# IVI-4.12: IviCounter Class Specification

September 6, 2017 Edition
Revision 2.0

© Copyright 2009-2010 IVI Foundation.
All Rights Reserved.

# Important Information

The IVI-4.12: IviCounter 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.

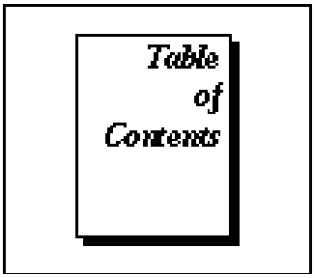
## Warranty

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 Overview of the IviCounter Specification ..... 9
1.1 Introduction ..... 9
1.2 IviCounter Class Overview ..... 9
1.3 References ..... 10
1.4 Definitions of Terms and Acronyms ..... 10

2 IviCounter Class Capabilities ..... 11
2.1 Introduction ..... 11
2.2 IviCounter Group Names ..... 11
2.3 Repeated Capability Names ..... 12
2.3.1 Channel ..... 12
2.4 Boolean Attribute and Parameter Values ..... 12
2.5 .NET Namespace ..... 12
2.6 .NET IviCounter Session Factory ..... 12

3 General Requirements ..... 15
3.1 Minimum Class Compliance ..... 15
3.1.1 Disable ..... 15
3.2 Capability Group Compliance ..... 15

4 IviCounterBase Capability Group ..... 16
4.1 IviCounterBase Capability Group Overview ..... 16
4.2 IviCounterBase Attributes ..... 16
4.2.1 Measurement Function ..... 19
4.2.2 Channel Count ..... 23
4.2.3 Channel Name (IVI-COM and IVI.NET Only) ..... 24
4.2.4 Channel Item (IVI-COM and .NET Only) ..... 25
4.2.5 Channel Impedance ..... 26
4.2.6 Channel Coupling ..... 27
4.2.7 Channel Attenuation ..... 29
4.2.8 Channel Level ..... 30
4.2.9 Channel Hysteresis ..... 31
4.2.10 Channel Slope ..... 32
4.2.11 Channel Filter Enabled ..... 34
4.2.12 Frequency Channel ..... 35
4.2.13 Frequency Estimate ..... 36
4.2.14 Frequency Resolution ..... 37
4.2.15 Frequency Aperture Time ..... 38
4.2.16 Frequency Estimate Auto ..... 39

4.2.17 Frequency Resolution Auto ..... 40
4.2.18 Period Channel ..... 41
4.2.19 Period Estimate ..... 42
4.2.20 Period Resolution ..... 43
4.2.21 Period Aperture Time ..... 44
4.2.22 Pulse Width Channel ..... 45
4.2.23 Pulse Width Estimate ..... 46
4.2.24 Pulse Width Resolution ..... 47
4.2.25 Duty Cycle Channel ..... 48
4.2.26 Duty Cycle Frequency Estimate ..... 49
4.2.27 Duty Cycle Resolution ..... 50
4.2.28 Edge Time Channel ..... 51
4.2.29 Edge Time Reference Type ..... 52
4.2.30 Edge Time Estimate ..... 54
4.2.31 Edge Time Resolution ..... 55
4.2.32 Edge Time High Reference ..... 56
4.2.33 Edge Time Low Reference ..... 57
4.2.34 Frequency Ratio Numerator Channel ..... 58
4.2.35 Frequency Ratio Denominator Channel ..... 59
4.2.36 Frequency Ratio Numerator Frequency Estimate ..... 60
4.2.37 Frequency Ratio Estimate ..... 61
4.2.38 Frequency Ratio Resolution ..... 62
4.2.39 Time Interval Start Channel ..... 63
4.2.40 Time Interval Stop Channel ..... 64
4.2.41 Time Interval Estimate ..... 65
4.2.42 Time Interval Resolution ..... 66
4.2.43 Phase Input Channel ..... 67
4.2.44 Phase Reference Channel ..... 68
4.2.45 Phase Frequency Estimate ..... 69
4.2.46 Phase Resolution ..... 70
4.2.47 Continuous Totalize Channel ..... 71
4.2.48 Gated Totalize Channel ..... 72
4.2.49 Gated Totalize Gate Source ..... 73
4.2.50 Gated Totalize Gate Slope ..... 74
4.2.51 Timed Totalize Channel ..... 76
4.2.52 Timed Totalize Gate Time ..... 77
4.2.53 Start Arm Type ..... 78
4.2.54 External Start Arm Source ..... 80
4.2.55 External Start Arm Level ..... 81
4.2.56 External Start Arm Slope ..... 82
4.2.57 External Start Arm Delay ..... 84
4.2.58 Stop Arm Type ..... 85
4.2.59 External Stop Arm Source ..... 87
4.2.60 External Stop Arm Level ..... 88
4.2.61 External Stop Arm Slope ..... 89
4.2.62 External Stop Arm Delay ..... 91
4.3 IviCounterBase Functions ..... 92
4.3.1 Abort ..... 94
4.3.2 Is Measurement Complete ..... 95
4.3.3 Get Channel Name (IVI-C Only) ..... 97
4.3.4 Configure Channel ..... 98
4.3.5 Configure Level ..... 100
4.3.6 Configure Slope (IVI-C Only) ..... 102
4.3.7 Configure Filter Enabled (IVI-C Only) ..... 103

4.3.8 Configure Frequency ..... 104
4.3.9 Configure Frequency With Aperture Time ..... 106
4.3.10 Configure Period ..... 107
4.3.11 Configure Period With Aperture Time ..... 108
4.3.12 Configure Pulse Width ..... 109
4.3.13 Configure Duty Cycle ..... 110
4.3.14 Configure Edge Time ..... 111
4.3.15 Configure Edge Time Reference Levels ..... 113
4.3.16 Configure Frequency Ratio ..... 115
4.3.17 Configure Time Interval ..... 117
4.3.18 Configure Phase ..... 119
4.3.19 Configure Continuous Totalize ..... 121
4.3.20 Start Continuous Totalize ..... 122
4.3.21 Stop Continuous Totalize ..... 123
4.3.22 Fetch Continuous Totalize Count ..... 124
4.3.23 Configure Gated Totalize ..... 125
4.3.24 Configure Timed Totalize ..... 126
4.3.25 Configure Start Arm (IVI-C Only) ..... 127
4.3.26 Configure External Start Arm ..... 128
4.3.27 Configure Stop Arm (IVI-C Only) ..... 130
4.3.28 Configure External Stop Arm ..... 131
4.3.29 Fetch ..... 133
4.3.30 Initiate ..... 135
4.3.31 Read ..... 136
4.4 IviCounterBase Behavior Model ..... 139

5 IviCounterFilter Extension Group ..... 141
5.1 IviCounterFilter Extension Group Overview ..... 141
5.2 IviCounterFilter Attributes ..... 141
5.2.1 Minimum Frequency ..... 142
5.2.2 Maximum Frequency ..... 143
5.3 IviCounterFilter Function ..... 144
5.3.1 Configure Filter ..... 145
5.4 IviCounterFilter Behavior Model ..... 146
5.5 IviCounterFilter Compliance Notes ..... 146

6 IviCounterTimeIntervalStopHoldoff Extension Group ..... 147
6.1 IviCounterTimeIntervalStopHoldoff Extension Group Overview ..... 147
6.2 IviCounterTimeIntervalStopHoldoff Attribute ..... 147
6.2.1 Time Interval Stop Holdoff ..... 148
6.3 IviCounterTimeIntervalStopHoldoff Function ..... 149
6.3.1 Configure Time Interval Stop Holdoff (IVI-C Only) ..... 150
6.4 IviCounterTimeIntervalStopHoldoff Behavior Model ..... 151
6.5 IviCounterTimeIntervalStopHoldoff Compliance Notes ..... 151

7 IviCounterVoltageMeasurement Extension Group ..... 152
7.1 IviCounterVoltageMeasurement Extension Group Overview ..... 152
7.2 IviCounterVoltageMeasurement Attributes ..... 152
7.2.1 Voltage Channel ..... 153
7.2.2 Voltage Estimate ..... 154
7.2.3 Voltage Resolution ..... 155

7.3 IviCounterVoltageMeasurement Function ..... 156
7.3.1 Configure Voltage Measurement ..... 157
7.4 IviCounterVoltageMeasurement Behavior Model ..... 159
7.5 IviCounterVoltageMeasurement Compliance Notes ..... 159

8 IviCounterEdgeTimeReferenceLevels Extension Group ..... 160
8.1 IviCounterEdgeTimeReferenceLevels Extension Group Overview ..... 160
8.2 IviCounterEdgeTimeReferenceLevels Behavior Model ..... 160
8.3 IviCounterEdgeTimeReferenceLevels Compliance Notes ..... 160

9 IviCounter Attribute ID Definitions ..... 161

10 IviCounter Attribute Value Definitions ..... 164

11 IviCounter Function Parameter Value Definitions ..... 168

12 IviCounter Error, Completion Code, and Exception Class Definitions 169

13 IviCounter Hierarchies ..... 171
13.1 IviCounter COM Hierarchy ..... 171
13.1.1 IviCounter COM Interfaces ..... 175
13.1.2 IviCounter COM Interface Reference Properties ..... 177
13.1.3 IviCounter COM Category ..... 178
13.2 IviCounter C Function Hierarchy ..... 179
13.3 IviCounter C Attribute Hierarchy ..... 181
13.4 IviCounter .NET Hierarchy ..... 184
13.4.1 IviCounter .NET Interface Reference Properties ..... 187

Appendix A Specific Driver Development Guidelines ..... 188
A.1 Introduction ..... 188
A.2 Disabling Unused Extensions ..... 188
A.3 Query Instrument Status ..... 189

Appendix B Interchangeability Checking Guidelines ..... 190
B.1 Introduction ..... 190
B.2 When to Perform Interchangeability Checking ..... 190
B.3 Interchangeability Checking of Measurement Function Attributes ..... 190
B.4 Interchangeability Checking Rules ..... 192

# IviCounter Class Specification

## IviCounter Revision History

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

**Table 1-1.** IviCounter Class Specification Revisions

| Revision Number | Date of Revision | Revision Notes |
|---|---|---|
| Revision 1.0 | 5/7/2009 | |
| Revision 2.0 | June 9, 2010 | Incorporated IVI.NET |
| Revision 2.0 | August 25, 2011 | Editorial IVI.NET change.<br>Change references to process-wide locking to AppDomain-wide locking.<br>Add an overload to the Create factory method that takes locking related parameters. |
| Revision 2.0 | June 21, 2013 | Editorial IVI.NET change.<br>Rename the IsMeasurementComplete() method to GetMeasurementComplete(). |
| Revision 2.0 | November 21, 2013 | Editorial IVI.NET change.<br>Updated the Filter.Configure() method to ConfigureFilter(). |
| Revision 2.0 | September 24, 2015 | Editorial Change – Clarified the use of one-based index for COM, and zero-based index for .NET for repeated capabilities in section 4.2.3. |
| Revision 2.0 | October 14, 2016 | Editorial Change – Modified header text for table 12.2 to indicate that the messages do not apply to .NET exceptions. |
| Revision 2.0 | September 6, 2017 | Editorial Change – Updated the name of the attribute value identifier for the Arm Type attribute in Sections 4.2.53 and 4.2.58 to match the name in Section 10. |

API Versions

| Architecture | Drivers that comply with version 2.0 comply with all of the versions below |
|---|---|
| C | 1.0, 2.0 |

| | |
|---|---|
| COM | 1.0, 2.0 |
| .NET | 2.0 |

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

# 1 Overview of the IviCounter Specification

## 1.1 Introduction

This specification defines the IVI class for counter timers. The IviCounter class is designed to support the typical counter timer as well as common extended functionality found in more complex instruments. This section summarizes the *IviCounter 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:

- IviCounter Class Overview
- References
- Definitions of Terms and Acronyms

## 1.2 IviCounter Class Overview

This specification defines the IVI class for counters. The IviCounter class is designed to support the typical frequency time interval counter as well as common extended functionality found in instruments that are more complex. The IviCounter class conceptualizes a counter as an instrument that can measure frequency of a signal, and can often perform the following measurements:

- Measure periods
- Measure pulse widths
- Measure rise and fall times
- Measure intervals between two events on different channels
- Measure phase between two signals
- Measure duty cycle of a signal
- Measure ratio of frequencies of two signals
- Totalize events

The IviCounter class is divided into the base capability group and extension groups. The IviCounterBase capability group is used to configure a counter for single and dual channel measurements as described in Section 4, *IviCounter Base Capability Group*.

In addition to the base capabilities, the IviCounter class defines extended capabilities for counters that have:

- High, low, and band pass filtering
- Time Interval Stop Holdoff
- Voltage Measurement

The IviCounter 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 and acronyms that are specific to the IviCounter class.

To make a measurement there must be: Arming, Gating and a Trigger, all of which are events.

| | |
|---|---|
| Arming | A pre-trigger condition that must be fulfilled before a measurement is allowed to start or stop. |
| Estimate | The approximate expected measurement value. |
| Event | A signal on a measurement channel passing through a specific voltage level with a specific slope. |
| External Arming | The start and stop arm may be configured to allow start and stop on rising or falling edges or both from an external source. Delays after start arm and/or stop arm may be programmed. |
| External Gating | Gating from an external source. |
| Gating | Time during which measurement is made. |
| Immediate Arming | Always armed. The programmed aperture time defines the start arm (synchronized with the trigger threshold) and the measurement time. |
| Internal Gating | The programmed aperture time defines the gate which is synchronized with the trigger threshold. |
| Resolution | The smallest distinguishable value required from a measurement. It is the quantization size, i.e. the smallest delta value that can be detected. |
| Trigger | The threshold at which measurement commences. |
| Trigger Level | The specific voltage level at which a Trigger occurs. |
| Trigger Slope | The direction of the voltage change that produces a Trigger. The voltage change for Positive slope is from a lower voltage level to a higher voltage level. The voltage change for Negative slope is from a higher voltage level to a lower voltage level. |

# 2 IviCounter Class Capabilities

## 2.1 Introduction

The IviCounter specification divides generic counter timer 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 IviCounter Group Names

The capability group names for the IviCounter 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.** IviCounter Group Names

| Group Name | Description |
|---|---|
| `IviCounterBase` | Base Capability Group: Configures, initiates, and returns results of a frequency measurement on any counter compliant with this class. This group defines a number of functions that are relevant in many of the other groups. |
| `IviCounterFilter` | Extension Group: IviCounter with the capability to specify the filter minimum and maximum frequencies of the input signal. |
| `IviCounterTimeIntervalStopHoldoff` | Extension Group: IviCounter with the capability to holdoff the stop trigger for time interval measurements. |
| `IviCounterVoltageMeasurement` | Extension Group: IviCounter with the capability to make voltage measurements. |
| `IviCounterEdgeTimeReferenceLevels` | Extension Group: IviCounter with the capability to make Percentage based Edge Time measurements. |

## 2.3 Repeated Capability Names

The IviCounter Class Specification defines one repeated capability. Refer to the sections of *IVI-3.1: Driver Architecture Specification* that deal with repeated capabilities. The relevant sections 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*. The following are repeated capabilities for the IviCounter Class Specification.

- Channel

### 2.3.1 Channel

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

## 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 IviCounter class is Ivi.Counter.

## 2.6 .NET IviCounter Session Factory

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

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

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

Refer to Section 4.3.11, *Multithread Safety*, of *IVI-3.1: Driver Architecture Specification* for a complete description of IVI.NET driver locking. Refer to Section 8, Table 8.2 *Required Lock Type Behavior for Drivers With the Same Access Key*, of *IVI-3.2, Inherent Capability Specification*, for an explanation of how the values for `lockType` and `accessKey` are used to determine the kind of multithreaded lock to use for the driver instance.

### .NET Method Prototype

```
IIviCounter Ivi.Counter.Create(String name);

IIviCounter Ivi.Counter.Create(String name,
                               Boolean idQuery,
                               Boolean reset);

IIviCounter Ivi.Counter.Create(String name,
                               Boolean idQuery,
                               Boolean reset,
                               String options);

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

### Parameters

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

| Outputs | Description | Base Type |
|---|---|---|
| Return Value | Interface reference to the IIviCounter interface of the driver referenced by session. | IIviCounter |

### Defined Values

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

### .NET Exceptions

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

### Usage

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

```
IIviCounter counter = IviCounter.Create("MyLogicalName");
```

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

# 3 General Requirements

This section describes the general requirements a specific driver 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 IviCounter 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 IviCounterBase capability group.

### 3.1.1 Disable

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

## 4.1 IviCounterBase Capability Group Overview

To measure frequency, the counter counts number of crossings of the measured signal through a given level at a given slope. Built-in hysteresis is used to reject signal noise. The counter takes the measurement for a specific amount of time called the gating time. Some counters provide autotrigger, which enables them to measure frequency without specifying level and direction of the crossing. Refer to the Section 4.4, *IviCounterBase Behavior Model* for a detailed description of the counter behavior and the usage of estimate and resolution attributes.

The most general triggering scheme for frequency or period measurement is given below.

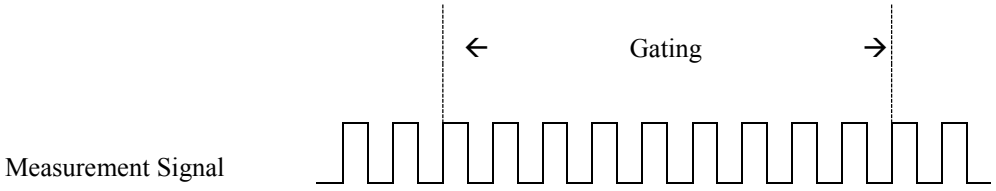



**Figure 4-1** Typical Counter Timer Diagram

## 4.2 IviCounterBase Attributes

The IviCounterBase capability group defines the following attributes:

- Measurement Function
- Channel Count
- Channel Name (IVI-COM and IVI.NET Only)
- Channel Item (IVI-COM and IVI.NET Only)
- Channel Impedance
- Channel Coupling
- Channel Attenuation
- Channel Level
- Channel Hysteresis
- Channel Slope
- Channel Filter Enabled

- Frequency Channel
- Frequency Estimate
- Frequency Resolution
- Frequency Aperture Time
- Frequency Estimate Auto
- Frequency Resolution Auto
- Period Channel
- Period Estimate
- Period Resolution
- Period Aperture Time
- Pulse Width Channel
- Pulse Width Estimate
- Pulse Width Resolution
- Duty Cycle Channel
- Duty Cycle Frequency Estimate
- Duty Cycle Resolution
- Edge Time Channel
- Edge Time Estimate
- Edge Time Resolution
- Edge Time High Reference
- Edge Time Low Reference
- Frequency Ratio Numerator Channel
- Frequency Ratio Denominator Channel
- Frequency Ratio Numerator Frequency Estimate
- Frequency Ratio Estimate
- Frequency Ratio Resolution
- Time Interval Start Channel
- Time Interval Stop Channel
- Time Interval Estimate

- Time Interval Resolution
- Phase Input Channel
- Phase Reference Channel
- Phase Frequency Estimate
- Phase Resolution
- Continuous Totalize Channel
- Gated Totalize Channel
- Gated Totalize Gate Source
- Gated Totalize Gate Slope
- Timed Totalize Channel
- Timed Totalize Gate Time
- Start Arm Type
- External Start Arm Source
- External Start Arm Level
- External Start Arm Slope
- External Start Arm Delay
- Stop Arm Type
- External Stop Arm Source
- External Stop Arm Level
- External Stop Arm Slope
- External Stop Arm Delay

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 9, *IviCounter Attribute ID Definitions*.

### 4.2.1 Measurement Function

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | N/A | None | Configure Frequency<br>Configure Frequency Manual<br>Configure Frequency With Aperture Time<br>Configure Period<br>Configure Period With Aperture Time<br>Configure Pulse Width<br>Configure Duty Cycle<br>Configure Edge Time<br>Configure Edge Time Reference Levels<br>Configure Frequency Ratio<br>Configure Time Interval<br>Configure Phase<br>Configure Continuous Totalize<br>Configure Gated Totalize<br>Configure Timed Totalize<br>Configure Voltage Measurement |

.NET Property Name

MeasurementFunction

.NET Enumeration Name

MeasurementFunction

COM Property Name

Function

COM Enumeration Name

IviCounterMeasurementFunctionEnum

C Constant Name

IVICOUNTER_ATTR_MEASUREMENT_FUNCTION

Description

Specifies the current measurement function of the Counter. The user sets the function by calling one of the configure measurement functions or the set attribute function. See configure measurement functions for

details on setting up a measurement. See the behavior model for proper usage of the Measurement Function 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.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Frequency | Sets the Counter to measure Frequency. | | |
| | | C | IVICOUNTER_VAL_FREQUENCY |
| | | COM | IviCounterFunctionFrequency |
| | | .NET | MeasurementFunction.Frequency |
| Frequency with Aperture | Sets the Counter to measure Frequency with Aperture. | | |
| | | C | IVICOUNTER_VAL_FREQUENCY_WITH_APERTURE |
| | | COM | IviCounterFunctionFrequencyWithAperture |
| | | .NET | MeasurementFunction.FrequencyWithAperture |
| Period | Sets the Counter to measure Period. | | |
| | | C | IVICOUNTER_VAL_PERIOD |
| | | COM | IviCounterFunctionPeriod |
| | | .NET | MeasurementFunction.Period |
| Period with Aperture | Sets the Counter to measure Period with Aperture. | | |
| | | C | IVICOUNTER_VAL_PERIOD_WITH_APERTURE |
| | | COM | IviCounterFunctionPeriodWithAperture |
| | | .NET | MeasurementFunction.PeriodWithAperture |
| Pulse Width | Sets the Counter to measure Pulse Width. | | |
| | | C | IVICOUNTER_VAL_PULSE_WIDTH |
| | | COM | IviCounterFunctionPulseWidth |
| | | .NET | MeasurementFunction.PulseWidth |
| Duty Cycle | Sets the Counter to measure Duty Cycle. | | |
| | | C | IVICOUNTER_VAL_DUTY_CYCLE |
| | | COM | IviCounterFunctionDutyCycle |
| | | .NET | MeasurementFunction.DutyCylce |
| Edge Time | Sets the Counter to measure Edge Time. | | |

| | | | |
|---|---|---|---|
| | | C | IVICOUNTER_VAL_EDGE_TIME |
| | | COM | IviCounterFunctionEdgeTime |
| | | .NET | MeasurementFunction.EdgeTime |
| Frequency Ratio | Sets the Counter to measure Frequency Ratio. | | |
| | | C | IVICOUNTER_VAL_FREQUENCY_RATIO |
| | | COM | IviCounterFunctionFrequencyRatio |
| | | .NET | MeasurementFunction.FrequencyRatio |
| Time Interval | Sets the Counter to measure Time Interval. | | |
| | | C | IVICOUNTER_VAL_TIME_INTERVAL |
| | | COM | IviCounterFunctionTimeInterval |
| | | .NET | MeasurementFunction.TimeInterval |
| Phase | Sets the Counter to measure Phase. | | |
| | | C | IVICOUNTER_VAL_PHASE |
| | | COM | IviCounterFunctionPhase |
| | | .NET | MeasurementFunction.Phase |
| Continuous Totalize | Sets the Counter to measure Continuous Totalize. | | |
| | | C | IVICOUNTER_VAL_CONTINUOUS_TOTALIZE |
| | | COM | IviCounterFunctionContinuousTotalize |
| | | .NET | MeasurementFunction.ContinuousTotalize |
| Gated Totalize | Sets the Counter to measure Gated Totalize. | | |
| | | C | IVICOUNTER_VAL_GATED_TOTALIZE |
| | | COM | IviCounterFunctionGatedTotalize |
| | | .NET | MeasurementFunction.GatedTotalize |
| Timed Totalize | Sets the Counter to measure Timed Totalize. | | |
| | | C | IVICOUNTER_VAL_TIMED_TOTALIZE |
| | | COM | IviCounterFunctionTimedTotalize |
| | | .NET | MeasurementFunction.TimedTotalize |
| DC Voltage | Sets the Counter to measure DC Voltage. | | |
| | | C | IVICOUNTER_VAL_DC_VOLTAGE |
| | | COM | IviCounterFunctionDCVoltage |
| | | .NET | MeasurementFunction.DCVoltage |
| Maximum Voltage | Sets the Counter to measure Maximum Voltage. | | |
| | | C | IVICOUNTER_VAL_MAXIMUM_VOLTAGE |
| | | COM | IviCounterFunctionMaximumVoltage |

| | | | |
|---|---|---|---|
| | | .NET | MeasurementFunction.MaximumVoltage |
| Minimum Voltage | Sets the Counter to measure Minimum Voltage. | | |
| | | C | IVICOUNTER_VAL_MINIMUM_VOLTAGE |
| | | COM | IviCounterFunctionMinimumVoltage |
| | | .NET | MeasurementFunction.MinimumVoltage |
| RMS Voltage | Sets the Counter to measure RMS Voltage. | | |
| | | C | IVICOUNTER_VAL_RMS_VOLTAGE |
| | | COM | IviCounterFunctionRMSVoltage |
| | | .NET | MeasurementFunction.RMSVoltage |
| Peak-to-Peak Voltage | Sets the Counter to measure Peak-to-Peak Voltage. | | |
| | | C | IVICOUNTER_VAL_PEAK_TO_PEAK_VOLTAGE |
| | | COM | IviCounterFunctionPeakToPeakVoltage |
| | | .NET | MeasurementFunction.PeakToPeakVoltage |

### Compliance Notes

1. If an IviCounter specific driver implements any of the defined values in the following table, it shall also implement the corresponding capability group:

| Value | Required Capability Group |
|---|---|
| DC Voltage | IviCounterVoltageMeasurement |
| Maximum Voltage | IviCounterVoltageMeasurement |
| Minimum Voltage | IviCounterVoltageMeasurement |
| RMS Voltage | IviCounterVoltageMeasurement |
| Peak-to-Peak Voltage | IviCounterVoltageMeasurement |

2. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_MEASUREMENT_FUNCTION_CLASS_EXT_BASE and less than IVICOUNTER_VAL_MEASUREMENT_FUNCTION_SPECIFIC_EXT_BASE.
3. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_MEASUREMENT_FUNCTION_SPECIFIC_EXT_BASE.
4. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_MEASUREMENT_FUNCTION_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of IVICOUNTER_VAL_MEASUREMENT_FUNCTION_SPECIFIC_EXT_BASE and IVICOUNTER_VAL_MEASUREMENT_FUNCTION_CLASS_EXT_BASE.

### 4.2.2 Channel Count

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

#### .NET Property Name

Channels.Count

This property is inherited from the base interface IIviRepeatedCapabilityIdentification.

#### COM Property Name

Channels.Count

#### C Constant Name

IVICOUNTER_ATTR_CHANNEL_COUNT

#### Description

Returns the number of available channels.

#### .NET Exceptions

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

### 4.2.3 Channel Name (IVI-COM and IVI.NET Only)

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | RO | Channels | None | None |

#### .NET Property Name

Channels[].Name

This property is inherited from IIviRepeatedCapabilityIdentification.

#### COM Property Name

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

#### C Constant Name

N/A

(Use the GetChannelName function.)

#### Description

Returns the physical repeated capability identifier defined by the specific driver for the channel that corresponds to the index that the user specifies.

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

Valid values for the Index parameter are between one and the value of the Channel Count attribute. 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.

### 4.2.4 Channel Item (IVI-COM and .NET Only)

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| IIviCounterChannel* | RO | Channels | None | None |

#### .NET Property Name

```
Channels[String name]
```

This indexer is inherited from `IIviRepeatedCapabilityCollection`. The string parameter uniquely identifies a particular channel in the `ChannelsCollection`.

#### COM Property Name

```
HRESULT Channels.Item ([in] BSTR Name,
                       [out,retval] IIviCounterChannel **pVal);
```

#### C Constant Name

```
N/A
```

#### Description

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

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

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

#### .NET Exceptions

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

### 4.2.5 Channel Impedance

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | Up | Configure Channel |

.NET Property Name

```
Channels[].Impedance
```

COM Property Name

```
Channels.Item().Impedance
```

C Constant Name

```
IVICOUNTER_ATTR_IMPEDANCE
```

Description

Specifies the input impedance of the channel in Ohms.

Common values are 50, 75, and 1,000,000.

#### .NET Exceptions

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

### 4.2.6 Channel Coupling

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | Channels | None | Configure Channel |

.NET Property Name

```
Channels[].Coupling
```

.NET Enumeration Name

```
Coupling
```

COM Property Name

```
Channels.Item().Coupling
```

COM Enumeration Name

```
IviCounterCouplingEnum
```

C Constant Name

```
IVICOUNTER_ATTR_COUPLING
```

Description

Specifies the electrical coupling method used on the input channel.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| AC | The counter AC couples the channel signal. | | |
| | | C | IVICOUNTER_VAL_AC |
| | | COM | IviCounterCouplingAC |
| | | .NET | Coupling.AC |
| DC | The counter DC couples the channel signal. | | |
| | | C | IVICOUNTER_VAL_DC |
| | | COM | IviCounterCouplingDC |
| | | .NET | Coupling.DC |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_COUPLING_CLASS_EXT_BASE and less than IVICOUNTER_VAL_COUPLING_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_COUPLING_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_COUPLING_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_COUPLING_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_COUPLING_CLASS_EXT_BASE`.

### 4.2.7 Channel Attenuation

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | None | Configure Channel |

.NET Property Name

```
Channels[].Attenuation
```

COM Property Name

```
Channels.Item().Attenuation
```

C Constant Name

```
IVICOUNTER_ATTR_ATTENUATION
```

Description

Specifies the scale factor by which the channel attenuates the input. Increasing this value decreases the sensitivity. For instance, setting this value to 10 attenuates the input by a factor of 10.

#### .NET Exceptions

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

### 4.2.8 Channel Level

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | None | Configure Level |

.NET Property Name

```
Channels[].Level
```

COM Property Name

```
Channels.Item().Level
```

C Constant Name

```
IVICOUNTER_ATTR_CHANNEL_LEVEL
```

Description

Specifies the voltage level the input signal must pass through to produce a count. Level is specified as the voltage at the input terminals and is independent of 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.

### 4.2.9 Channel Hysteresis

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | None | Configure Level |

.NET Property Name

```
Channels[].Hysteresis
```

COM Property Name

```
Channels.Item().Hysteresis
```

C Constant Name

```
IVICOUNTER_ATTR_CHANNEL_HYSTERESIS
```

Description

Specifies the Hysteresis value in volts. Hysteresis sets how far a signal must fall below the level before a rising edge can again be detected, and how far a signal must rise above the level before a falling edge can again be detected. Its function is to eliminate false events caused by signal noise. Hysteresis is specified as the voltage at the input terminals and is independent of 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.

### 4.2.10 Channel Slope

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | Channels | None | Configure Channel Slope |

.NET Property Name

```
Channels[].Slope
```

.NET Enumeration Name

```
Slope
```

COM Property Name

```
Channels.Item().Slope
```

COM Enumeration Name

```
IviCounterSlopeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_CHANNEL_SLOPE
```

Description

Specifies whether a rising (positive) or a falling (negative) edge triggers the counter.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Positive | A positive (rising) edge passing through the trigger level triggers the counter. | | |
| | | C | IVICOUNTER_VAL_POSITIVE |
| | | COM | IviCounterlSlopePositive |
| | | .NET | Slope.Positive |
| Negative | A negative (falling) edge passing through the trigger level triggers the counter. | | |
| | | C | IVICOUNTER_VAL_NEGATIVE |
| | | COM | IviCounterSlopeNegative |
| | | .NET | Slope.Negative |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE and less than IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE`.

### 4.2.11 Channel Filter Enabled

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViBoolean | R/W | Channels | None | Configure Filter Enabled |

.NET Property Name

```
Channels[].FilterEnabled
```

COM Property Name

```
Channels.Item().FilterEnabled
```

C Constant Name

```
IVICOUNTER_ATTR_FILTER_ENABLED
```

Description

Specifies if the filter on the selected channel is enabled.

**.NET Exceptions**

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

### 4.2.12 Frequency Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Frequency |

.NET Property Name

Frequency.Channel

COM Property Name

Frequency.Channel

C Constant Name

IVICOUNTER_ATTR_FREQUENCY_CHANNEL

Description

Specifies the input channel the frequency is measured on.

**.NET Exceptions**

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

### 4.2.13 Frequency Estimate

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

.NET Property Name

```
Frequency.Estimate
```

COM Property Name

```
Frequency.Estimate
```

C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_ESTIMATE
```

Description

Specifies the estimated frequency, in hertz, for the frequency function. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement.

Setting this attribute overrides the Frequency Aperture Time and sets the Frequency Estimate Auto to false.

**.NET Exceptions**

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

### 4.2.14 Frequency Resolution

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

.NET Property Name

```
Frequency.Resolution
```

COM Property Name

```
Frequency.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_RESOLUTION
```

Description

Specifies the resolution of the measurement, in hertz, for the frequency function.

Setting this attribute overrides the Frequency Aperture Time and sets the Frequency Resolution Auto to false.

#### .NET Exceptions

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

### 4.2.15 Frequency Aperture Time

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Frequency With Aperture Time |

.NET Property Name

```
Frequency.ApertureTime
```

COM Property Name

```
Frequency.ApertureTime
```

C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_APERTURE_TIME
```

Description

Specifies the aperture time for the frequency with aperture time function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

Setting this attribute overrides the Frequency Estimate and Frequency Resolution.

This attribute can be read to determine the value of aperture time selected by the driver based on the Frequency Estimate and Frequency Resolution.

#### .NET Exceptions

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

### 4.2.16 Frequency Estimate Auto

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

.NET Property Name

```
Frequency. EstimateAuto
```

COM Property Name

```
Frequency.EstimateAuto
```

C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_ESTIMATE_AUTO
```

Description

Specifies if the Counter Frequency Estimate Auto is enabled. Use the Frequency Estimate Auto attribute to enable auto frequency selection. If this attribute is set to True, the instrument automatically determines the best frequency estimate for the measurement. If this attribute is set to False, the user specifies the frequency estimate of the measurement by explicitly setting the Frequency Estimate attribute.

#### .NET Exceptions

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

## 4.2.17 Frequency Resolution Auto

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

.NET Property Name

```
Frequency.ResolutionAuto
```

COM Property Name

```
Frequency.ResolutionAuto
```

C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_RESOLUTION_AUTO
```

Description

Specifies if the Counter Frequency Resolution Auto is enabled. Use the Frequency Resolution Auto attribute to enable auto resolution selection. If this attribute is set to True, the instrument automatically determines the best frequency resolution for the measurement. If this attribute is set to False, the user specifies the frequency resolution of the measurement by explicitly setting the Frequency Resolution attribute.

### .NET Exceptions

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

### 4.2.18 Period Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Period |

.NET Property Name

Period.Channel

COM Property Name

Period.Channel

C Constant Name

IVICOUNTER_ATTR_PERIOD_CHANNEL

Description

Specifies the input channel the period is measured on.

**.NET Exceptions**

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

### 4.2.19 Period Estimate

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Period |

.NET Property Name

Period.Estimate

COM Property Name

Period.Estimate

C Constant Name

IVICOUNTER_ATTR_PERIOD_ESTIMATE

Description

Specifies the estimated period for the period function. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.20 Period Resolution

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Period |

.NET Property Name

```
Period.Resolution
```

COM Property Name

```
Period.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_PERIOD_RESOLUTION
```

Description

Specifies the resolution of the measurement for the period function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

#### .NET Exceptions

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

### 4.2.21 Period Aperture Time

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Period with Aperture Time |

.NET Property Name

```
Period.ApertureTime
```

COM Property Name

```
Period.ApertureTime
```

C Constant Name

```
IVICOUNTER_ATTR_PERIOD_APERTURE_TIME
```

Description

Specifies the aperture time for the period with aperture time function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

Setting this attribute overrides the Period Estimate and Period Resolution.

This attribute can be read to determine the value of aperture time selected by the driver based on the Period Estimate and Period Resolution.

#### .NET Exceptions

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

### 4.2.22 Pulse Width Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Pulse Width |

#### .NET Property Name

PulseWidth.Channel

#### COM Property Name

PulseWidth.Channel

#### C Constant Name

IVICOUNTER_ATTR_PULSE_WIDTH_CHANNEL

#### Description

Specifies the input channel the pulse width is measured on.

#### .NET Exceptions

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

### 4.2.23 Pulse Width Estimate

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Pulse Width |

.NET Property Name

PulseWidth.Estimate

COM Property Name

PulseWidth.Estimate

C Constant Name

IVICOUNTER_ATTR_PULSE_WIDTH_ESTIMATE

Description

Specifies the estimated pulse width for the pulse width function. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.24 Pulse Width Resolution

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Pulse Width |

.NET Property Name

```
PulseWidth.Resolution
```

COM Property Name

```
PulseWidth.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_PULSE_WIDTH_RESOLUTION
```

Description

Specifies the resolution of the measurement for the pulse width function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.25 Duty Cycle Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Duty Cycle |

.NET Property Name

```
DutyCycle.Channel
```

COM Property Name

```
DutyCycle.Channel
```

C Constant Name

```
IVICOUNTER_ATTR_DUTY_CYCLE_CHANNEL
```

Description

Specifies the input channel the duty cycle is measured on.

**.NET Exceptions**

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

## 4.2.26 Duty Cycle Frequency Estimate

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

.NET Property Name

```
DutyCycle.FrequencyEstimate
```

COM Property Name

```
DutyCycle.FrequencyEstimate
```

C Constant Name

```
IVICOUNTER_ATTR_DUTY_CYCLE_FREQUENCY_ESTIMATE
```

Description

Specifies the estimated frequency, in hertz, for the duty cycle function. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement.

### .NET Exceptions

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

## 4.2.27 Duty Cycle Resolution

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

### .NET Property Name

DutyCycle.Resolution

### COM Property Name

DutyCycle.Resolution

### C Constant Name

IVICOUNTER_ATTR_DUTY_CYCLE_RESOLUTION

### Description

Specifies the resolution for the duty cycle function. Duty Cycle Resolution is a unitless value.

### .NET Exceptions

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

### 4.2.28 Edge Time Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Edge Time |

#### .NET Property Name

```
EdgeTime.Channel
```

#### COM Property Name

```
EdgeTime.Channel
```

#### C Constant Name

```
IVICOUNTER_ATTR_EDGE_TIME_CHANNEL
```

#### Description

Specifies the input channel the edge time is measured on.

#### .NET Exceptions

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

## 4.2.29 Edge Time Reference Type

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | N/A | None | Configure Edge Time Reference Levels |

.NET Property Name

```
EdgeTime.ReferenceType
```

.NET Enumeration Name

```
ReferenceType
```

COM Property Name

```
EdgeTime.ReferenceType
```

COM Enumeration Name

```
IviCounterEdgeTimeReferenceTypeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_EDGE_TIME_REFERENCE_TYPE
```

Description

Specifies the current reference type of the Counter. That is if the Edge Time High Referenace and Edge Time Low Reference are interpretted as percentage of peak-peak or absolute volts.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Voltage | Sets the Counter to measure Edge Time based on voltage reference levels. | | |
| | | C | IVICOUNTER_VAL_VOLTAGE_REFERENCE_TYPE |
| | | COM | IviCounterReferenceTypeVoltage |
| | | .NET | ReferenceType.Voltage |
| Percent | Sets the Counter to measure Edge Time based on percentage reference levels. | | |
| | | C | IVICOUNTER_VAL_PERCENT_REFERENCE_TYPE |
| | | COM | IviCounterReferenceTypePercent |
| | | .NET | ReferenceType.Percent |

### .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 IviCounter specific driver implements any of the defined values in the following table, it shall also implement the corresponding capability group:

| Value | Required Capability Group |
|---|---|
| Percent | IviCounterEdgeTimeReferenceLevels |

2. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_REFERENCE_TYPE_CLASS_EXT_BASE and less than IVICOUNTER_VAL_REFERENCE_TYPE_SPECIFIC_EXT_BASE.
3. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_REFERENCE_TYPE_SPECIFIC_EXT_BASE.
4. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_REFERENCE_TYPE_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of IVICOUNTER_VAL_REFERENCE_TYPE_SPECIFIC_EXT_BASE and IVICOUNTER_VAL_REFERENCE_TYPE_CLASS_EXT_BASE.

## 4.2.30 Edge Time Estimate

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Edge Time |

.NET Property Name

EdgeTime.Estimate

COM Property Name

EdgeTime.Estimate

C Constant Name

IVICOUNTER_ATTR_EDGE_TIME_ESTIMATE

Description

Specifies the estimated edge time for the edge time function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.31 Edge Time Resolution

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Edge Time |

.NET Property Name

```
EdgeTime.Resolution
```

COM Property Name

```
EdgeTime.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_EDGE_TIME_RESOLUTION
```

Description

Specifies the resolution of the measurement for the edge time function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

## 4.2.32 Edge Time High Reference

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

.NET Property Name

```
EdgeTime.HighReference
```

COM Property Name

```
EdgeTime.HighReference
```

C Constant Name

```
IVICOUNTER_ATTR_EDGE_TIME_HIGH_REFERENCE
```

Description

Specifies the high reference level for the edge time function. For a Rise Time measurement, this is the level where the measurement stops and for a Fall Time measurements, this is the level where the measurement starts.

### .NET Exceptions

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

### 4.2.33 Edge Time Low Reference

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

.NET Property Name

```
EdgeTime.LowReference
```

COM Property Name

```
EdgeTime.LowReference
```

C Constant Name

```
IVICOUNTER_ATTR_EDGE_TIME_LOW_REFERENCE
```

Description

Specifies the low reference level for the edge time function. For a Rise Time measurement, this is the level where the measurement starts and for a Fall Time measurements, this is the level where the measurement stops.

#### .NET Exceptions

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

### 4.2.34 Frequency Ratio Numerator Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Frequency Ratio |

.NET Property Name

FrequencyRatio.NumeratorChannel

COM Property Name

FrequencyRatio.NumeratorChannel

C Constant Name

IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_CHANNEL

Description

Specifies the input channel the frequency ratio is measured on.

**.NET Exceptions**

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

## 4.2.35 Frequency Ratio Denominator Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Frequency Ratio |

### .NET Property Name

```
FrequencyRatio.DenominatorChannel
```

### COM Property Name

```
FrequencyRatio.DenominatorChannel
```

### C Constant Name

```
IVICOUNTER_ATTR_FREQUENCY_RATIO_DENOMINATOR_CHANNEL
```

### Description

Specifies the input denominator channel the frequency ratio is measured on.

### .NET Exceptions

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

## 4.2.36 Frequency Ratio Numerator Frequency Estimate

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

.NET Property Name

FrequencyRatio.NumeratorFrequencyEstimate

COM Property Name

FrequencyRatio.NumeratorFrequencyEstimate

C Constant Name

IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_FREQUENCY_ESTIMATE

Description

Specifies the estimated numerator frequency, in hertz, for the frequency ratio function. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement.

### .NET Exceptions

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

## 4.2.37 Frequency Ratio Estimate

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

### .NET Property Name

FrequencyRatio.Estimate

### COM Property Name

FrequencyRatio.Estimate

### C Constant Name

IVICOUNTER_ATTR_FREQUENCY_RATIO_ESTIMATE

### Description

Specifies the estimated frequency ratio for the frequency ratio function. Frequency Ratio Estimate is unitless value.

### .NET Exceptions

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

### 4.2.38 Frequency Ratio Resolution

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

#### .NET Property Name

FrequencyRatio.Resolution

#### COM Property Name

FrequencyRatio.Resolution

#### C Constant Name

IVICOUNTER_ATTR_FREQUENCY_RATIO_RESOLUTION

#### Description

Specifies the frequency ratio resolution of the frequency ratio function. Frequency Ratio Resolution is unitless value.

#### .NET Exceptions

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

### 4.2.39 Time Interval Start Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Time Interval |

.NET Property Name

```
TimeInterval.StartChannel
```

COM Property Name

```
TimeInterval.StartChannel
```

C Constant Name

```
IVICOUNTER_ATTR_TIME_INTERVAL_START_CHANNEL
```

Description

Specifies the start channel used to perform the time interval function.

**.NET Exceptions**

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

## 4.2.40 Time Interval Stop Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Time Interval |

### .NET Property Name

```
TimeInterval.StopChannel
```

### COM Property Name

```
TimeInterval.StopChannel
```

### C Constant Name

```
IVICOUNTER_ATTR_TIME_INTERVAL_STOP_CHANNEL
```

### Description

Specifies the stop channel used to perform the time interval function.

### .NET Exceptions

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

### 4.2.41 Time Interval Estimate

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Time Interval |

.NET Property Name

TimeInterval.Estimate

COM Property Name

TimeInterval.Estimate

C Constant Name

IVICOUNTER_ATTR_TIME_INTERVAL_ESTIMATE

Description

Specifies the estimated time interval for the time interval function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.42 Time Interval Resolution

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Time Interval |

.NET Property Name

```
TimeInterval.Resolution
```

COM Property Name

```
TimeInterval.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_TIME_INTERVAL_RESOLUTION
```

Description

Specifies the resolution of the measurement for the time interval function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.NET Exceptions**

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

### 4.2.43 Phase Input Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Phase |

.NET Property Name

```
Phase.InputChannel
```

COM Property Name

```
Phase.InputChannel
```

C Constant Name

```
IVICOUNTER_ATTR_PHASE_INPUT_CHANNEL
```

Description

Specifies the input channel the phase is measured on.

**.NET Exceptions**

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

### 4.2.44 Phase Reference Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Phase |

#### .NET Property Name

```
Phase.ReferenceChannel
```

#### COM Property Name

```
Phase.ReferenceChannel
```

#### C Constant Name

```
IVICOUNTER_ATTR_PHASE_REFERENCE_CHANNEL
```

#### Description

Specifies the reference channel for the phase measurement.

#### .NET Exceptions

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

## 4.2.45 Phase Frequency Estimate

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

.NET Property Name

```
Phase.FrequencyEstimate
```

COM Property Name

```
Phase.FrequencyEstimate
```

C Constant Name

```
IVICOUNTER_ATTR_PHASE_FREQUENCY_ESTIMATE
```

Description

Specifies the estimated frequency, in hertz, for the phase function reference channel. The driver uses this to optimize the configuration of the instrument for the input signal. The driver typically use this to set the duration of the measurement.

### .NET Exceptions

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

## 4.2.46 Phase Resolution

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

### .NET Property Name

```
Phase.Resolution
```

### COM Property Name

```
Phase.Resolution
```

### C Constant Name

```
IVICOUNTER_ATTR_PHASE_RESOLUTION
```

### Description

Specifies the resolution of the measurement, in degrees, for the phase function reference channel.

### .NET Exceptions

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

## 4.2.47 Continuous Totalize Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Continuous Totalize |

### .NET Property Name

TotalizeContinuous.Channel

### COM Property Name

TotalizeContinuous.Channel

### C Constant Name

IVICOUNTER_ATTR_CONTINUOUS_TOTALIZE_CHANNEL

### Description

Specifies the input channel for the continuous totalize function.

### .NET Exceptions

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

## 4.2.48 Gated Totalize Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Gated Totalize |

### .NET Property Name

TotalizeGated.Channel

### COM Property Name

TotalizeGated.Channel

### C Constant Name

IVICOUNTER_ATTR_GATED_TOTALIZE_CHANNEL

### Description

Specifies the input channel for the gated totalize function.

### .NET Exceptions

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

## 4.2.49 Gated Totalize Gate Source

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Gated Totalize |

### .NET Property Name

TotalizeGated.GateSource

### COM Property Name

TotalizeGated.GateSource

### C Constant Name

IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SOURCE

### Description

Specifies the gate source for the gated totalize function.

### Defined Values

The value can be a channel name alias, a driver-specific channel string, or one of the values from IVI-3.3. *Standard Cross-Class Capabilities Specification*.

### .NET Exceptions

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

## 4.2.50 Gated Totalize Gate Slope

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

.NET Property Name

```
TotalizeGated.GateSlope
```

.NET Enumeration Name

```
Slope
```

COM Property Name

```
TotalizeGated.GateSlope
```

COM Enumeration Name

```
IviCounterSlopeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SLOPE
```

Description

Specifies the gate slope that enables the gated totalize function.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Positive | A positive (rising) edge passing through the trigger level enables the gate. | | |
| | | C | IVICOUNTER_VAL_POSITIVE |
| | | COM | IviCounterSlopePositive |
| | | .NET | Slope.Positive |
| Negative | A negative (falling) edge passing through the trigger level enables the gate. | | |
| | | C | IVICOUNTER_VAL_NEGATIVE |
| | | COM | IviCounterSlopeNegative |
| | | .NET | Slope.Negative |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE and less than IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE`.

### 4.2.51 Timed Totalize Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Timed Totalize |

.NET Property Name

```
TotalizeTimed.Channel
```

COM Property Name

```
TotalizeTimed.Channel
```

C Constant Name

```
IVICOUNTER_ATTR_TIMED_TOTALIZE_CHANNEL
```

Description

Specifies the input channel for the timed totalize function.

**.NET Exceptions**

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

## 4.2.52 Timed Totalize Gate Time

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure Timed Totalize |

.NET Property Name

TotalizeTimed.GateTime

COM Property Name

TotalizeTimed.GateTime

C Constant Name

IVICOUNTER_ATTR_TIMED_TOTALIZE_GATE_TIME

Description

Specifies the gate time for the timed totalize function. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

### .NET Exceptions

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

## 4.2.53 Start Arm Type

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

.NET Property Name

Arm.Start.Type

.NET Enumeration Name

ArmType

COM Property Name

Arm.Start.Type

COM Enumeration Name

IviCounterArmTypeEnum

C Constant Name

IVICOUNTER_ATTR_START_ARM_TYPE

Description

Specifies the start arm type for armed measurements.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Immediate | Immediately proceed with the measurement without waiting for an arm event | | |
| | | C | IVICOUNTER_VAL_IMMEDIATE_ARM_TYPE |
| | | COM | IviCounterArmImmediate |
| | | .NET | ArmType.Immediate |
| External | Wait for the External Start Arm Source event before proceeding with the measurement | | |
| | | C | IVICOUNTER_VAL_EXTERNAL_ARM_TYPE |
| | | COM | IviCounterArmExternal |
| | | .NET | ArmType.External |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_START_ARM_CLASS_EXT_BASE and less than IVICOUNTER_VAL_START_ARM_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_START_ARM_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_START_ARM_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_START_ARM_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_START_ARM_CLASS_EXT_BASE`.

## 4.2.54 External Start Arm Source

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure External Start Arm |

.NET Property Name

```
Arm.Start.External.Source
```

COM Property Name

```
Arm.Start.External.Source
```

C Constant Name

```
IVICOUNTER_ATTR_EXTERNAL_START_ARM_SOURCE
```

Description

Specifies the start arm source for external armed measurements.

Defined Values

The value can be a channel name alias, a driver-specific channel string, or one of the values from IVI-3.3. *Standard Cross-Class Capabilities Specification.*

**.NET Exceptions**

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

## 4.2.55 External Start Arm Level

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | N/A | None | Configure External Start Arm |

.NET Property Name

Arm.Start.External.Level

COM Property Name

Arm.Start.External.Level

C Constant Name

IVICOUNTER_ATTR_EXTERNAL_START_ARM_LEVEL

Description

Specifies the voltage level in volts that starts external armed measurements.

### .NET Exceptions

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

## 4.2.56 External Start Arm Slope

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | N/A | None | Configure External Start Arm |

.NET Property Name

```
Arm.Start.External.Slope
```

.NET Enumeration Name

```
Slope
```

COM Property Name

```
Arm.Start.External.Slope
```

COM Enumeration Name

```
IviCounterSlopeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_EXTERNAL_START_ARM_SLOPE
```

Description

Specifies the signal slope that starts external armed measurements.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Positive | A positive (rising) edge passing through the trigger level triggers the gate. | | |
| | | C | IVICOUNTER_VAL_POSITIVE |
| | | COM | IviCounterSlopePositive |
| | | .NET | Slope.Positive |
| Negative | A negative (falling) edge passing through the trigger level triggers the gate. | | |
| | | C | IVICOUNTER_VAL_NEGATIVE |
| | | COM | IviCounterSlopeNegative |
| | | .NET | Slope.Negative |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE and less than IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE`.

## 4.2.57 External Start Arm Delay

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure External Start Arm |

.NET Property Name

Arm.Start.External.Delay

COM Property Name

Arm.Start.External.Delay

C Constant Name

IVICOUNTER_ATTR_EXTERNAL_START_ARM_DELAY

Description

Specifies the delay used after an external armed measurement has been armed. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.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. Positive values only.

## 4.2.58 Stop Arm Type

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

.NET Property Name

```
Arm.Stop.Type
```

.NET Enumeration Name

```
ArmType
```

COM Property Name

```
Arm.Stop.Type
```

COM Enumeration Name

```
IviCounterArmTypeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_STOP_ARM_TYPE
```

Description

Specifies the stop arm type for armed measurements.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Immediate | Immediately end the measurement without waiting for an arm event | | |
| | | C | IVICOUNTER_VAL_IMMEDIATE_ARM_TYPE |
| | | COM | IviCounterArmImmediate |
| | | .NET | ArmType.Immediate |
| External | Wait for the External Stop Arm Source event before proceeding with the External Stop Arm Delay | | |
| | | C | IVICOUNTER_VAL_EXTERNAL_ARM_TYPE |
| | | COM | IviCounterArmExternal |
| | | .NET | ArmType.External |

**.NET Exceptions**

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_STOP_ARM_CLASS_EXT_BASE and less than IVICOUNTER_VAL_STOP_ARM_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_STOP_ARM_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_STOP_ARM_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_STOP_ARM_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_STOP_ARM_CLASS_EXT_BASE`.

## 4.2.59 External Stop Arm Source

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure External Stop Arm |

### .NET Property Name

Arm.Stop.External.Source

### COM Property Name

Arm.Stop.External.Source

### C Constant Name

IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SOURCE

### Description

Specifies the stop arm source for external armed measurements.

### Defined Values

The value can be a channel name alias, a driver-specific channel string, or one of the values from IVI-3.3. *Standard Cross-Class Capabilities Specification.*

### .NET Exceptions

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

## 4.2.60 External Stop Arm Level

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | N/A | None | Configure External Stop Arm |

.NET Property Name

```
Arm.Stop.External.Level
```

COM Property Name

```
Arm.Stop.External.Level
```

C Constant Name

```
IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_LEVEL
```

Description

Specifies the voltage level in volts that stops external armed measurements. The External Stop Arm Delay, if non-zero, is applied before the measurement stops.

### .NET Exceptions

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

## 4.2.61 External Stop Arm Slope

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViInt32 | R/W | N/A | None | Configure External Stop Arm |

.NET Property Name

```
Arm.Stop.External.Slope
```

.NET Enumeration Name

```
Slope
```

COM Property Name

```
Arm.Stop.External.Slope
```

COM Enumeration Name

```
IviCounterSlopeEnum
```

C Constant Name

```
IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SLOPE
```

Description

Specifies the signal slope that stops external armed measurements. The External Stop Arm Delay, if non-zero, is applied before the measurements stops.

Defined Values

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Positive | A positive (rising) edge passing through the trigger level triggers the gate. | | |
| | | C | IVICOUNTER_VAL_POSITIVE |
| | | COM | IviCounterSlopePositive |
| | | .NET | Slope.Positive |
| Negative | A negative (falling) edge passing through the trigger level triggers the gate. | | |
| | | C | IVICOUNTER_VAL_NEGATIVE |
| | | COM | IviCounterSlopeNegative |
| | | .NET | Slope.Negative |

### .NET Exceptions

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE and less than IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.
3. If an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE.

See Section 10, *IviCounter Attribute Value Definitions*, for the definitions of `IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE` and `IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE`.

## 4.2.62 External Stop Arm Delay

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | None | Configure External Stop Arm |

.NET Property Name

```
Arm.Stop.External.Delay
```

COM Property Name

```
Arm.Stop.External.Delay
```

C Constant Name

```
IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_DELAY
```

Description

Specifies the delay after the External Arm Stop event has occurred until the measurement stops. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**.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. Positive values only.

## 4.3 IviCounterBase Functions

The IviCounterBase capability group defines the following functions:

- Abort
- Is Measurement Complete
- Configure Channel
- Configure Level
- Configure Slope (IVI-C Only)
- Configure Filter Enabled (IVI-C Only)
- Configure Frequency
- Configure Frequency Manual
- Configure Frequency With Aperture Time
- Configure Period
- Configure Period With Aperture Time
- Configure Pulse Width
- Configure Duty Cycle
- Configure Edge Time
- Configure Edge Time Reference Levels
- Configure Frequency Ratio
- Configure Time Interval
- Configure Phase
- Configure Continuous Totalize
- Start Continuous Totalize
- Stop Continuous Totalize
- Fetch Continuous Totalize Count
- Configure Gated Totalize
- Configure Timed Totalize
- Configure Start Arm (IVI-C Only)
- Configure External Start Arm
- Configure Stop Arm (IVI-C Only)

- Configure External Stop Arm
- Fetch
- Initiate
- Read

This section describes the behavior and requirements of each function.

### 4.3.1 Abort

#### Description

Aborts a previously initiated measurement.

#### .NET Method Prototype

```
void Measurement.Abort ();
```

#### COM Method Prototype

```
HRESULT Measurement.Abort ();
```

#### C Function Prototype

```
ViStatus IviCounter_Abort (ViSession Vi);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |

#### Return Values

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

#### .NET Exceptions

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

### 4.3.2 Is Measurement Complete

#### Description

Returns whether a measurement is in progress, complete, or if the status is unknown.

#### .NET Method Prototype

```
MeasurementStatus Measurement.GetMeasurementComplete ();
```

#### COM Method Prototype

```
HRESULT Measurement.IsMeasurementComplete ([out, retval]
                            IviCounterMeasurementStatusEnum* MeasurementStatus);
```

#### C Function Prototype

```
ViStatus IviCounter_IsMeasurementComplete (ViSession Vi,
                              ViInt32 *MeasurementStatus);
```

#### Parameters

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

| Outputs | Description | Base Type |
|---|---|---|
| `MeasurementStatus` | Returns the measurement status. | `ViInt32` |

#### Defined Values for Status Parameter

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Measurement Complete | The counter timer has completed the measurement. | | |
| | | C | `IVICOUNTER_VAL_MEASUREMENT_COMPLETE` |
| | | COM | `IviCounterMeasurementStatusComplete` |
| | | .NET | `MeasurementStatus.Complete` |
| Measurement In Progress | The counter timer is still acquiring data. | | |
| | | C | `IVICOUNTER_VAL_MEASUREMENT_IN_PROGRESS` |
| | | COM | `IviCounterMeasurementStatusInProgress` |
| | | .NET | `MeasurementStatus.InProgress` |
| Measurement Status Unknown | The counter timer cannot determine the status of the measurement. | | |
| | | C | `IVICOUNTER_VAL_MEASUREMENT_STATUS_UNKNOWN` |
| | | COM | `IviCounterMeasurementStatusUnknown` |

| | | | |
|---|---|---|---|
| | | .NET | `MeasurementStatus.Unknown` |

### Return Values

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

### Compliance Notes

1. If an IVI specific driver defines additional values for this parameter, the actual values shall be greater than or equal to IviCounter Specific Driver Value Extension Base. See Section 10, *IviCounter Attribute Value Definitions* for more information.

### .NET Exceptions

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

### 4.3.3 Get Channel Name (IVI-C Only)

#### Description

This function returns the physical channel identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the ChannelIndex parameter is less than one or greater than the value of the Channel Count attribute, the function returns an empty string in the ChannelName parameter and returns an error.

#### .NET Method Prototype

```
N/A

(Use the Channels[].Name property)
```

#### COM Method Prototype

```
N/A

(Use the Channels.Name property)
```

#### C Function Prototype

```
ViStatus IviCounter_GetChannelName (ViSession Vi,
                                    ViInt32 ChannelIndex,
                                    ViInt32 ChannelNameBufferSize,
                                    ViChar ChannelName[]);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| ChannelIndex | A one-based index that defines which name to return. | ViInt32 |
| ChannelName BufferSize | The number of bytes in the ViChar array that the user specifies for the ChannelName parameter. | ViInt32 |

| Outputs | Description | Base Type |
|---|---|---|
| ChannelName | The buffer into which the function returns the channel name that corresponds to the index the user specifies.<br>The caller may pass VI_NULL for this parameter if the ChannelNameBufferSize parameter is 0. | ViChar[] |

#### Return Values

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

### 4.3.4 Configure Channel

Description

Configures the Impedance, Coupling, and Attenuation attributes of the counter channel.

.NET Method Prototype

```
void Channels[].Configure (Double impedance,
                           Coupling coupling,
                           Double attenuation);
```

COM Method Prototype

```
HRESULT Channels.Item().Configure ([in] DOUBLE Impedance,
                                   [in] IviCounterCouplingEnum Coupling,
                                   [in] DOUBLE Attenuation);
```

C Function Prototype

```
ViStatus IviCounter_ConfigureChannel (ViSession Vi,
                                      ViConstString Channel,
                                      ViReal64 Impedance,
                                      ViInt32 Coupling,
                                      ViReal64 Attenuation);
```

Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel you want to configure. | ViConstString |
| Impedance | Specifies the impedance you want the Counter to use to couple the input signal for the channel. The driver sets the Channel Impedance attribute to this value. See the attribute description for more information. | ViReal64 |
| Coupling | Specifies how you want the Counter to couple the input signal for the channel. The driver sets the Channel Coupling attribute to this value. See the attribute description for more information. | ViInt32 |
| Attenuation | Specifies the attenuation you want the Counter to use for the channel. The driver sets the Channel Attenuation attribute to this value. See the attribute description for more information. | ViReal64 |

Return Values

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

.NET Exceptions

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

### 4.3.5 Configure Level

#### Description

Configures the Level and Hysteresis attributes for a channel.

#### .NET Method Prototype

```
void Channels[].ConfigureLevel (Double triggerLevel,
                                Double hysteresis);
```

#### COM Method Prototype

```
HRESULT Channels.Item().ConfigureLevel ([in] DOUBLE TriggerLevel,
                                        [in] DOUBLE Hysteresis);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureChannelLevel (ViSession Vi,
                                           ViConstString Channel,
                                           ViReal64 TriggerLevel,
                                           ViReal64 Hysteresis);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the channel you want to configure. | `ViConstString` |
| `TriggerLevel` | Specifies the Trigger Level you want the Counter to use for the channel. The driver sets the Channel Level attribute to this value. See the attribute description for more information. | `ViReal64` |
| `Hysteresis` | Specifies the Hysteresis you want the Counter to use for the channel. The driver sets the Channel Hysteresis attribute to this value. See the attribute description for more information. | `ViReal64` |

#### Return Values

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

#### Compliance Notes

1. Trigger Level is specified as voltage at the input terminals that is independent of instrument attenuation.
2. Hysteresis is specified as voltage at the input terminals that is independent of instrument 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 method.

### 4.3.6 Configure Slope (IVI-C Only)

#### Description

Configures the Slope attribute for a channel.

#### .NET Method Prototype

```
N/A
(Use the Channels[].Slope property)
```

#### COM Method Prototype

```
N/A
(use the Channels.Item().Slope property)
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureChannelSlope (ViSession Vi,
                                           ViConstString Channel,
                                           ViInt32 Slope);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel you want to configure. | ViConstString |
| Slope | Specifies the Slope you want the Counter to use for the channel. The driver sets the Channel Slope attribute to this value. See the attribute description for more information. | ViInt32 |

#### Return Values

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

### 4.3.7 Configure Filter Enabled (IVI-C Only)

#### Description

Configures the Filter Enabled attribute for a channel.

#### .NET Method Prototype

```
N/A
(Use the Channels[].FilterEnabled property)
```

#### COM Method Prototype

```
N/A
(use the Channels.Item().FilterEnabled property)
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureChannelFilterEnabled (ViSession Vi,
                                                   ViConstString Channel,
                                                   ViBoolean FilterEnabled);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the channel you want to configure. | `ViConstString` |
| `FilterEnabled` | Specifies if you want the Counter Filter Enabled for the channel. The driver sets the Channel Filter Enabled attribute to this value. See the attribute description for more information. | `ViBoolean` |

#### Return Values

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

### 4.3.8 Configure Frequency

#### Description

These functions provide both manual and auto frequency configuration. The Configure Manual function configures the Estimate and Resolution attributes for a frequency measurement for a particular channel. The Configure function configures the instrument to determine the best estimate and resolution for the selected channel.

The Configure function sets Frequency Estimate Auto and Frequency Resolution Auto true. When the Frequency Estimate Auto or Frequency Resolution Auto are true, the Aperture Time attribute can be read to determine the Aperture Time selected by the driver.

The default conditions for automatic measurements are:

- Mode: Frequency
- Trigger Level: Auto
- Trigger Slope: Positive
- Impedance: 1 MOhm
- Attenuation: 1X
- Coupling: AC
- Filter: Off

#### .NET Method Prototype

```
void Frequency.Configure (String channel);
void Frequency.ConfigureManual (String channel,
                                Double estimate,
                                Double resolution);
```

#### COM Method Prototype

```
HRESULT Frequency.Configure ([in] BSTR Channel);

HRESULT Frequency.ConfigureManual ([in] BSTR Channel,
                                   [in] DOUBLE Estimate,
                                   [in] DOUBLE Resolution);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureFrequency (ViSession Vi,
                                        ViConstString Channel);

ViStatus IviCounter_ConfigureFrequencyManual (ViSession Vi,
                                              ViConstString Channel,
                                              ViReal64 Estimate,
                                              ViReal64 Resolution);
```

Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the frequency will be measured on. The driver sets the Frequency Channel attribute to this value. | ViConstString |
| Estimate | Specifies the estimated frequency. The driver sets the Frequency Estimate attribute to this value. See the attribute description for more information. | ViReal64 |
| Resolution | Specifies the resolution of the frequency measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Frequency Resolution attribute to this value. See the attribute description for more information. | ViReal64 |

Return Values

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

.NET Exceptions

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

### 4.3.9 Configure Frequency With Aperture Time

#### Description

Configures a frequency measurement based on the specified aperture time.

#### .NET Method Prototype

```
void Frequency.ConfigureWithAperture (String channel,
                                      PrecisionTimeSpan apertureTime);
```

#### COM Method Prototype

```
HRESULT Frequency.ConfigureWithAperture ([in] BSTR Channel,
                                         [in] DOUBLE ApertureTime);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureFrequencyWithApertureTime (ViSession Vi,
                                                        ViConstString Channel,
                                                        ViReal64 ApertureTime);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the channel the frequency will be measured on. The driver sets the Frequency Channel attribute to this value. | `ViConstString` |
| `ApertureTime` | Specifies the aperture time of the frequency measurement. The driver sets the Frequency Aperture Time to this value. See the attribute description for more information. | `ViReal64 (C/COM)`<br>`PrecisionTimeSpan (.NET)` |

#### Return Values

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

#### .NET Exceptions

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

## 4.3.10 Configure Period

### Description

Configures the estimate and resolution attributes for a period measurement.

### .NET Method Prototype

```
void Period.Configure (String channel,
                       PrecisionTimeSpan estimate,
                       PrecisionTimeSpan resolution);
```

### COM Method Prototype

```
HRESULT Period.Configure ([in] BSTR Channel,
                          [in] DOUBLE Estimate,
                          [in] DOUBLE Resolution);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigurePeriod (ViSession Vi,
                                     ViConstString Channel,
                                     ViReal64 Estimate,
                                     ViReal64 Resolution);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the period will be measured on. The driver sets the Period Channel attribute to this value. | ViConstString |
| Estimate | Specifies the estimated period. The driver sets the Period Estimate attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |
| Resolution | Specifies the resolution of the period measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Period Resolution attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

### Return Values

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

### .NET Exceptions

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

## 4.3.11 Configure Period With Aperture Time

### Description

Configures a period measurement based on the specified aperture time.

### .NET Method Prototype

```
void Period.ConfigureWithAperture (String channel,
                                   PrecisionTimeSpan apertureTime);
```

### COM Method Prototype

```
HRESULT Period.ConfigureWithAperture ([in] BSTR Channel,
                                      [in] DOUBLE ApertureTime);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigurePeriodWithApertureTime (ViSession Vi,
                                                     ViConstString Channel,
                                                     ViReal64 ApertureTime);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the period will be measured on. The driver sets the Period Channel attribute to this value. | ViConstString |
| ApertureTime | Specifies the aperture time of the period measurement. The driver sets the Period Aperture Time attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

### Return Values

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

### .NET Exceptions

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

### 4.3.12 Configure Pulse Width

#### Description

Configures the estimate and resolution attributes for a pulse width measurement.

#### .NET Method Prototype

```
void PulseWidth.Configure (String channel,
                           PrecisionTimeSpan estimate,
                           PrecisionTimeSpan resolution);
```

#### COM Method Prototype

```
HRESULT PulseWidth.Configure ([in] BSTR Channel,
                              [in] DOUBLE Estimate,
                              [in] DOUBLE Resolution);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigurePulseWidth (ViSession Vi,
                                         ViConstString Channel,
                                         ViReal64 Estimate,
                                         ViReal64 Resolution);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the pulse width will be measured on. The driver sets the Pulse Width Channel attribute to this value. | ViConstString |
| Estimate | Specifies the estimated pulse width. The driver sets the Pulse Width Estimate attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |
| Resolution | Specifies the resolution of the pulse width measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Pulse Width Resolution attribute this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

#### Return Values

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

#### .NET Exceptions

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

## 4.3.13 Configure Duty Cycle

### Description

Configures the frequency estimate and resolution attributes for a duty cycle measurement.

### .NET Method Prototype

```
void DutyCycle.Configure (String channel,
                          Double frequencyEstimate,
                          Double resolution);
```

### COM Method Prototype

```
HRESULT DutyCycle.Configure ([in] BSTR Channel,
                             [in] DOUBLE FrequencyEstimate,
                             [in] DOUBLE Resolution);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureDutyCycle (ViSession Vi,
                                        ViConstString Channel,
                                        ViReal64 FrequencyEstimate,
                                        ViReal64 Resolution);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the duty cycle will be measured on. The driver sets the Duty Cycle Frequency Channel attribute to this value. | ViConstString |
| FrequencyEstimate | Specifies the estimated frequency. The driver sets the Duty Cycle Frequency Estimate attribute to this value. See the attribute description for more information. | ViReal64 |
| Resolution | Specifies the resolution of the duty cycle measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets Duty Cycle Resolution attribute to this value. See the attribute description for more information. | ViReal64 |

### Return Values

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

### .NET Exceptions

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

### 4.3.14 Configure Edge Time

#### Description

Configures an edge time measurement. The estimate and resolution attributes are set to the values specified. The edge time reference type is set to percentage, and the edge time low reference and edge time high reference are set to 10% and 90% respectively. If the channel slope is positive a rise-time measurement is performed, if the channel slope is negative, a fall-time measurement is performed.

#### .NET Method Prototype

```
void EdgeTime.Configure (String channel,
                         PrecisionTimeSpan estimate,
                         PrecisionTimeSpan resolution);
```

#### COM Method Prototype

```
HRESULT EdgeTime.Configure ([in] BSTR Channel,
                            [in] DOUBLE Estimate,
                            [in] DOUBLE Resolution);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureEdgeTime (ViSession Vi,
                                       ViConstString Channel,
                                       ViReal64 Estimate,
                                       ViReal64 Resolution);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the edge time will be measured on. The driver sets the Edge Time Channel attribute to this value. | ViConstString |
| Estimate | Specifies the estimated edge time. The driver sets the Edge Time Estimate attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |
| Resolution | Specifies the resolution of the edge time measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Edge Time Resolution attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

#### Return Values

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

.NET Exceptions

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

## 4.3.15 Configure Edge Time Reference Levels

### Description

Configures the reference type, estimate, resolution, high reference level, and low reference level attributes for an edge time measurement. If the channel slope is positive a rise-time measurement is performed, if the channel slope is negative, a fall-time measurement is performed.

### .NET Method Prototype

```
void EdgeTime.ConfigureReferenceLevels (String channel,
                                        ReferenceType referenceType,
                                        PrecisionTimeSpan estimate,
                                        PrecisionTimeSpan resolution,
                                        Double highReference,
                                        Double lowReference);
```

### COM Method Prototype

```
HRESULT EdgeTime.ConfigureReferenceLevels ([in] BSTR Channel,
                          [in] IviCounterEdgeTimeReferenceTypeEnum ReferenceType,
                                             [in] DOUBLE Estimate,
                                             [in] DOUBLE Resolution,
                                             [in] DOUBLE HighReference,
                                             [in] DOUBLE LowReference);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureEdgeTimeReferenceLevels (ViSession Vi,
                                                      ViConstString Channel,
                                                      ViInt32 ReferenceType,
                                                      ViReal64 Estimate,
                                                      ViReal64 Resolution);
                                                      ViReal64 HighReference,
                                                      ViReal64 LowReference);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel the edge time will be measured on. The driver sets the Edge Time Channel attribute to this value. | ViConstString |
| ReferenceType | Specifies the reference type. The driver sets the Reference Type attribute to this value. See the attribute description for more information. | ViInt32 |
| Estimate | Specifies the estimated edge time. The driver sets the Edge Time Estimate attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

| | | |
|---|---|---|
| Resolution | Specifies the resolution of the edge time measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Edge Time Resolution attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |
| HighReference | Specifies the high reference level for the edge time. The driver sets the Edge Time High Reference attribute to this value. See the attribute description for more information. | ViReal64 |
| LowReference | Specifies the low reference level for the edge time. The driver sets the Edge Time Low Reference attribute to this value. See the attribute description for more information. | ViReal64 |

### Return Values

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

### .NET Exceptions

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

## 4.3.16 Configure Frequency Ratio

### Description

Configures the estimated frequencies, and resolution attributes and specifies the numerator and denominator channels for a frequency ratio measurement.

### .NET Method Prototype

```
void FrequencyRatio.Configure (String numeratorChannel,
                               String denominatorChannel,
                               Double numeratorFrequencyEstimate,
                               Double estimate,
                               Double resolution);
```

### COM Method Prototype

```
HRESULT FrequencyRatio.Configure ([in] BSTR NumeratorChannel,
                                  [in] BSTR DenominatorChannel,
                                  [in] DOUBLE NumeratorFrequencyEstimate,
                                  [in] DOUBLE Estimate,
                                  [in] DOUBLE Resolution);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureFrequencyRatio
                                        (ViSession Vi,
                                         ViConstString NumeratorChannel,
                                         ViConstString DenominatorChannel,
                                         ViReal64 NumeratorFrequencyEstimate,
                                         ViReal64 Estimate,
                                         ViReal64 Resolution);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| NumeratorChannel | Specifies the numerator channel the frequency ratio will be measured on. The driver sets the Frequency Ratio Numerator Channel attribute to this value. | ViConstString |
| DenominatorChannel | Specifies the denominator channel the frequency ratio will be measured on. The driver sets the Frequency Ratio Denominator Channel attribute to this value. | ViConstString |
| NumeratorFrequencyEstimate | Specifies the estimated frequency for the numerator of the frequency ratio. The driver sets the Frequency Ratio Numerator Frequency Estimate attribute to this value. See the attribute description for more information. | ViReal64 |

| | | |
|---|---|---|
| Estimate | Specifies the estimated frequency ratio measurement. The driver sets the Frequency Ratio Estimate attribute to this value. | ViReal64 |
| Resolution | Specifies the resolution of the frequency ratio measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Frequency Ratio Resolution attribute to this value. See the attribute description for more information. | ViReal64 |

### Return Values

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

### .NET Exceptions

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

## 4.3.17 Configure Time Interval

### Description

Configures the estimate and resolution attributes and specifies the start and stop channels for a time interval measurement.

### .NET Method Prototype

```
void TimeInterval.Configure (String startChannel,
                             String stopChannel,
                             PrecisionTimeSpan estimate,
                             PrecisionTimeSpan resolution);
```

### COM Method Prototype

```
HRESULT TimeInterval.Configure ([in] BSTR StartChannel,
                                [in] BSTR StopChannel,
                                [in] DOUBLE Estimate,
                                [in] DOUBLE Resolution);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureTimeInterval (ViSession Vi,
                                           ViConstString StartChannel,
                                           ViConstString StopChannel,
                                           ViReal64 Estimate,
                                           ViReal64 Resolution);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| StartChannel | Specifies the channel that will start the time interval measurement. The driver sets the Time Interval Start Channel attribute to this value. | ViConstString |
| StopChannel | Specifies the channel that will stop the time interval measurement. The driver sets the Time Interval Stop Channel attribute to this value. | ViConstString |
| Estimate | Specifies the estimated time interval. The driver sets the Time Interval Estimate attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |
| Resolution | Specifies the resolution of the time interval measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Time Interval Resolution attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

### Return Values

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

### .NET Exceptions

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

## 4.3.18 Configure Phase

### Description

Configures the estimate and resolution attributes and specifies the input and reference channels for a phase measurement.

### .NET Method Prototype

```
void Phase.Configure (String inputChannel,
                      String referenceChannel,
                      Double frequencyEstimate,
                      Double resolution);
```

### COM Method Prototype

```
HRESULT Phase.Configure ([in] BSTR InputChannel,
                         [in] BSTR ReferenceChannel,
                         [in] DOUBLE FrequencyEstimate,
                         [in] DOUBLE Resolution);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigurePhase (ViSession Vi,
                                    ViConstString InputChannel,
                                    ViConstString ReferenceChannel,
                                    ViReal64 FrequencyEstimate,
                                    ViReal64 Resolution);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| InputChannel | Specifies the input channel the phase will be measured on. The driver sets the Phase Input Channel attribute to this value. | ViConstString |
| ReferenceChannel | Specifies the channel the phase measurement will be referenced to. The driver sets the Phase Reference Channel attribute to this value. | ViConstString |
| FrequencyEstimate | Specifies the estimated phase. The driver sets the Phase Frequency Estimate attribute to this value. See the attribute description for more information. | ViReal64 |
| Resolution | Specifies the resolution of the phase measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Phase Frequency Resolution attribute to this value. See the attribute description for more information. | ViReal64 |

### Return Values

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

### .NET Exceptions

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

## 4.3.19 Configure Continuous Totalize

### Description

Configures the counter for a continuous totalize measurement. Start continuous totalize clears the count and starts the accumulation of counts. Stop continuous totalize stops the accumulation of counts. Fetch continuous totalize can be called if the count is accumulating or stopped to retrieve the current count.

### .NET Method Prototype

```
void TotalizeContinuous.Configure (String channel);
```

### COM Method Prototype

```
HRESULT TotalizeContinuous.Configure ([in] BSTR Channel);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureContinuousTotalize (ViSession Vi,
                                                 ViConstString Channel);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the input channel. . The driver sets the Totalize Channel attribute to this value. | `ViConstString` |

### Return Values

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

### .NET Exceptions

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

## 4.3.20 Start Continuous Totalize

### Description

Clears the count and starts the counter for a continuous totalize measurement. Refer to Section 4.3.19, *Configure Continuous Totalize* for details.

### .NET Method Prototype

```
void TotalizeContinuous.Start ();
```

### COM Method Prototype

```
HRESULT TotalizeContinuous.Start ();
```

### C Function Prototype

```
ViStatus IviCounter_StartContinuousTotalize (ViSession Vi);
```

### Parameters

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

### Return Values

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

### .NET Exceptions

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

### 4.3.21 Stop Continuous Totalize

#### Description

Stops the accumulation of counts for a continuous totalize measurement. Refer to Section 4.3.19, *Configure Continuous Totalize* for details.

#### .NET Method Prototype

```
void TotalizeContinuous.Stop ();
```

#### COM Method Prototype

```
HRESULT TotalizeContinuous.Stop ();
```

#### C Function Prototype

```
ViStatus IviCounter_StopContinuousTotalize (ViSession Vi);
```

#### Parameters

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

#### Return Values

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

#### .NET Exceptions

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

## 4.3.22 Fetch Continuous Totalize Count

### Description

Retrieves the current count while the counter is continuously totalizing. Refer to Section 4.3.19, *Configure Continuous Totalize* for details.

### .NET Method Prototype

```
Int32 TotalizeContinuous.FetchCount ();
```

### COM Method Prototype

```
HRESULT TotalizeContinuous.FetchCount ([out, retval] LONG* Measurement);
```

### C Function Prototype

```
ViStatus IviCounter_FetchContinuousTotalizeCount (ViSession Vi,
                                                  ViInt32 *Measurement);
```

### Parameters

| Inputs | Description | Datatype |
|---|---|---|
| Vi | Instrument handle | ViSession |

| Outputs | Description | Datatype |
|---|---|---|
| Measurement | Returns the measured value. | ViInt32 |

### Return Values

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

### .NET Exceptions

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

## 4.3.23 Configure Gated Totalize

### Description

Specifies the channel to use for the gate source and configures the gate slope attribute for a gated totalize measurement.

### .NET Method Prototype

```
void TotalizeGated.Configure (String channel,
                              String gateSource,
                              Slope gateSlope);
```

### COM Method Prototype

```
HRESULT TotalizeGated.Configure([in] BSTR Channel,
                                [in] BSTR GateSource,
                                [in] IviCounterSlopeEnum GateSlope);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureGatedTotalize (ViSession Vi,
                                            ViConstString Channel,
                                            ViConstString GateSource,
                                            ViInt32 GateSlope);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel that will be counted. The driver sets Gated Totalize Channel attribute to this value. | ViConstString |
| GateSource | Specifies the channel that will gate the counted channel. The driver sets Gated Totalize Gate Source attribute to this value. See the attribute description for more information. | ViConstString |
| GateSlope | Specifies the gate slope you want to configure. The driver sets the Gated Totalize Gate Slope attribute to this value. See the attribute description for more information. | ViInt32 |

### Return Values

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

### .NET Exceptions

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

## 4.3.24 Configure Timed Totalize

### Description

Sets the measurement function to Timed Totalize and configures the gate time attribute.

### .NET Method Prototype

```
void TotalizeTimed.Configure (String channel,
                              PrecisionTimeSpan gateTime);
```

### COM Method Prototype

```
HRESULT TotalizeTimed.Configure ([in] BSTR Channel,
                                 [in] DOUBLE GateTime);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureTimedTotalize (ViSession Vi,
                                            ViConstString Channel,
                                            ViReal64 GateTime);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the channel that will be counted. The driver sets the Gated Totalize Channel attribute to this value. | `ViConstString` |
| `GateTime` | Specifies the gate time that will gate the counter channel. The driver sets the Gated Totalize Gate Time attribute to this value. See the attribute description for more information. | `ViReal64 (C/COM)`<br>`PrecisionTimeSpan (.NET)` |

### Return Values

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

### .NET Exceptions

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

## 4.3.25 Configure Start Arm (IVI-C Only)

### Description

Configures the Start Arm for armed measurements.

### .NET Method Prototype

```
N/A
(use the Arm.Start.Type property)
```

### COM Method Prototype

```
N/A
(use the Arm.Start.Type property)
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureStartArm (ViSession Vi,
                                       ViInt32 Type);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Unique identifier for an IVI session | `ViSession` |
| `Type` | Specifies the Start Arm type for armed measurements. The driver sets the Start Arm Type attribute to this value. See the attribute description for more information. | `ViInt32` |

### Return Values

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

## 4.3.26 Configure External Start Arm

### Description

Specifies the External Start Arm Source and configures the Level, Slope and Delay attributes.

### .NET Method Prototype

```
void ExternalStartArm.Configure (String source,
                                 Double level,
                                 Slope slope,
                                 PrecisionTimeSpan delay);
```

### COM Method Prototype

```
HRESULT Arm.Start.External.Configure
                                ([in] BSTR Source,
                                 [in] DOUBLE Level,
                                 [in] IviCounterSlopeEnum Slope,
                                 [in] DOUBLE Delay);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureExternalStartArm (ViSession Vi,
                                               ViConstString Source,
                                               ViReal64 Level,
                                               ViInt32 Slope,
                                               ViReal64 Delay);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Unique identifier for an IVI session | ViSession |
| Source | Specifies the External Start Arm source for armed measurements. The driver sets the External Start Arm Source attribute to this value. | ViConstString |
| Level | Specifies the Level you want the Counter to use for the External Start Arm. The driver sets the External Start Arm Level attribute to this value. See the attribute description for more information. | ViReal64 |
| Slope | Specifies the External Start Arm slope. The driver sets the External Start Arm Slope attribute to this value. See the attribute description for more information. | ViInt32 |
| Delay | Specifies the External Start Arm Delay. The driver sets the External Start Arm Delay attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

### Return Values

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

.NET Exceptions

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

## 4.3.27 Configure Stop Arm (IVI-C Only)

### Description

Configures the Stop Arm for armed measurements.

### .NET Method Prototype

```
N/A
(use the Arm.Stop.Type property)
```

### COM Method Prototype

```
N/A
(use the Arm.Stop.Type property)
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureStopArm (ViSession Vi
                                      ViInt32 Type);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Unique identifier for an IVI session | `ViSession` |
| `Type` | Specifies the Stop Arm type for armed measurements. The driver sets the Stop Arm attribute to this value. See the attribute description for more information. | `ViInt32` |

### Return Values

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

## 4.3.28 Configure External Stop Arm

### Description

Specifies the External Stop Arm Source and configures the Level, Slope and Delay attributes.

### .NET Method Prototype

```
void ExternalStopArm.Configure (String source,
                                Double level,
                                Slope slope,
                                PrecisionTimeSpan delay);
```

### COM Method Prototype

```
HRESULT Arm.Stop.External.Configure
                                ([in] BSTR Source,
                                 [in] DOUBLE Level,
                                 [in] IviCounterSlopeEnum Slope,
                                 [in] DOUBLE Delay);
```

### C Function Prototype

```
ViStatus IviCounter_ConfigureExternalStopArm (ViSession Vi,
                                              ViConstString Source,
                                              ViReal64 Level,
                                              ViInt32 Slope,
                                              ViReal64 Delay);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Unique identifier for an IVI session | ViSession |
| Source | Specifies the External Stop Arm source for armed measurements. The driver sets the External Stop Arm Source attribute to this value. | ViConstString |
| Level | Specifies the Level you want the Counter to use for the External Stop Arm. The driver sets the External Stop Arm Level attribute to this value. See the attribute description for more information. | ViReal64 |
| Slope | Specifies the External Stop Arm slope. The driver sets the External Stop Arm Slope attribute to this value. See the attribute description for more information. | ViInt32 |
| Delay | Specifies the External Stop Arm Delay. The driver sets the External Stop Arm Delay attribute to this value. See the attribute description for more information. | ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) |

### Return Values

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

.NET Exceptions

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

## 4.3.29 Fetch

### Description

Retrieves the result from a previously initiated measurement.

Use the Initiate function to start a measurement. The Is Measurement Complete function may be used to determine when the measurement is complete.

You can call the Read function instead of the Initiate function. The Read function starts a measurement. It then waits for the measurement to complete, obtains the measured value, and returns the measured value. You call this function separately for any other measurements that you want to obtain on a specific channel.

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

### .NET Method Prototype

```
Double Measurement.Fetch ();
```

### COM Method Prototype

```
HRESULT Measurement.Fetch ([out, retval] DOUBLE* Measurement);
```

### C Function Prototype

```
ViStatus IviCounter_Fetch (ViSession Vi,
                           ViReal64 *Measurement);
```

### Parameters

| Inputs | Description | Datatype |
|---|---|---|
| Vi | Instrument handle | ViSession |

| Outputs | Description | Datatype |
|---|---|---|
| Measurement | Returns the measured value. | ViReal64 |

### Return Values

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 Code | Description |
|---|---|
| Over Range | Measurement is over 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.

### Units

Units:  The units depend on the Measurement Function you select as shown below.

| Measurement Function | Units |
|---|---|
| Frequency | Hertz |
| Period | Seconds |
| Pulse Width | Seconds |
| Duty Cycle | Percent |
| Edge Time | Seconds |
| Frequency Ratio | Unitless |
| Time Interval | Seconds |
| Phase | Degrees |
| Totalize | Occurences |
| DC Voltage | Volts |
| RMS Voltage | Volts RMS |
| Peak-to-Peak Voltage | Volts Peak-to-Peak |

## 4.3.30 Initiate

### Description

Initiates a measurement based on the current configuration. You must configure the measurement type and input channel before calling this function. After you call this function, if the arm type is immediate the measurement commences immediately; if the arm type is external the Counter leaves the Idle state and waits for a start arm. To retrieve the measurement, call the Fetch function.

This function does not check the instrument status. Typically, you call this function only in a sequence of calls to other low-level driver functions. The sequence performs one operation. You use the low-level functions to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the IviCounter_error_query function at the conclusion of the sequence.

### .NET Method Prototype

```
void Measurement.Initiate ();
```

### COM Method Prototype

```
HRESULT Measurement.Initiate ();
```

### C Function Prototype

```
ViStatus IviCounter_Initiate (ViSession Vi);
```

### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |

### Return Values

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

### .NET Exceptions

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

### 4.3.31 Read

#### Description

Initiates and fetches a measurement based on the current configuration. Read waits a maximum of MaxTimeMilliseconds (C/COM) or maxTime (.NET) for the imstrument to return a measurement. Read generates an error if it exceeds the MaxTimeMilliseconds or maxTime.

#### .NET Method Prototype

```
Double Measurement.Read (PrecisionTimeSpan maximumTime);
```

#### COM Method Prototype

```
HRESULT Measurement.Read ([in] LONG MaxTimeMilliseconds,
                          [out, retval] DOUBLE* Measurement);
```

#### C Function Prototype

```
ViStatus IviCounter_Read (ViSession Vi,
                          ViInt32 MaxTimeMilliseconds,
                          ViReal64 *Measurement);
```

#### Parameters

| Inputs | Description | Datatype |
|---|---|---|
| `Vi` | Instrument handle | `ViSession` |
| `MaxTimeMilliseconds (C/COM)` | Pass the maximum length of time in which to allow the read waveform measurement operation to complete. If the operation does not complete within this time interval, the function returns the Maximum Time Exceeded error. When this occurs, you can call IviCounter_Abort to cancel the read waveform operation and return the Counter to the Idle state. | `ViInt32` |
| `maximumTime (.NET)` | Pass the maximum length of time in which to allow the read waveform measurement operation to complete. If the operation does not complete within this time interval, the method throws the Maximum Time Exceeded error. When this occurs, you can call IviCounter_Abort to cancel the read waveform operation and return the Counter to the Idle state. | `PrecisionTimeSpan` |

| Outputs | Description | Datatype |
|---|---|---|
| `Measurement` | Returns the measured value. | `ViReal64` |

Defined Values for MaxTimeMilliseconds Parameter (C/COM)

| 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. | | |
| | | C | IVICOUNTER_VAL_MAX_TIME_IMMEDIATE |
| | | COM | IviCounterMaxTimeImmediate |
| Max Time Infinite | Sets timeout to infinite. The function waits indefinitely for the measurement to complete. | | |
| | | C | IVICOUNTER_VAL_MAX_TIME_INFINITE |
| | | COM | IviCounterMaxTimeInfinite |

Defined Values for maximumTime Parameter (.NET)

| Name | Description | | |
|---|---|---|---|
| | | Language | Identifier |
| Max Time Immediate | Sets timeout to immediate. The function returns immediately. If no valid measurement value exists, the function throws the Max Time Exceeded eexception. | | |
| | | .NET | PrecisionTimeSpan.Zero |
| Max Time Infinite | Sets timeout to infinite. 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 counter timer 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.
Note that the .NET MaxTimeExceededException is defined in *IVI-3.2: Inherent Capabilities Specification*.

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

| Warning | Description |
|---|---|
| Measure Uncalibrated | This warning indicates that the data was captured while the counter timer was in an uncalibrated state. |

### Units

Units: The units depend on the Measurement Function you select as shown below.

| Measurement Function | Units |
|---|---|
| `Frequency` | Hertz |
| `Period` | Seconds |
| `Pulse Width` | Seconds |
| `Duty Cycle` | Percent |
| `Edge Time` | Seconds |
| `Frequency Ratio` | Unitless |
| `Time Interval` | Seconds |
| `Phase` | Degrees |
| `Totalize` | Occurences |
| `DC Voltage` | Volts |
| `RMS Voltage` | Volts RMS |
| `Peak-to-Peak Voltage` | Volts Peak-to-Peak |

### Compliance Notes

An IviCounter specific driver is not required to implement the Max Time Immediate or the Max Time Infinite defined values for the `MaxTimeMilliseconds` parameter to be compliant with the IviCounterBase capability group.

## 4.4 IviCounterBase Behavior Model

The following state diagram shows relationships between IviCounter frequency measurement capabilities (i.e. non-totalize measurements) and counter behavior.

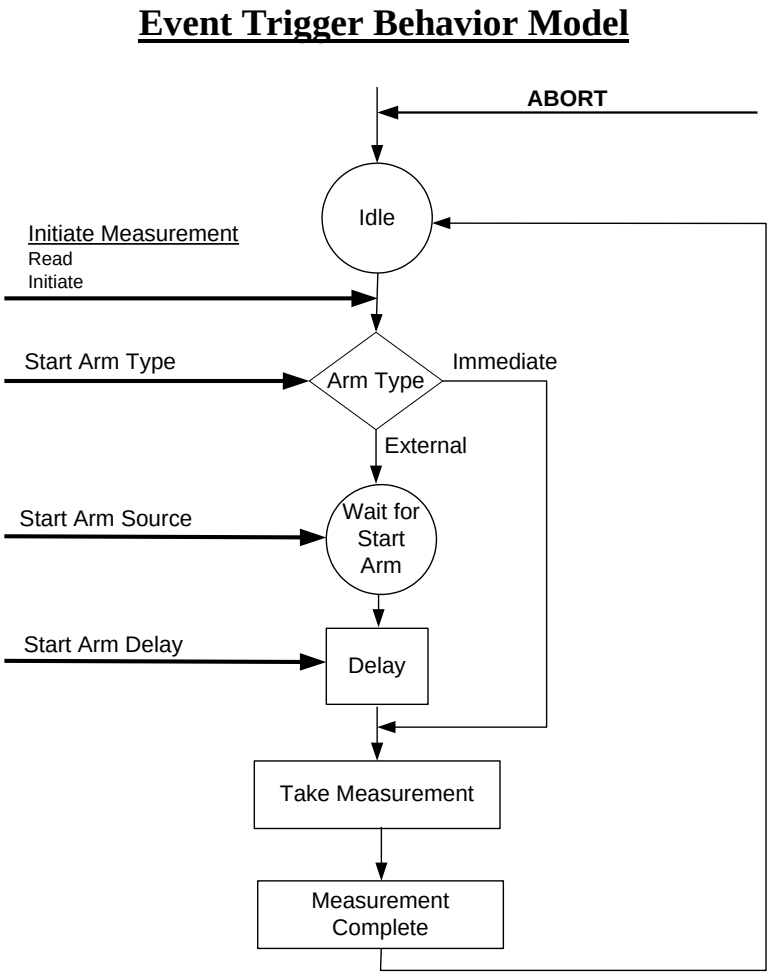



**Figure 4-2.** IviCounter Event Trigger Behavior Model

The following state diagram shows relationships between IviCounter totalize measurement capabilities (i.e. non-frequency measurements) and analyzer behavior.

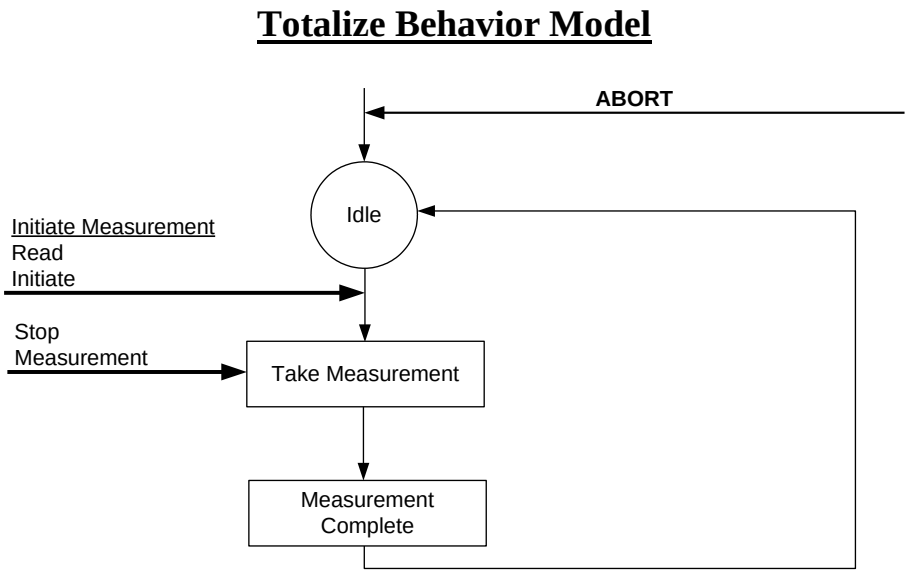



**Figure 4-3.** IviCounter Totalize Behavior Model

The main state in the IviCounter Class is the Idle state. The Counter 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 Counter while it is in the Idle state by calling the configure measurement functions. Each measurement has corresponding estimate and resolution parameters that characterize the measurement. The estimate and resolution are used within the driver to calculate and set physical instrument parameters. For example, the Frequency Estimate and Resolution for the Measure Frequency function provide the necessary information for calculating instrument parameters such as gate time, averaging on or off, digits of resolution, etc. The IviCounter attributes can also be configured individually with the Set Attribute function.

The Measure, Read, and Initiate functions cause the Counter to leave the Idle state and transition to the *Wait-For-End-Of-Measurement* state. The Measure and Read functions do not return until the measurement process is complete and the Counter has returned to the Idle state. The Initiate function returns as soon as the Counter leaves the Idle state.

The Fetch function is used to retrieve data measurements that were initiated by the Initiate function. The measurement data returned from the Measure, Read, and Fetch functions is acquired after the Counter has left *Wait-For-End-Of-Measurement* state.

# 5 IviCounterFilter Extension Group

## 5.1 IviCounterFilter Extension Group Overview

The IviCounterFilter extension group defines extensions for setting the minimum and maximum filter frequencies for the input signal.

## 5.2 IviCounterFilter Attributes

The IviCounterFilter capability group defines the following attributes:

- Minimum Frequency
- Maximum Frequency

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 8, *IviCounter Attribute ID Definitions.*

### 5.2.1 Minimum Frequency

| Data Type | Access | Applies To | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | Down | Configure Filter |

#### .NET Property Name

```
Channels[].MinimumFrequency
```

#### COM Property Name

```
Channels.Item().MinimumFrequency
```

#### C Constant Name

```
IVICOUNTER_ATTR_FILTER_MINIMUM_FREQUENCY
```

#### Description

Specifies the low cutoff frequency for the filter in hertz. Set to zero to disable low frequency filtering.

#### .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. MinimumFrequency must be less than the MaximumFrequency. If zero is specified for the minimum frequency, the instruments minimum frequency shall be used.

### 5.2.2 Maximum Frequency

| Data Type | Access | Applies To | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 | R/W | Channels | Up | Configure Filter |

#### .NET Property Name

```
Channels[].MaximumFrequency
```

#### COM Property Name

```
Channels.Item().MaximumFrequency
```

#### C Constant Name

```
IVICOUNTER_ATTR_FILTER_MAXIMUM_FREQUENCY
```

#### Description

Specifies the high cutoff frequency for the filter in hertz. Set to positive infinity to disable high frequency filtering.

#### .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. MaximumFrequency must be greater than the MinimumFrequency. If positive infinity is specified for the maximum frequency, the instruments maximum frequency shall be used.

## 5.3 IviCounterFilter Function

The IviCounterFilter capability group defines the following extended function:

- Configure Filter

This section describes the behavior and requirements of the function.

### 5.3.1 Configure Filter

#### Description

Sets the bandpass filter minimum and maximum frequencies for a channel.

#### .NET Method Prototype

```
void Channels[].ConfigureFilter (Double minimumFrequency,
                                 Double maximumFrequency);
```

#### COM Method Prototype

```
HRESULT Channels.Item().ConfigureFilter ([in] DOUBLE MinimumFrequency,
                                         [in] DOUBLE MaximumFrequency);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureFilter (ViSession Vi,
                                     ViConstString Channel,
                                     ViReal64 MinimumFrequency,
                                     ViReal64 MaximumFrequency);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| Vi | Instrument handle. | ViSession |
| Channel | Specifies the channel name you want to configure. | ViConstString |
| MinimumFrequency | Specifies the minimum filter frequency. The driver uses this value to set the Minimum Frequency attribute. See the attribute description for more details. | ViReal64 |
| MaximumFrequency | Specifies the maximum filter frequency. The driver uses this value to set the Maximum Frequency attribute. See the attribute description for more details. | ViReal64 |

#### Return Values

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.

## 5.4 IviCounterFilter Behavior Model

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

## 5.5 IviCounterFilter Compliance Notes

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

# 6 IviCounterTimeIntervalStopHoldoff Extension Group

## 6.1 IviCounterTimeIntervalStopHoldoff Extension Group Overview

The IviCounterTimeIntervalStopHoldoff extension group defines attributes and functions for setting the delay time for the Time Interval functions.

## 6.2 IviCounterTimeIntervalStopHoldoff Attribute

The IviCounterTimeIntervalStopHoldoff capability group defines the following attribute:

- Time Interval Stop Holdoff

This section describes the behavior and requirements of the attribute. The actual value for the attribute ID is defined in Section 8, *IviCounter Attribute ID Definitions.*

### 6.2.1 Time Interval Stop Holdoff

| Data Type | Access | Applies To | Coercion | High Level Functions |
|---|---|---|---|---|
| ViReal64 (C/COM)<br>PrecisionTimeSpan (.NET) | R/W | N/A | See Note | Configure Time Interval Stop Holdoff |

#### .NET Property Name

```
TimeInterval.StopHoldoff
```

#### COM Property Name

```
TimeInterval.StopHoldoff
```

#### C Constant Name

```
IVICOUNTER_ATTR_TIME_INTERVAL_STOP_HOLDOFF
```

#### Description

Specifies the stop holdoff time for a Time Interval measurement. The stop holdoff time is the time from the Time Interval Start Channel Trigger until the Time Interval Stop Channel Trigger is enabled. For C and COM, the units are seconds. For .NET, the units are implicit in the PrecisionTimeSpan type.

**Note**: Many counters have a small, non-zero value as the minimum value for this attribute. To configure the instrument to use the shortest stop hold-off, the user can specify a value of zero for this attribute. Therefore, the IVI Class-Compliant specific driver shall coerce any value between zero and the minimum value to the minimum value. No other coercion is allowed on this attribute..

#### .NET Exceptions

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

## 6.3 IviCounterTimeIntervalStopHoldoff Function

The IviCounterTimeIntervalStopHoldoff capability group defines the following extended function:

- Configure Time Interval Stop Holdoff (IVI-C)

This section describes the behavior and requirements of the function.

### 6.3.1 Configure Time Interval Stop Holdoff (IVI-C Only)

#### Description

Configures the Time Interval Stop Holdoff time in seconds.

#### .NET Method Prototype

```
N/A
(Use the TimeInterval.StopHoldoff Property)
```

#### COM Method Prototype

```
N/A
(Use the TimeInterval.StopHoldoff Property)
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureTimeIntervalStopHoldoff (ViSession Vi,
                                                      ViReal64 Time);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Time` | Specifies the value of the Stop Holdoff in seconds | `ViReal64` |

#### Return Values

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

## 6.4 IviCounterTimeIntervalStopHoldoff Behavior Model

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

## 6.5 IviCounterTimeIntervalStopHoldoff Compliance Notes

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

# 7 IviCounterVoltageMeasurement Extension Group

## 7.1 IviCounterVoltageMeasurement Extension Group Overview

The IviCounterVoltageMeasurement extension group defines extensions for making voltage measurements on the input signal. The following measurements are provided:

- DC Voltage
- Maximum Voltage
- Minimum Voltage
- RMS Voltage
- Peak-to-Peak Voltage

## 7.2 IviCounterVoltageMeasurement Attributes

The IviCounterVoltageMeasurement capability group defines the following attributes:

- Voltage Channel
- Voltage Estimate
- Voltage Resolution

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 9, *IviCounter Attribute ID Definitions.*

### 7.2.1 Voltage Channel

| Data Type | Access | Applies to | Coercion | High Level Functions |
|---|---|---|---|---|
| ViString | R/W | N/A | None | Configure Voltage |

.NET Property Name

Voltage.Channel

COM Property Name

Voltage.Channel

C Constant Name

IVICOUNTER_ATTR_VOLTAGE_CHANNEL

Description

Specifies the input channel the voltage is measured on.

**.NET Exceptions**

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

### 7.2.2 Voltage Estimate

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

.NET Property Name

```
Voltage.Estimate
```

COM Property Name

```
Voltage.Estimate
```

C Constant Name

```
IVICOUNTER_ATTR_VOLTAGE_ESTIMATE
```

Description

Specifies the estimated voltage, in volts, for the voltage function.

**.NET Exceptions**

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

### 7.2.3 Voltage Resolution

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

.NET Property Name

```
Voltage.Resolution
```

COM Property Name

```
Voltage.Resolution
```

C Constant Name

```
IVICOUNTER_ATTR_VOLTAGE_RESOLUTION
```

Description

Specifies the resolution of the measurement, in volts, for the voltage function.

**.NET Exceptions**

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

## 7.3 IviCounterVoltageMeasurement Function

The IviCounterVoltageMeasurement capability group defines the following extended function:

- Configure Voltage

This section describes the behavior and requirements of the function.

### 7.3.1 Configure Voltage Measurement

#### Description

Configures the voltage function, the estimate, and the resolution attributes for a voltage measurement.

#### .NET Method Prototype

```
void Voltage.Configure (String channel,
                        MeasurementFunction measurementFunction,
                        Double estimate,
                        Double resolution);
```

#### COM Method Prototype

```
HRESULT Voltage.Configure ([in] BSTR Channel,
                           [in] IviCounterMeasurementFunctionEnum Function,
                           [in] DOUBLE Estimate,
                           [in] DOUBLE Resolution);
```

#### C Function Prototype

```
ViStatus IviCounter_ConfigureVoltage (ViSession Vi,
                                      ViConstString Channel,
                                      ViInt32 Function,
                                      ViReal64 Estimate,
                                      ViReal64 Resolution);
```

#### Parameters

| Inputs | Description | Base Type |
|---|---|---|
| `Vi` | Instrument handle. | `ViSession` |
| `Channel` | Specifies the channel the voltage will be measured on. The driver sets the Voltage Channel attribute to this value. | `ViConstString` |
| `Function (C/COM)`<br>`measurementFunction (.NET)` | Specifies the voltage function. Valid functions: DC Voltage, Maximum Voltage, Minimum Voltage, RMS Voltage, and Peak-to-Peak Voltage. The driver sets the Measurement Function attribute to this value. See Section 4.2.1, *Measurement Function* for details. | `ViInt32` |
| `Estimate` | Specifies the estimated voltage. The driver sets the Voltage Estimate attribute to this value. | `ViReal64` |
| `Resolution` | Specifies the resolution of the voltage measurement. It is the quantization size, i.e. the smallest delta value that can be detected. The driver sets the Voltage Resolution attribute to this value. | `ViReal64` |

#### Return Values

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

.NET Exceptions

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

## 7.4 IviCounterVoltageMeasurement Behavior Model

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

## 7.5 IviCounterVoltageMeasurement Compliance Notes

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

# 8 IviCounterEdgeTimeReferenceLevels Extension Group

## 8.1 IviCounterEdgeTimeReferenceLevels Extension Group Overview

The IviCounterEdgeTimeReferenceLevels extension implements percentage base edge time measurements. The base IviCounterEdgeTimeReferenceLevels function is used and the valid ReferenceType value is Percent.

## 8.2 IviCounterEdgeTimeReferenceLevels Behavior Model

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

## 8.3 IviCounterEdgeTimeReferenceLevels Compliance Notes

1. For a specific driver to comply with the IviCounterEdgeTimeReferenceLevels extension, it shall be compliant with the IviCounterBase capability group

# 9 IviCounter Attribute ID Definitions

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

**Table 9-1.** IviCounter Attributes ID Values

| Attribute Name | ID Definition |
|---|---|
| IVICOUNTER_ATTR_MEASUREMENT_FUNCTION | IVI_CLASS_ATTR_BASE + 3 |
| IVICOUNTER_ATTR_CHANNEL_COUNT | IVI_INHERENT_ATTR_BASE + 203 |
| IVICOUNTER_ATTR_IMPEDANCE | IVI_CLASS_ATTR_BASE + 4 |
| IVICOUNTER_ATTR_COUPLING | IVI_CLASS_ATTR_BASE + 5 |
| IVICOUNTER_ATTR_ATTENUATION | IVI_CLASS_ATTR_BASE + 6 |
| IVICOUNTER_ATTR_CHANNEL_LEVEL | IVI_CLASS_ATTR_BASE + 7 |
| IVICOUNTER_ATTR_CHANNEL_HYSTERESIS | IVI_CLASS_ATTR_BASE + 8 |
| IVICOUNTER_ATTR_CHANNEL_SLOPE | IVI_CLASS_ATTR_BASE + 9 |
| IVICOUNTER_ATTR_FILTER_ENABLED | IVI_CLASS_ATTR_BASE + 10 |
| IVICOUNTER_ATTR_FREQUENCY_CHANNEL | IVI_CLASS_ATTR_BASE + 11 |
| IVICOUNTER_ATTR_FREQUENCY_ESTIMATE | IVI_CLASS_ATTR_BASE + 12 |
| IVICOUNTER_ATTR_FREQUENCY_RESOLUTION | IVI_CLASS_ATTR_BASE + 13 |
| IVICOUNTER_ATTR_FREQUENCY_APERTURE_TIME | IVI_CLASS_ATTR_BASE + 14 |
| IVICOUNTER_ATTR_FREQUENCY_ESTIMATE_AUTO | IVI_CLASS_ATTR_BASE + 15 |
| IVICOUNTER_ATTR_FREQUENCY_RESOLUTION_AUTO | IVI_CLASS_ATTR_BASE + 16 |
| IVICOUNTER_ATTR_PERIOD_CHANNEL | IVI_CLASS_ATTR_BASE + 18 |
| IVICOUNTER_ATTR_PERIOD_ESTIMATE | IVI_CLASS_ATTR_BASE + 19 |
| IVICOUNTER_ATTR_PERIOD_RESOLUTION | IVI_CLASS_ATTR_BASE + 20 |
| IVICOUNTER_ATTR_PERIOD_APERTURE_TIME | IVI_CLASS_ATTR_BASE + 21 |
| IVICOUNTER_ATTR_PULSE_WIDTH_CHANNEL | IVI_CLASS_ATTR_BASE + 22 |
| IVICOUNTER_ATTR_PULSE_WIDTH_ESTIMATE | IVI_CLASS_ATTR_BASE + 23 |
| IVICOUNTER_ATTR_PULSE_WIDTH_RESOLUTION | IVI_CLASS_ATTR_BASE + 24 |
| IVICOUNTER_ATTR_DUTY_CYCLE_CHANNEL | IVI_CLASS_ATTR_BASE + 25 |
| IVICOUNTER_ATTR_DUTY_CYCLE_FREQUENCY_ESTIMATE | IVI_CLASS_ATTR_BASE + 26 |
| IVICOUNTER_ATTR_DUTY_CYCLE_RESOLUTION | IVI_CLASS_ATTR_BASE + 27 |
| IVICOUNTER_ATTR_EDGE_TIME_CHANNEL | IVI_CLASS_ATTR_BASE + 28 |
| IVICOUNTER_ATTR_EDGE_TIME_REFERENCE_TYPE | IVI_CLASS_ATTR_BASE + 29 |
| IVICOUNTER_ATTR_EDGE_TIME_ESTIMATE | IVI_CLASS_ATTR_BASE + 30 |
| IVICOUNTER_ATTR_EDGE_TIME_RESOLUTION | IVI_CLASS_ATTR_BASE + 31 |
| IVICOUNTER_ATTR_EDGE_TIME_HIGH_REFERENCE | IVI_CLASS_ATTR_BASE + 32 |

**Table 9-1.** IviCounter Attributes ID Values

| Attribute Name | ID Definition |
|---|---|
| IVICOUNTER_ATTR_EDGE_TIME_LOW_REFERENCE | IVI_CLASS_ATTR_BASE + 33 |
| IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_CHANNEL | IVI_CLASS_ATTR_BASE + 34 |
| IVICOUNTER_ATTR_FREQUENCY_RATIO_DENOMINATOR_CHANNEL | IVI_CLASS_ATTR_BASE + 35 |
| IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_FREQUENCY_ESTIMATE | IVI_CLASS_ATTR_BASE + 36 |
| IVICOUNTER_ATTR_FREQUENCY_RATIO_ESTIMATE | IVI_CLASS_ATTR_BASE + 37 |
| IVICOUNTER_ATTR_FREQUENCY_RATIO_RESOLUTION | IVI_CLASS_ATTR_BASE + 38 |
| IVICOUNTER_ATTR_TIME_INTERVAL_START_CHANNEL | IVI_CLASS_ATTR_BASE + 39 |
| IVICOUNTER_ATTR_TIME_INTERVAL_STOP_CHANNEL | IVI_CLASS_ATTR_BASE + 40 |
| IVICOUNTER_ATTR_TIME_INTERVAL_ESTIMATE | IVI_CLASS_ATTR_BASE + 41 |
| IVICOUNTER_ATTR_TIME_INTERVAL_RESOLUTION | IVI_CLASS_ATTR_BASE + 42 |
| IVICOUNTER_ATTR_PHASE_INPUT_CHANNEL | IVI_CLASS_ATTR_BASE + 43 |
| IVICOUNTER_ATTR_PHASE_REFERENCE_CHANNEL | IVI_CLASS_ATTR_BASE + 44 |
| IVICOUNTER_ATTR_PHASE_FREQUENCY_ESTIMATE | IVI_CLASS_ATTR_BASE + 45 |
| IVICOUNTER_ATTR_PHASE_RESOLUTION | IVI_CLASS_ATTR_BASE + 46 |
| IVICOUNTER_ATTR_CONTINUOUS_TOTALIZE_CHANNEL | IVI_CLASS_ATTR_BASE + 47 |
| IVICOUNTER_ATTR_GATED_TOTALIZE_CHANNEL | IVI_CLASS_ATTR_BASE + 48 |
| IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SOURCE | IVI_CLASS_ATTR_BASE + 49 |
| IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SLOPE | IVI_CLASS_ATTR_BASE + 50 |
| IVICOUNTER_ATTR_TIMED_TOTALIZE_CHANNEL | IVI_CLASS_ATTR_BASE + 51 |
| IVICOUNTER_ATTR_TIMED_TOTALIZE_GATE_TIME | IVI_CLASS_ATTR_BASE + 52 |
| IVICOUNTER_ATTR_START_ARM_TYPE | IVI_CLASS_ATTR_BASE + 53 |
| IVICOUNTER_ATTR_EXTERNAL_START_ARM_SOURCE | IVI_CLASS_ATTR_BASE + 54 |
| IVICOUNTER_ATTR_EXTERNAL_START_ARM_LEVEL | IVI_CLASS_ATTR_BASE + 55 |
| IVICOUNTER_ATTR_EXTERNAL_START_ARM_SLOPE | IVI_CLASS_ATTR_BASE + 56 |
| IVICOUNTER_ATTR_EXTERNAL_START_ARM_DELAY | IVI_CLASS_ATTR_BASE + 57 |
| IVICOUNTER_ATTR_STOP_ARM_TYPE | IVI_CLASS_ATTR_BASE + 58 |
| IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SOURCE | IVI_CLASS_ATTR_BASE + 59 |
| IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_LEVEL | IVI_CLASS_ATTR_BASE + 60 |
| IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SLOPE | IVI_CLASS_ATTR_BASE + 61 |
| IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_DELAY | IVI_CLASS_ATTR_BASE + 62 |
| IVICOUNTER_ATTR_FILTER_MINIMUM_FREQUENCY | IVI_CLASS_ATTR_BASE + 501 |
| IVICOUNTER_ATTR_FILTER_MAXIMUM_FREQUENCY | IVI_CLASS_ATTR_BASE + 502 |
| IVICOUNTER_ATTR_TIME_INTERVAL_STOP_HOLDOFF | IVI_CLASS_ATTR_BASE + 601 |
| IVICOUNTER_ATTR_VOLTAGE_CHANNEL | IVI_CLASS_ATTR_BASE + 701 |

**Table 9-1.** IviCounter Attributes ID Values

| Attribute Name | ID Definition |
|---|---|
| IVICOUNTER_ATTR_VOLTAGE_ESTIMATE | IVI_CLASS_ATTR_BASE + 702 |
| IVICOUNTER_ATTR_VOLTAGE_RESOLUTION | IVI_CLASS_ATTR_BASE + 703 |

# 10 IviCounter Attribute Value Definitions

This section specifies the actual value for each defined attribute value.

Coupling

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| AC | C | IVICOUNTER_VAL_AC | 1 |
| | COM | IviCounterCouplingAC | 1 |
| | .NET | Coupling.AC | 0 |
| DC | C | IVICOUNTER_VAL_DC | 2 |
| | COM | IviCounterCouplingDC | 2 |
| | .NET | Coupling.DC | 1 |
| Coupling Class Extension Base | C | IVICOUNTER_VAL_COUPLING_CLASS_EXT_BASE | 500 |
| Coupling Specific Extension Base | C | IVICOUNTER_VAL_COUPLING_SPECIFIC_EXT_BASE | 1000 |
| | COM | | 1000 |

Slope

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Negative | .NET | Slope.Negative | 1 |
| | C | IVICOUNTER_VAL_NEGATIVE | 0 |
| | COM | IviCounterSlopeNegative | 0 |
| Positive | .NET | Slope.Positive | 0 |
| | C | IVICOUNTER_VAL_POSITIVE | 1 |
| | COM | IviCounterSlopePositive | 1 |
| Slope Class Extension Base | C | IVICOUNTER_VAL_SLOPE_CLASS_EXT_BASE | 500 |
| Slope Specific Extension Base | C | IVICOUNTER_VAL_SLOPE_SPECIFIC_EXT_BASE | 1000 |
| | COM | | 1000 |

Measurement Function

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Frequency | C | IVICOUNTER_VAL_FREQUENCY | 1 |
| | COM | IviCounterFunctionFrequency | 1 |
| | .NET | MeasurementFunction.Frequency | 0 |
| Frequency with Aperture | C | IVICOUNTER_VAL_FREQUENCY_WITH_APERTURE | 2 |
| | COM | IviCounterFunctionFrequencyWithAperture | 2 |
| | .NET | MeasurementFunction.FrequencyWithAperture | 1 |
| Period | C | IVICOUNTER_VAL_PERIOD | 3 |
| | COM | IviCounterFunctionPeriod | 3 |
| | .NET | MeasurementFunction.Period | 2 |
| Period with Aperture | C | IVICOUNTER_VAL_PERIOD_WITH_APERTURE | 4 |
| | COM | IviCounterFunctionPeriodWithAperture | 4 |
| | .NET | MeasurementFunction.PeriodWithAperture | 3 |
| Pulse Width | C | IVICOUNTER_VAL_PULSE_WIDTH | 5 |
| | COM | IviCounterFunctionPulseWidth | 5 |
| | .NET | MeasurementFunction.PulseWidth | 4 |
| Duty Cycle | C | IVICOUNTER_VAL_DUTY_CYCLE | 6 |
| | COM | IviCounterFunctionDutyCycle | 6 |
| | .NET | MeasurementFunction.DutyCycle | 5 |
| Edge Time | C | IVICOUNTER_VAL_EDGE_TIME | 7 |
| | COM | IviCounterFunctionEdgeTime | 7 |
| | .NET | MeasurementFunction.EdgeTime | 6 |
| Frequency Ratio | C | IVICOUNTER_VAL_FREQUENCY_RATIO | 8 |
| | COM | IviCounterFunctionFrequencyRatio | 8 |
| | .NET | MeasurementFunction.FrequencyRatio | 7 |
| Time Interval | C | IVICOUNTER_VAL_TIME_INTERVAL | 9 |
| | COM | IviCounterFunctionTimeInterval | 9 |
| | .NET | MeasurementFunction.TimeInterval | 8 |
| Phase | C | IVICOUNTER_VAL_PHASE | 10 |
| | COM | IviCounterFunctionPhase | 10 |
| | .NET | MeasurementFunction.Phase | 9 |
| Continuous Totalize | C | IVICOUNTER_VAL_CONTINUOUS_TOTALIZE | 11 |
| | COM | IviCounterFunctionContinuousTotalize | 11 |
| | .NET | MeasurementFunction.ContinuousTotalize | 10 |
| Gated Totalize | C | IVICOUNTER_VAL_GATED_TOTALIZE | 12 |
| | COM | IviCounterFunctionGatedTotalize | 12 |
| | .NET | MeasurementFunction.GatedTotalize | 11 |

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Timed Totalize | C | IVICOUNTER_VAL_TIMED_TOTALIZE | 13 |
| | COM | IviCounterFunctionTimedTotalize | 13 |
| | .NET | MeasurementFunction.TimedTotalize | 12 |
| DC Voltage | C | IVICOUNTER_VAL_DC_VOLTAGE | 14 |
| | COM | IviCounterFunctionDCVoltage | 14 |
| | .NET | MeasurementFunction.DCVoltage | 13 |
| Maximum Voltage | C | IVICOUNTER_VAL_MAXIMUM_VOLTAGE | 15 |
| | COM | IviCounterFunctionMaximumVoltage | 15 |
| | .NET | MeasurementFunction.MaximumVoltage | 14 |
| Minimum Voltage | C | IVICOUNTER_VAL_MINIMUM_VOLTAGE | 16 |
| | COM | IviCounterFunctionMinimumVoltage | 16 |
| | .NET | MeasurementFunction.MinimumVoltage | 15 |
| RMS Voltage | C | IVICOUNTER_VAL_RMS_VOLTAGE | 17 |
| | COM | IviCounterFunctionRMSVoltage | 17 |
| | .NET | MeasurementFunction.RMSVoltage | 16 |
| Peak-to-Peak Voltage | C | IVICOUNTER_VAL_PEAK_TO_PEAK_VOLTAGE | 18 |
| | COM | IviCounterFunctionPeakToPeakVoltage | 18 |
| | .NET | MeasurementFunction.PeakToPeakVoltage | 17 |
| Measurement Function Class Extension Base | C | IVICOUNTER_VAL_MEASUREMENT_FUNCTION_CLASS_EXT_BASE | 500 |
| Measurement Function Specific Extension Base | C | IVICOUNTER_VAL_MEASUREMENT_FUNCTION_SPECIFIC_EXT_BASE | 1000 |
| | COM | | 1000 |

Arm Type

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Immediate | C | IVICOUNTER_VAL_IMMEDIATE_ARM_TYPE | 1 |
| | COM | IviCounterArmTypeImmediate | 1 |
| | .NET | ArmType.Immediate | 0 |
| External | C | IVICOUNTER_VAL_EXTERNAL_ARM_TYPE | 2 |
| | COM | IviCounterArmTypeExternal | 2 |
| | .NET | ArmType.External | 1 |

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Arm Class Extension Base | C | IVICOUNTER_VAL_ARM_TYPE_CLASS_EXT_BASE | 500 |
| Arm Specific Extension Base | C | IVICOUNTER_VAL_ARM_TYPE_SPECIFIC_EXT_BASE | 1000 |
| | COM | | 1000 |

Reference Type

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Voltage | C | IVICOUNTER_VAL_VOLTAGE_REFERENCE_TYPE | 1 |
| | COM | IviCounterReferenceTypeVoltage | 1 |
| | .NET | ReferenceType.Voltage | 0 |
| Percent | C | IVICOUNTER_VAL_PERCENT_REFERENCE_TYPE | 2 |
| | COM | IviCounterReferenceTypePercent | 2 |
| | .NET | ReferenceType.Percent | 1 |
| Reference Class Extension Base | C | IVICOUNTER_VAL_REFERENCE_TYPE_CLASS_EXT_BASE | 500 |
| Reference Specific Extension Base | C | IVICOUNTER_VAL_REFERENCE_TYPE_SPECIFIC_EXT_BASE | 1000 |
| | COM | | 1000 |

# 11 IviCounter Function Parameter Value Definitions

This section specifies the actual values for each function parameter that defines values.

Maximum Time

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Max Time Immediate | C | IVICOUNTER_VAL_MAX_TIME_IMMEDIATE | 0 |
| | COM | IviCounterMaxTimeImmediate | 0 |
| Max Time Infinite | C | IVICOUNTER_VAL_MAX_TIME_INFINITE | -1 |
| | COM | IviCounterMaxTimeInfinite | -1 |

Measurement Status

Parameter: Status

| *Value Name* | *Language* | *Identifier* | *Actual Value* |
|---|---|---|---|
| Measurement Complete | C | IVICOUNTER_VAL_MEASUREMENT_COMPLETE | 1 |
| | COM | IviCounterMeasurementStatusComplete | 1 |
| | .NET | MeasurementStatus.Complete | 1 |
| Measurement In Progress | C | IVICOUNTER_VAL_MEASUREMENT_IN_PROGRESS | 0 |
| | COM | IviCounterMeasurementStatusInProgress | 0 |
| | .NET | MeasurementStatus.InProgress | 0 |
| Measurement Status Unknown | C | IVICOUNTER_VAL_MEASUREMENT_STATUS_UNKNOWN | -1 |
| | COM | IviCounterMeasurementStatusUnknown | -1 |
| | .NET | MeasurementStatus.Unknown | 2 |

# 12 IviCounter Error, Completion Code, and Exception Class Definitions

The table below specifies the actual value for each status code, and the actual exception class and warning GUID for each IVI.NET exception and warning that the IviCounter class specification defines. Note that warnings are raised as events in .NET. See IVI-3.1 and IVI-3.2 for details.

**Table 12-1.** IviCounter Completion Codes

| *Error Name* | *Description* | | | |
|---|---|---|---|---|
| | | *Language* | *Identifier* | *Value(hex)* |
| Measure Uncalibrated Warning | Uncalibrated measurement | | | |
| | | C | IVICOUNTER_WARN_MEASURE_UNCALIBRATED | 0x3FFA2001 |
| | | COM | S_IVICOUNTER_MEASURE_UNCALIBRATED | 0x00042001 |
| | | .NET | N/A | {2D606B73-601F-40f6-AE07-2FDB5F086A09} |
| Over Range Warning | Measurement overrange | | | |
| | | C | IVICOUNTER_WARN_OVER_RANGE | 0x3FFA2002 |
| | | COM | S_IVICOUNTER_OVER_RANGE | 0x00042002 |
| | | .NET | N/A | N/A |
| Max Time Exceeded | Max Time Exceeded | | | |
| | | C | IVICOUNTER_ERROR_MAX_TIME_EXCEEDED | 0xBFFA2003 |
| | | COM | E_IVICOUNTER_MAX_TIME_EXCEEDED | 0x80042003 |
| | | .NET | Ivi.Driver.MaxTimeExceededException | IVI defined exception (see IVI 3.2) |

Table 12-2 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. For .NET, exception default message strings are defined with the exception.

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

**Table 12-2.** IviCounter 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." |
| Max Time Exceeded | "%s: The maximum waiting time for this operation was exceeded." |

# 13 IviCounter Hierarchies

## *13.1 IviCounter COM Hierarchy*

**Table 13-1.** IviCounter COM Hierarchy

| COM Interface Hierarchy | Generic Name | Type |
|---|---|---|
| Function | Measurement Function | P |
| **Arm** | | |
| **Start** | | |
| Type | Start Arm Type | P |
| **External** | | |
| Configure | Configure External Start Arm | M |
| Source | External Start Arm Source | P |
| Level | External Start Arm Level | P |
| Slope | External Start Arm Slope | P |
| Delay | External Start Arm Delay | P |
| **Stop** | | |
| Type | Stop Arm Type | P |
| **External** | | |
| Configure | Configure External Stop Arm | M |
| Source | External Stop Arm Source | P |
| Level | External Stop Arm Level | P |
| Slope | External Stop Arm Slope | P |
| Delay | External Stop Arm Delay | P |
| **Channels** | | |
| Count | Channel Count | P |
| Name | Channel Name | P |
| Item | Channel Item | P |
| **Item** | | |
| Configure | Configure Channel | M |
| Impedance | Channel Impedance | P |
| Coupling | Channel Coupling | P |
| Attenuation | Channel Attenuation | P |
| ConfigureLevel | Configure Level | M |
| Level | Channel Level | P |
| Hysteresis | Channel Hysteresis | P |
| Slope | Channel Slope | P |
| FilterEnabled | Channel Filter Enabled | P |

| Table 13-1. IviCounter COM Hierarchy | | |
|---|---|---|
| **COM Interface Hierarchy** | **Generic Name** | **Type** |
| ConfigureFilter | Configure Filter | M |
| MaximumFrequency | Maximum Frequency | P |
| MinimumFrequency | Minimum Frequency | P |
| **DutyCycle** | | |
| Configure | Configure Duty Cycle | M |
| Channel | Duty Cycle Channel | P |
| FrequencyEstimate | Duty Cycle Frequency Estimate | P |
| Resolution | Duty Cycle Resolution | P |
| **EdgeTime** | | |
| Configure | Configure Edge Time | M |
| ConfigureReferenceLevels | Configure Edge Time Reference Levels | M |
| Channel | Edge Time Channel | P |
| ReferenceType | Edge Time Reference Type | P |
| Estimate | Edge Time Estimate | P |
| Resolution | Edge Time Resolution | P |
| HighReference | Edge Time High Reference | P |
| LowReference | Edge Time Low Reference | P |
| **Frequency** | | |
| Configure | Configure Frequency | M |
| ConfigureManual | Configure Manual Frequency | M |
| Channel | Frequency Channel | P |
| Estimate | Frequency Estimate | P |
| Resolution | Frequency Resolution | P |
| ConfigureWithAperture | Configure Frequency with Aperture Time | M |
| ApertureTime | Frequency Aperture Time | P |
| EstimateAuto | Frequency Estimate Auto | P |
| ResolutionAuto | Frequency Resolution Auto | P |
| **FrequencyRatio** | | |
| Configure | Configure Frequency Ratio | M |
| NumeratorChannel | Frequency Ratio Numerator Channel | P |
| DenominatorChannel | Frequency Ratio Denominator Channel | P |
| NumeratorFrequencyEstimate | Frequency Ratio Numerator Frequency Estimate | P |
| Estimate | Frequency Ratio Estimate | P |
| Resolution | Frequency Ratio Resolution | P |
| **Measurement** | | |
| IsMeasurementComplete | Is Measurement Complete | M |

| **Table 13-1.** IviCounter COM Hierarchy | | |
|---|---|---|
| **COM Interface Hierarchy** | **Generic Name** | **Type** |
| Read | Read | M |
| Initiate | Initiate | M |
| Fetch | Fetch | M |
| Abort | Abort | M |
| **Period** | | |
| Configure | Configure Period | M |
| Channel | Period Channel | P |
| Estimate | Period Estimate | P |
| Resolution | Period Resolution | P |
| ConfigureWithAperture | Configure Period with Aperture Time | M |
| ApertureTime | Period Aperture Time | P |
| **Phase** | | |
| Configure | Configure Phase | M |
| InputChannel | Phase Input Channel | P |
| ReferenceChannel | Phase Reference Channel | P |
| FrequencyEstimate | Phase Frequency Estimate | P |
| Resolution | Phase Resolution | P |
| **PulseWidth** | | |
| Configure | Configure Pulse Width | M |
| Channel | Pulse Width Channel | P |
| Estimate | Pulse Width Estimate | P |
| Resolution | Pulse Width Resolution | P |
| **TimeInterval** | | |
| Configure | Configure Time Interval | M |
| StartChannel | Time Interval Start Channel | P |
| StopChannel | Time Interval Stop Channel | P |
| Estimate | Time Interval Estimate | P |
| Resolution | Time Interval Resolution | P |
| StopHoldoff | Time Interval Stop Holdoff | P |
| **TotalizeContinuous** | | |
| Configure | Configure Continuous Totalize | M |
| Channel | Continuous Totalize Channel | P |
| Start | Start Continuous Totalize | M |
| Stop | Stop Continuous Totalize | M |
| FetchCount | Fetch Continuous Totalize Count | M |
| **TotalizeGated** | | |

| **Table 13-1.** IviCounter COM Hierarchy | | |
|---|---|---|
| **COM Interface Hierarchy** | **Generic Name** | **Type** |
| Configure | Configure Gated Totalize | M |
| Channel | Gated Totalize Channel | P |
| GateSource | Gated Totalize Gate Source | P |
| GateSlope | Gated Totalize Gate Slope | P |
| **TotalizeTimed** | | |
| Configure | Configure Timed Totalize | M |
| Channel | Timed Totalize Channel | P |
| GateTime | Timed Totalize Gate Time | P |
| **Voltage** | | |
| Configure | Configure Voltage Measurement | M |
| Channel | Voltage Channel | P |
| Estimate | Voltage Estimate | P |
| Resolution | Voltage Resolution | P |

### 13.1.1 IviCounter COM Interfaces

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

- `IIviCounterChannels`
- `IIviCounterChannel`
- `IIviCounterFrequency`
- `IIviCounterPeriod`
- `IIviCounterPulseWidth`
- `IIviCounterDutyCycle`
- `IIviCounterEdgeTime`
- `IIviCounterFrequencyRatio`
- `IIviCounterTimeInterval`
- `IIviCounterPhase`
- `IIviCounterTotalizeContinuous`
- `IIviCounterTotalizeGated`
- `IIviCounterTotalizeTimed`
- `IIviCounterArm`
- `IIviCounterMeasurement`
- `IIviCounterVoltage`

The IIviCounterArm interface contains interface reference properties for accessing the following additional IviCounter external mixer interfaces:

- `IIviCounterArmStart`
- `IIviCounterArmStop`

The IIviCounterArmStart interface contains interface reference properties for accessing the following additional IviCounter external mixer interfaces:

- `IIviCounterArmStartExternal`

The IIviCounterArmStop interface contains interface reference properties for accessing the following additional IviCounter external mixer interfaces:

- `IIviCounterArmStopExternal`

The IIviCounterChannels interface contains methods and properties for accessing a collection of objects that implement the IIviCounterChannel interface.

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

**Table 13-2.** IviCounter Interface GUIDs

| Interface | GUID |
|---|---|
| IIviCounter | { 47ed5378-a398-11d4-ba58-000064657374 } |
| IIviCounterChannels | { 47ed5379-a398-11d4-ba58-000064657374 } |
| IIviCounterChannel | { 47ed537a-a398-11d4-ba58-000064657374 } |
| IIviCounterPulseWidth | { 47ed537b-a398-11d4-ba58-000064657374 } |
| IIviCounterDutyCycle | { 47ed537c-a398-11d4-ba58-000064657374 } |
| IIviCounterEdgeTime | { 47ed537d-a398-11d4-ba58-000064657374 } |
| IIviCounterFrequencyRatio | { 47ed537e-a398-11d4-ba58-000064657374 } |
| IIviCounterTimeInterval | { 47ed537f-a398-11d4-ba58-000064657374 } |
| IIviCounterPhase | { 47ed5380-a398-11d4-ba58-000064657374 } |
| IIviCounterTotalizeTimed | { 47ed5381-a398-11d4-ba58-000064657374 } |
| IIviCounterArm | { 47ed5382-a398-11d4-ba58-000064657374 } |
| IIviCounterFrequency | { 47ed5383-a398-11d4-ba58-000064657374 } |
| IIviCounterPeriod | { 47ed5384-a398-11d4-ba58-000064657374 } |
| IIviCounterTotalizeContinuous | { 47ed5385-a398-11d4-ba58-000064657374 } |
| IIviCounterTotalizeGated | { 47ed5386-a398-11d4-ba58-000064657374 } |
| IIviCounterArmStart | { 47ed5387-a398-11d4-ba58-000064657374 } |
| IIviCounterArmStartExternal | { 47ed5388-a398-11d4-ba58-000064657374 } |
| IIviCounterArmStop | { 47ed5389-a398-11d4-ba58-000064657374 } |
| IIviCounterArmStopExternal | { 47ed538a-a398-11d4-ba58-000064657374 } |
| IIviCounterMeasurement | { 47ed538c-a398-11d4-ba58-000064657374 } |
| IIviCounterVoltage | { 47ed538d-a398-11d4-ba58-000064657374 } |

### 13.1.2 IviCounter COM Interface Reference Properties

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

| Data Type | COM Property Name |
|---|---|
| IiviCounterChannels | Channels |
| IIviCounterFrequency | Frequency |
| IIviCounterPeriod | Period |
| IIviCounterPulseWidth | PulseWidth |
| IIviCounterDutyCycle | DutyCycle |
| IIviCounterEdgeTime | EdgeTime |
| IIvicounterFrequencyRatio | FrequencyRatio |
| IIviCounterTimeInterval | TimeInterval |
| IIviCounterPhase | Phase |
| IIviCounterTotalizeContinuous | TotalizeContinuous |
| IIviCounterTotalizeGated | TotalizeGated |
| IIviCounterTotalizeTimed | TotalizeTimed |
| IIviCounterArm | Arm |
| IIviCounterArmStart | Start |
| IIviCounterArmStartExternal | External |
| IIviCounterArmStop | Stop |
| IIviCounterArmStopExternal | External |
| IIviCounterMeasurement | Measurement |
| IIviCounterVoltage | Voltage |

### 13.1.3 IviCounter COM Category

The IviCounter class COM Category shall be “IviCounter”, and the Category ID (CATID) shall be

{ 47ed515d-a398-11d4-ba58-000064657374 }.

## 13.2 IviCounter C Function Hierarchy

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

**Table 13-3.** IviCounter C Function Hierarchy

| Name or Class | Function Name |
|---|---|
| **Configuration…** | |
| Configure Channel | IviCounter_ConfigureChannel |
| Configure Level | IviCounter_ConfigureChannelLevel |
| Configure Slope | IviCounter_ConfigureChannelSlope |
| Configure Filter Enabled | IviCounter_ConfigureChannelFilterEnabled |
| Configure Frequency | IviCounter_ConfigureFrequency<br>IviCounter_ConfigureFrequencyManual |
| Configure Frequency With Aperture Time | IviCounter_ConfigureFrequencyWithApertureTime |
| Configure Period | IviCounter_ConfigurePeriod |
| Configure Period With Aperture Time | IviCounter_ConfigurePeriodWithApertureTime |
| Configure Pulse Width | IviCounter_ConfigurePulseWidth |
| Configure Duty Cycle | IviCounter_ConfigureDutyCycle |
| Configure Edge Time | IviCounter_ConfigureEdgeTime |
| Configure Edge Time Reference Levels | IviCounter_ConfigureEdgeTimeReferenceLevels |
| Configure Frequency Ratio | IviCounter_ConfigureFrequencyRatio |
| Configure Time Interval | IviCounter_ConfigureTimeInterval |
| Configure Phase | IviCounter_ConfigurePhase |
| Configure Continuous Totalize | IviCounter_ConfigureContinuousTotalize |
| Configure Gated Totalize | IviCounter_ConfigureGatedTotalize |
| Configure Timed Totalize | IviCounter_ConfigureTimedTotalize |
| Configure Start Arm | IviCounter_ConfigureStartArm |
| Configure External Start Arm | IviCounter_ConfigureExternalStartArm |
| Configure Stop Arm | IviCounter_ConfigureStopArm |
| Configure External Stop Arm | IviCounter_ConfigureExternalStopArm |
| Configure Filter | IviCounter_ConfigureFilter |
| Configure Time Interval Stop Holdoff | IviCounter_ConfigureTimeIntervalStopHoldoff |
| Configure Voltage | IviCounter_ConfigureVoltage |
| **Measurement…** | |
| Start Continuous Totalize | IviCounter_StartContinuousTotalize |
| Stop Continuous Totalize | IviCounter_StopContinuousTotalize |
| Fetch Continuous Totalize Count | IviCounter_FetchContinuousTotalizeCount |
| Read | IviCounter_Read |
| **Low Level Measurement…** | |

| **Table 13-3.** IviCounter C Function Hierarchy | |
|---|---|
| **Name or Class** | **Function Name** |
| Initiate | `IviCounter_Initiate` |
| Abort | `IviCounter_Abort` |
| Fetch | `IviCounter_Fetch` |
| IsMeasurementComplete | `IviCounter_IsMeasurementComplete` |

## 13.3 IviCounter C Attribute Hierarchy

The IviCounter class attribute hierarchy is shown in the following table.

**Table 13-4.** IviCounter C attributes Hierarchy

| Category or Generic Attribute Name | C Defined Constant |
|---|---|
| *Arming* | |
| Start Arm Type | IVICOUNTER_ATTR_START_ARM_TYPE |
| External Start Arm Source | IVICOUNTER_ATTR_EXTERNAL_START_ARM_SOURCE |
| External Start Arm Level | IVICOUNTER_ATTR_EXTERNAL_START_ARM_LEVEL |
| External Start Arm Slope | IVICOUNTER_ATTR_EXTERNAL_START_ARM_SLOPE |
| External Start Arm Delay | IVICOUNTER_ATTR_EXTERNAL_START_ARM_DELAY |
| Stop Arm Type | IVICOUNTER_ATTR_STOP_ARM_TYPE |
| External Stop Arm Source | IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SOURCE |
| External Stop Arm Level | IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_LEVEL |
| External Stop Arm Slope | IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_SLOPE |
| External Stop Arm Delay | IVICOUNTER_ATTR_EXTERNAL_STOP_ARM_DELAY |
| | |
| *Channel* | |
| Channel Count | IVICOUNTER_ATTR_CHANNEL_COUNT |
| Channel Impedance | IVICOUNTER_ATTR_IMPEDANCE |
| Channel Coupling | IVICOUNTER_ATTR_COUPLING |
| Channel Attenuation | IVICOUNTER_ATTR_ATTENUATION |
| Channel Level | IVICOUNTER_ATTR_CHANNEL_LEVEL |
| Channel Hysteresis | IVICOUNTER_ATTR_CHANNEL_HYSTERESIS |
| Channel Slope | IVICOUNTER_ATTR_CHANNEL_SLOPE |
| Channel Filter Enabled | IVICOUNTER_ATTR_FILTER_ENABLED |
| Minimum Frequency | IVICOUNTER_ATTR_FILTER_MINIMUM_FREQUENCY |
| Maximum Frequency | IVICOUNTER_ATTR_FILTER_MAXIMUM_FREQUENCY |
| | |
| *Duty Cycle Measurement* | |
| Duty Cycle Channel | IVICOUNTER_ATTR_DUTY_CYCLE_CHANNEL |
| Duty Cycle Frequency Estimate | IVICOUNTER_ATTR_DUTY_CYCLE_FREQEUNCY_ESTIMATE |
| Duty Cycle Resolution | IVICOUNTER_ATTR_DUTY_CYCLE_RESOLUTION |
| | |
| *Edge Time Measurement* | |
| Edge Time Channel | IVICOUNTER_ATTR_EDGE_TIME_CHANNEL |
| Edge Time Reference Type | IVICOUNTER_ATTR_EDGE_TIME_REFERENCE_TYPE |

| **Table 13-4.** IviCounter C attributes Hierarchy | |
|---|---|
| **Category or Generic Attribute Name** | **C Defined Constant** |
| Edge Time Estimate | IVICOUNTER_ATTR_EDGE_TIME_ESTIMATE |
| Edge Time Resolution | IVICOUNTER_ATTR_EDGE_TIME_RESOLUTION |
| Edge Time High Reference | IVICOUNTER_ATTR_EDGE_TIME_HIGH_REFERENCE |
| Edge Time Low Reference | IVICOUNTER_ATTR_EDGE_TIME_LOW_REFERENCE |
| | |
| *Frequency Measurement* | |
| Frequency Channel | IVICOUNTER_ATTR_FREQUENCY_CHANNEL |
| Frequency Estimate | IVICOUNTER_ATTR_FREQUENCY_ESTIMATE |
| Frequency Resolution | IVICOUNTER_ATTR_FREQUENCY_RESOLUTION |
| Frequency Aperture Time | IVICOUNTER_ATTR_FREQUENCY_APERTURE_TIME |
| Frequency Estimate Auto | IVICOUNTER_ATTR_FREQUENCY_ESTIMATE_AUTO |
| Frequency Resolution Auto | IVICOUNTER_ATTR_FREQUENCY_RESOLUTION_AUTO |
| | |
| *Frequency Ratio Measurement* | |
| Frequency Ratio Numerator Channel | IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_CHANNEL |
| Frequency Ratio Denominator Channel | IVICOUNTER_ATTR_FREQUENCY_RATIO_DENOMINATOR_CHANNEL |
| Frequency Ratio Numerator Frequency Estimate | IVICOUNTER_ATTR_FREQUENCY_RATIO_NUMERATOR_FREQUENCY_ESTIMATE |
| Frequency Ratio Estimate | IVICOUNTER_ATTR_FREQUENCY_RATIO_ESTIMATE |
| Frequency Ratio Resolution | IVICOUNTER_ATTR_FREQUENCY_RATIO_RESOLUTION |
| | |
| *Measurement* | |
| Measurement Function | IVICOUNTER_ATTR_MEASUREMENT_FUNCTION |
| | |
| *Period Measurement* | |
| Period Channel | IVICOUNTER_ATTR_PERIOD_CHANNEL |
| Period Estimate | IVICOUNTER_ATTR_PERIOD_ESTIMATE |
| Period Resolution | IVICOUNTER_ATTR_PERIOD_RESOLUTION |
| Period Aperture Time | IVICOUNTER_ATTR_PERIOD_APERTURE_TIME |
| | |
| *Phase Measurement* | |
| Phase Input Channel | IVICOUNTER_ATTR_PHASE_INPUT_CHANNEL |
| Phase Reference Channel | IVICOUNTER_ATTR_PHASE_REFERENCE_CHANNEL |
| Phase Frequency Estimate | IVICOUNTER_ATTR_PHASE_FREQUENCY_ESTIMATE |

| **Table 13-4.** IviCounter C attributes Hierarchy | |
|---|---|
| **Category or Generic Attribute Name** | **C Defined Constant** |
| Phase Resolution | IVICOUNTER_ATTR_PHASE_RESOLUTION |
| | |
| *Pulse Width Measurement* | |
| Pulse Width Channel | IVICOUNTER_ATTR_PULSE_WIDTH_CHANNEL |
| Pulse Width Estimate | IVICOUNTER_ATTR_PULSE_WIDTH_ESTIMATE |
| Pulse Width Resolution | IVICOUNTER_ATTR_PULSE_WIDTH_RESOLUTION |
| | |
| *Time Interval Measurement* | |
| Time Interval Start Channel | IVICOUNTER_ATTR_TIME_INTERVAL_START_CHANNEL |
| Time Interval Stop Channel | IVICOUNTER_ATTR_TIME_INTERVAL_STOP_CHANNEL |
| Time Interval Estimate | IVICOUNTER_ATTR_TIME_INTERVAL_ESTIMATE |
| Time Interval Resolution | IVICOUNTER_ATTR_TIME_INTERVAL_RESOLUTION |
| Time Interval Stop Holdoff | IVICOUNTER_ATTR_TIME_INTERVAL_STOP_HOLDOFF |
| | |
| *Totalize Measurement* | |
| Continuous Totalize Channel | IVICOUNTER_ATTR_CONTINUOUS_TOTALIZE_CHANNEL |
| Gated Totalize Channel | IVICOUNTER_ATTR_GATED_TOTALIZE_CHANNEL |
| Gated Totalize Gate Source | IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SOURCE |
| Gated Totalize Gate Slope | IVICOUNTER_ATTR_GATED_TOTALIZE_GATE_SLOPE |
| Timed Totalize Channel | IVICOUNTER_ATTR_TIMED_TOTALIZE_CHANNEL |
| Timed Totalize Gate Time | IVICOUNTER_ATTR_TIMED_TOTALIZE_GATE_TIME |
| | |
| *Voltage Measurement* | |
| Voltage Channel | IVICOUNTER_ATTR_VOLTAGE_CHANNEL |
| Voltage Estimate | IVICOUNTER_ATTR_VOLTAGE_ESTIMATE |
| Voltage Resolution | IVICOUNTER_ATTR_VOLTAGE_RESOLUTION |

## 13.4 IviCounter .NET Hierarchy

The full IviCounter .NET Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.1, *.NET Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

**Table 13-5.** IviCounter .NET Hierarchy

| .NET Interface Hierarchy | Generic Name | Type |
|---|---|---|
| MeasurementFunction | Measurement Function | P |
| **Channels** | | |
| Count | Channel Count | P |
| Name | Channel Name | P |
| **Channels[]** | | |
| Configure | Configure Channel | M |
| Impedance | Impedance | P |
| Coupling | Coupling | P |
| Attenuation | Attenuation | P |
| ConfigureLevel | Configure Level | M |
| Level | Level | P |
| Hysteresis | Hysteresis | P |
| Slope | Slope | P |
| FilterEnabled | Filter Enabled | P |
| ConfigureFilter | Configure Filter | M |
| MaximumFrequency | Maximum Frequency | P |
| MinimumFrequency | Minimum Frequency | P |
| **Frequency** | | |
| Configure | Auto Configure Frequency | M |
| ConfigureManual | Manual Configure Frequency | M |
| Channel | Frequency Channel | P |
| Estimate | Frequency Estimate | P |
| Resolution | Frequency Resolution | P |
| ConfigureWithAperture | Configure Frequency with Aperture Time | M |
| ApertureTime | Aperture Time | P |
| AutoEstimate | Auto Frequency Estimate | P |
| AutoResolution | Auto Frequency Resolution | P |
| AutoApertureTime | Auto Aperture Time | P |
| **Period** | | |
| Configure | Configure Period | M |
| Channel | Period Channel | P |
| Estimate | Period Estimate | P |

| Table 13-5. IviCounter .NET Hierarchy | | |
|---|---|---|
| **.NET Interface Hierarchy** | **Generic Name** | **Type** |
| Resolution | Period Resolution | P |
| ConfigureWithAperture | Configure Period with Aperture Time | M |
| ApertureTime | Aperture Time | P |
| **PulseWidth** | | |
| Configure | Configure Pulse Width | M |
| Channel | Pulse Width Channel | P |
| Estimate | Pulse Width Estimate | P |
| Resolution | Pulse Width Resolution | P |
| **DutyCycle** | | |
| Configure | Configure Duty Cycle | M |
| Channel | Duty Cycle Channel | P |
| FrequencyEstimate | Frequency Estimate | P |
| Resolution | Duty Cycle Resolution | P |
| **EdgeTime** | | |
| Configure | Configure Edge Time | M |
| Channel | Edge Time Channel | P |
| Estimate | Edge Time Estimate | P |
| Resolution | Edge Time Resolution | P |
| HighReference | High Reference | P |
| LowReference | Low Reference | P |
| **FrequencyRatio** | | |
| Configure | Configure Frequency Ratio | M |
| NumeratorChannel | Numerator Channel | P |
| DenominatorChannel | Denominator Channel | P |
| FrequencyEstimate | Frequency Estimate | P |
| Estimate | Frequency Ratio Estimate | P |
| Resolution | Frequency Ratio Resolution | P |
| **TimeInterval** | | |
| Configure | Configure Time Interval | M |
| StartChannel | Start Channel | P |
| StopChannel | Stop Channel | P |
| Estimate | Time Interval Estimate | P |
| Resolution | Time Interval Resolution | P |
| Delay | Time Interval Delay | P |
| **Phase** | | |
| Configure | Configure Phase | M |

| **Table 13-5.** IviCounter .NET Hierarchy | | |
|---|---|---|
| **.NET Interface Hierarchy** | **Generic Name** | **Type** |
| InputChannel | Input Channel | P |
| ReferenceChannel | Reference Channel | P |
| FrequencyEstimate | Frequency Estimate | P |
| Resolution | Phase Resolution | P |
| **TotalizeContinuous** | | |
| Configure | Configure Continuous Totalize | M |
| Channel | Continuous Totalize Channel | P |
| Start | Zero and Start Totalize | M |
| Stop | Stop Totalize | M |
| FetchValue | Get Current Value | M |
| **TotalizeGated** | | |
| Configure | Configure Gated Totalize | M |
| Channel | Gated Totalize Channel | P |
| GateSource | Gate Source | P |
| GateSlope | Gate Slope | P |
| **TotalizeTimed** | | |
| Configure | Configure Timed Totalize | M |
| Channel | Timed Totalize Channel | P |
| GateTime | Gate Time | P |
| **Arm** | | |
| **Start** | | |
| Configure | Configure Start Arm | M |
| Type | Start Arm Type | P |
| **External** | | |
| Configure | Configure External Start Arm | M |
| Source | External Start Arm Source | P |
| Level | External Start Arm Level | P |
| Slope | External Start Arm Slope | P |
| Delay | External Start Arm Delay | P |
| **Stop** | | |
| Configure | Configure Stop Arm | M |
| Type | Stop Arm Type | P |
| **External** | | |
| Configure | Configure External Stop Arm | M |
| Source | External Stop Arm Source | P |
| Level | External Stop Arm Level | P |

| Table 13-5. IviCounter .NET Hierarchy | | |
|---|---|---|
| **.NET Interface Hierarchy** | **Generic Name** | **Type** |
| Slope | External Stop Arm Slope | P |
| Delay | External Stop Arm Delay | P |
| **Measurement** | | |
| GetMeasurementComplete | Is Measurement Complete | M |
| Read | Read | M |
| Initiate | Initiate | M |
| Fetch | Fetch | M |
| Abort | Abort | M |
| **Voltage** | | |
| Configure | Configure Voltage | M |
| Channel | Voltage Channel | P |
| Estimate | Voltage Estimate | P |
| Resolution | Voltage Resolution | P |

### 13.4.1 IviCounter .NET Interface Reference Properties

Interface reference properties are used to navigate the IviCounter .NET hierarchy. This section describes the interface reference properties that the IIviCounter interface defines.

| **Data Type** | **.NET Property Name** |
|---|---|
| IIviCounterChannel | Channels[] |
| IiviCounterChannels | Channels |
| IIviCounterFrequency | Frequency |
| IIviCounterPeriod | Period |
| IIviCounterPulseWidth | PulseWidth |
| IIviCounterDutyCycle | DutyCycle |
| IIviCounterEdgeTime | EdgeTime |
| IIvicounterFrequencyRatio | FrequencyRatio |
| IIviCounterTimeInterval | TimeInterval |
| IIviCounterPhase | Phase |
| IIviCounterTotalizeContinuous | TotalizeContinuous |
| IIviCounterTotalizeGated | TotalizeGated |
| IIviCounterTotalizeTimed | TotalizeTimed |
| IIviCounterArm | Arm |
| IIviCounterArmStart | Arm.Start |
| IIviCounterArmStartExternal | Arm.Start.External |
| IIviCounterArmStop | Arm.Stop |
| IIviCounterArmStopExternal | Arm.Stop.External |
| IIviCounterMeasurement | Measurement |
| IIviCounterVoltage | Voltage |

# Appendix A Specific Driver Development Guidelines

## A.1 Introduction

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

## A.2 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>_InitWithOptions`, or `<prefix>_reset` functions execute. This assumes that the extension capability groups remain disabled until the application program explicitly uses them. For the large majority of instruments, this assumption is true.

Under certain conditions, a specific driver might have to implement a more complex approach. For some instruments, configuring a capability group might affect instrument settings that correspond to an unused extension capability group. If these instrument settings affect the behavior of the instrument, then this might result in an interchangeability problem. If this can occur, the specific driver 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 IviCounterFilter Extension Group

The IviCounterFilter extension group is disabled for a channel when the Channel Filter Enabled attribute is set to False.

### Disabling the IviCounterTimeIntervalStopHoldoff Extension Group

Setting the attribute to zero effectively disables the IviCounterTimeIntervalStopHoldoff extension group.

### Disabling the IviCounterVoltageMeasurement Extension Group

Some measurements that the user selects with the Measurement Function attribute require an extension group to further configure the measurement. The values for the Measurement Function attribute that require additional extension capability groups are shown in the following table.

- IVICOUNTER_VAL_DC_VOLTAGE
- IVICOUNTER_VAL_MAXIMUM_VOLTAGE
- IVICOUNTER_VAL_MINIMUM_VOLTAGE
- IVICOUNTER_VAL_RMS_VOLTAGE
- IVICOUNTER_VAL_PEAK_TO_PEAK_VOLTAGE

When the Measurement Function attribute is set to one of these values, the IviCounterVoltageMeasurement Extension Group affects the behavior of the instrument. Otherwise, the extension capability group does not affect the behavior of the instrument and is effectively disabled. Therefore, this section does not recommend how to disable this extension capability group.

### Disabling the IviCounterEdgeTimeReferenceLevels Extension Group

Some reference level types that the user selects with the IviCounterEdgeTimeReferenceLevels ReferenceType attribute require an extension group to further configure the measurement. The values for the ReferenceType attribute that require additional extension capability groups are shown in the following table.

- IVICOUNTER_VAL_PERCENT_REFERENCE_TYPE

When the Reference Level attribute is set to one of these values, the IviCounterEdgeTimeReferenceLevels Extension Group affects the behavior of the instrument. Otherwise, the extension capability group does not affect the behavior of the instrument and is effectively disabled. Therefore, this section does not recommend how to disable this extension capability group.

## *A.3 Query Instrument Status*

Based on the value of the Query Instrument Status attribute, a specific driver may check the status of the instrument to see if it has encountered an error. In specific driver functions, the status check should not occur in the lowest-level signal generation functions Initiate, Abort, Fetch, and Fetch Count. 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 would also add unnecessary overhead to the specific instrument driver.

# Appendix B Interchangeability Checking Guidelines

## B.1 Introduction

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

## B.2 When to Perform Interchangeability Checking

Interchangeability checking occurs when all of the following conditions are met:

- The `Interchange Check` attribute is set to True
- The user calls one of the following functions.
  - `Initiate`
  - `Read`

## B.3 Interchangeability Checking of Measurement Function Attributes

There are attributes associated with most individual function values. Interchangeability checking is only performed on the attributes that correspond to the function that is enabled.

**Table B.1** Attributes Values Enabled by the Measurement Function

| Measurement Function Attribute Setting | Attributes Enabled |
|---|---|
| **`Frequency`** | |
| | Frequency Channel |
| | Frequency Estimate |
| | Frequency Resolution |
| | Aperture Time |
| | Frequency Estimate Auto |
| | Frequency Resolution Auto |
| **`Period`** | |
| | Period Channel |
| | Period Estimate |
| | Period Resolution |
| | Aperture Time |
| **`PulseWidth`** | |
| | Pulse Width Channel |
| | Pulse Width Estimate |
| | Pulse Width Resolution |
| **`DutyCycle`** | |

**Table B.1** Attributes Values Enabled by the Measurement Function

| Measurement Function Attribute Setting | Attributes Enabled |
|---|---|
| | Duty Cycle Channel |
| | Duty Cycle Frequency Estimate |
| | Duty Cycle Resolution |
| **EdgeTime** | |
| | Edge Time Channel |
| | Edge Time Estimate |
| | Edge Time Resolution |
| | High Reference |
| | Low Reference |
| **FrequencyRatio** | |
| | Numerator Channel |
| | Denominator Channel |
| | Frequency Estimate for the Numerator Channel |
| | Frequency Ratio Estimate |
| | Frequency Ratio Resolution |
| **TimeInterval** | |
| | Start Channel |
| | Stop Channel |
| | Time Interval Estimate |
| | Time Interval Resolution |
| | Time Interval Stop Holdoff |
| **Phase** | |
| | Input Channel |
| | Reference Channel |
| | Frequency Estimate |
| | Phase Resolution |
| **TotalizeContinuous** | |
| | Continuous Totalize Channel |
| **TotalizeGated** | |
| | Gated Totalize Channel |
| | Gate Source |
| | Gate Slope |
| **TotalizeTimed** | |
| | Timed Totalize Channel |
| | Gate Time |

| **Table B.1** Attributes Values Enabled by the Measurement Function | |
|---|---|
| **Measurement Function Attribute Setting** | **Attributes Enabled** |
| **`DC Voltage, Maximum Voltage, Minimum Voltage, RMS Voltage, Peak-to-Peak Voltage`** | |
| | Voltage Channel |
| | Voltage Estimate |
| | Voltage Resolution |

## *B.4 Interchangeability Checking Rules*

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the base capability group. In addition, interchangeability checking is performed on 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.

### IviCounterFilter Extension Group

If the Filter Frequencies attributes have not been set or the Filter is disabled then the following attributes are not required to be in a user specified state:

- Minimum Frequency
- Maximum Frequency

### IviCounterTimeIntervalStopHoldoff Extension Group

If the Time Interval Stop Holdoff attribute has not been set or the Time Interval Stop Holdoff is disabled, then the following attribute is not required to be in a user specified state:

- Time Interval Stop Holdoff

### IviCounterVoltageMeasurement Extension Group

The interchangebility rules for this extension group are defined in the table in the previous section.

### IviCounterEdgeTimeReferenceLevels Extension Group

The interchangebility rules for this extension group are defined in the table in the previous section.