAnTriggerExternal	{ 47ed52ce-a398-11d4-ba58-000064657374 }
IIviSpecAnTriggerVideo	{ 47ed52cf-a398-11d4-ba58-000064657374 }

# 20.2.2 IviSpecAn COM Interface Reference Properties

Interface reference properties are used to navigate the IviSpecAn COM hierarchy. This section describes the interface reference properties that the IviSpecAn defines.

Data Type	COM Property Name
IIviSpecAnAcquisition	RO
IiviSpecAnDisplay	Display
IIviSpecAnExternalMixer	ExternalMixer
IIviSpecAnExternalMixerBias	ExternalMixer.Bias
IIviSpecAnExternalMixerConversionL ossTable	ExternalMixer.ConversionLossTabl e
IiviSpecAnFrequency	Frequency
IiviSpecAnLevel	Level
IiviSpecAnMarker	Marker
IiviSpecAnMarkerFrequencyCounter	Marker.FrequencyCounter
IiviSpecAnPreselector	Preselector
IiviSpecAnSweepCoupling	Sweep Coupling
IiviSpecAnTraces	Traces
IIviSpecAnTracesMath	Traces.Math
IiviSpecAnTrigger	Trigger
IiviSpecAnTriggerExternal	Trigger.External
IiviSpecAnTriggerVideo	Trigger.Video

# 20.2.3 IviSpecAn COM Category

The IviSpecAn class COM Category shall be "IviSpecAn", and the Category ID (CATID) shall be  $\{47ed5126-a398-11d4-ba58-000064657374\}$ .

# 20.3 IviSpecAn C Function Hierarchy

The IviSpecAn class function hierarchy is shown in the following table.

Table 19-3. IviSpecAn C Function Heirarchy	
Name or Class	Function Name
Configuration	
Configure Frequency Start Stop	<pre>IviSpecAn_ConfigureFrequencyStartStop</pre>
Configure Frequency Center Span	IviSpecAn_ConfigureFrequencyCenterSpan
Configure Frequency Offset	IviSpecAn_ConfigureFrequnecyOffset
Configure Sweep Coupling	IviSpecAn_ConfigureSweepCoupling
Configure Acquisition	IviSpecAn_ConfigureAcquisition
Configure Level	IviSpecAn_ConfigureLevel
Configure Trace Type	<pre>IviSpecAn_ConfigureTraceType</pre>
Get Trace Name	IviSpecAn_GetTraceName
Query Trace Size	IviSpecAn_QueryTraceSize
Multitrace	
Add Traces	IviSpecAn_AddTraces
Exchange Traces	IviSpecAn_ExchangeTraces
Copy Trace	IviSpecAn_CopyTrace
Subtract Traces	IviSpecAn_SubtractTraces
Marker	
Configure Marker Enabled	IviSpecAn_ConfigureMarkerEnabled
Configure Marker Frequency Counter	IviSpecAn_ConfigureMarkerFrequencyCounter
Configure Marker Search	IviSpecAn_ConfigureMarkerSearch
Configure Signal Track Enabled	IviSpecAn_ConfigureSignalTrackEnabled
Disable All Markers	IviSpecAn_DisableAllMarkers
Marker Search	IviSpecAn_MarkerSearch
Move Marker	IviSpecAn_MoveMarker
Query Marker	IviSpecAn_QueryMarker
Set Active Marker	IviSpecAn_SetActiveMarker
Set Instrument From Marker	IviSpecAn_SetInstrumentFromMarker
Query Marker Type	<pre>IviSpecAn_QueryMarkerType</pre>
Make Marker Delta	IviSpecAn_MakeMarkerDelta
Query Reference Marker	IviSpecAn_QueryReferenceMarker
Get Marker Name	IviSpecAn_GetMarkerName
Trigger	
Configure Trigger Source	IviSpecAn_ConfigureTriggerSource
Configure External Trigger	IviSpecAn_ConfigureExternalTrigger
Configure Video Trigger	IviSpecAn_ConfigureVideoTrigger

Table 19-3. IviSpecAn C Function Heirarchy		
Name or Class	Function Name	
External Mixer		
Configure External Mixer Enabled	IviSpecAn_ConfigureExternalMixerEnabled	
Configure External Mixer Number Of Ports	<pre>IviSpecAn_ConfigureExternalMixerNumberOfPo rts</pre>	
Configure External Mixer	IviSpecAn_ConfigureExternalMixer	
Configure External Mixer Bias Enabled	<pre>IviSpecAn_ConfigureExternalMixerBiasEnable d</pre>	
Configure External Mixer Bias	IviSpecAn_ConfigureExternalMixerBias	
Configure Conversion Loss Table Enabled	<pre>IviSpecAn_ConfigureConversionLossTableEnab led</pre>	
Configure Conversion Loss Table	IviSpecAn_ConfigureConversionLossTable	
Preselector		
Peak Preselector	IviSpecAn_PeakPreselector	
Measurement		
Read Y Trace	IviSpecAn_ReadYTrace	
Fetch Y Trace	IviSpecAn_FetchYTrace	
Low Level Measurement		
Initiate	IviSpecAn_Initiate	
Abort	IviSpecAn_Abort	
Acquisition Status	IviSpecAn_AcquisitionStatus	
Send Software Trigger	<pre>IviSpecAn_SendSoftwareTrigger</pre>	

# **Specific Driver Development Guidelines**

#### Introduction

This section describes situations driver developers should be aware of when developing a specific instrument driver that complies with the IviSpecAn class.

#### Time Domain Measurements

In certain situations the spectrum analyzer may be used to take time domain rather than frequency domain measurements. Typical time domain measurements are oscilloscope traces, but when a spectrum analyzer is taking a time domain measurement, the amplitude of only one frequency is measured. To set the spectrum analyzer to a time domain mode, the start and stop frequency have to be set to the same frequency; the frequency of interest. Sweeping with the start and stop set to the same frequency causes the analyzer to take many independent measurements of that frequency over time (in the time domain). This is very useful in determining such things as the change in output power of a transmitter as it warms up. Another use is to capture a peak of a pulsed signal at a particular frequency.

# Noise Level, and Test Speed Optimizations

Note the following paragraphs offer suggestions on ways to optimize the Spectrum Analyzer to take measurements. Following these suggestions does not necessarily have the desired results. All suggestions are merely that; suggestions. Spectrum analyzers allow the user to set many different attributes that affect different aspects of a measurement. Therefore, it is imperative to use the attributes effectively to ensure that the measurements that are taken are the most valid possible.

To have the lowest noise level, typically the analyzer is set up with a very low resolution bandwidth, a video bandwidth of about 10 times the resolution bandwidth, the least attenuation that is practical, a very slow sweep time, and the narrowest start frequency to stop frequency span that is practical.

The fastest speed is accomplished by using almost the opposite setup as for low noise level. Higher resolution bandwidth, a video bandwidth still of about 10 times the resolution bandwidth, a fairly narrow start frequency to stop frequency span, and of course, a fast sweep speed.

# **Disabling Unused Extensions**

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  $Prefix_init$ ,  $Prefix_init$  in the options, or  $Prefix_init$  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 shall 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 IviSpecAnMultitrace Extension Capability Group

The IviSpecAnMultitrace extension group affects the instrument behavior only when the end user calls any of its functions. Therefore, this specification does not recommend attribute values that disable the IviSpecAnMultitrace extension group.

#### Disabling the IviSpecAnMarker Extension Group

The IviSpecAnMarker extension group affects the instrument behavior only when any of its markers are enabled. Therefore the recommended technique to disable the group is to set the Marker Enabled attribute to False for each marker. This functionality is also encapsulated in the function Disable All Markers.

Table 0-1. Values for Disabling the IviSpecAnMarker Extension Group

Attribute	Value
Marker Enabled	False

#### Disabling the IviSpecAnTrigger Extension Group

Attribute values that effectively disable the IviSpecAnTrigger extension group are shown in the following table.

Table 0-2. Values for Disabling the IviSpecAnTrigger Extension Group

Attribute	Value
Trigger Source	Immediate

## Disabling the IviSpecAnExternalTrigger Extension Group

The IviSpecAnExternalTrigger extension group affects instrument behavior only when the Trigger Source attribute is set to External. Therefore, this specification does not recommend attribute values that disable the IviSpecAnExternalTrigger extension group.

## Disabling the IviSpecAnVideoTrigger Extension Group

The IviSpecAnVideoTrigger extension group affects instrument behavior only when the Trigger Source attribute is set to Video. Therefore, this specification does not recommend attribute values that disable the IviSpecAnVideoTrigger extension group.

#### Disabling the IviSpecAnDisplay Extension Group

The IviSpecAnDisplay extension group affects instrument behavior only when its attributes are used. Therefore, this specification does not recommend attribute values that disable the IviSpecAnDisplay extension group.

## Disabling the IviSpecAnMarkerType Extension Group

The IviSpecAnMarkerType extension group does not affect instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviSpecAnMarkerType extension group.

## Disabling the IviSpecAnDeltaMarker Extension Group

The IviSpecAnDeltaMarker extension group affects instrument behavior only when the Marker Type attribute is set to Delta for some marker and Marker Enabled is True for that same marker. Therefore, this specification does not recommend attribute values that disable the IviSpecAnDeltaMarker extension group.

#### Disabling the IviSpecAnExternalMixer Extension Group

Attribute values that effectively disable the IviSpecAnExternalMixer extension group are shown in the following table.

Table 0-3. Values for Disabling the IviSpecAnExternalMixer Extension Group

Attribute	Value
External Mixer Enabled	False

#### Disabling the IviSpecAnPreselector Extension Group

The IviSpecAnPreselector extension group affects the instrument behavior only when the end user calls the Peak Preselector function. Therefore, this specification does not recommend attribute values that disable the IviSpecAnPreselector extension group.

# **Query Instrument Status**

Based on the value of <code>QueryInstrumentStatus</code>, the instrument may be queried by the specific driver to determine if it has encountered an error. In specific driver functions, the status check should not occur in the lowest-level signal generation functions <code>Initiate</code>, <code>Abort</code>, and <code>Fetch Y Trace</code>. These functions are intended to give the application developer low-level control over signal generation. When calling these functions, the application developer is responsible for checking the status of the instrument. Checking status in every function at this level does add unnecessary overhead to the specific instrument driver.

# Interchangeability Checking Guidelines

#### Introduction

IVI drivers have a feature called interchangeability checking. Interchangeability checking returns a warning when it encounters a situation were the application program might not produce the same behavior when the user attempts to use a different instrument.

# 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.

- 1. Initiate
- 2. Read Y Trace

# Interchangeability Checking Rules

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the IviSpecAnBase capability group. In addition, interchangeability checking is performed on each extension capability group that the application program uses. An extension capability group is considered to be used by the application program after any of the following occur:

The application program calls a function that belongs to the extension capability group.

The application program accesses an attribute that belongs to the extension capability group.

The application program sets an attribute in another capability group to a value that requires the presence of the extension capability 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.

#### IviSpecAnBase Capability Group

- 1. If the Attenuation Auto attribute is set to True, then the Attenuation attribute need not be in a user specified state.
- 2. If the Detector Type Auto attribute is set to True, then the Detector Type attribute need not be in a user specified state.

- 3. If the Resolution Bandwidth Auto attribute is set to True, then the Resolution Bandwidth attribute need not be in a user specified state.
- 4. If the Sweep Time Auto attribute is set to True, then the Sweep Time attribute need not be in a user specified state.
- 5. If the Trace Type attribute is set to Clear Write, then the Number Of Sweeps attribute need not be in a user specified state.
- 6. If the Video Bandwidth Auto attribute is set to True, then the Video Bandwidth attribute need not be in a user specified state.

## IviSpecAnMultitrace Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnMultitrace capability group.

#### IviSpecAnMarker Extension Group

- 1. If the Marker Enabled attribute is set to False for a particular marker, then the Marker Position, Marker Trace, Marker Frequency Counter Enabled, Marker Frequency Counter Resolution, and Signal Track Enabled attributes need not be in a user specified state for that marker.
- 2. If the Marker Enabled attribute is set to False for all markers, then the Marker Threshold and Peak Excursion attributes need not be in a user specified state.
- 3. If the Marker Frequency Counter Enabled attribute is set to False for a particular marker, then the Marker Frequency Counter Resolution attribute need not be in a user specified state for that marker.

## IviSpecAnTrigger Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnTrigger capability group.

#### IviSpecAnExternalTrigger Extension Group

The attributes of this extension group must be in a user specified state if and only if the Trigger Source attribute is set to External.

#### IviSpecAnVideoTrigger Extension Group

The attributes of this extension group must be in a user specified state if and only if the Trigger Source attribute is set to Video.

#### IviSpecAnDisplay Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnDisplay capability group.

#### IviSpecAnMarkerType Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnMarkerType capability group.

## IviSpecAnDeltaMarker Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnDeltaMarker capability group.

## IviSpecAnExternalMixing Extension Group

- 1. If the External Mixer Enabled attribute is set to False, then all remaining group attributes need not be in a user specified state.
- 2. If the External Mixer Bias Enabled attribute is set to False, then the External Mixer Bias and External Mixer Bias Limit attributes need not be in a user specified state.

# IviSpecAnPreselector Extension Group

No additional interchangeability rules or exceptions are defined for the IviSpecAnPreselector capability group.