IVI-4.3: IviFgen Class Specification

October 14, 2016 Edition
Revision 5.2

Important Information

The IviFgen Class Specification (IVI-4.3) 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.

IviFgen Class Specification

IviFgen Revision History

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

Table 1-1. IviFgen Class Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 1.0	August 21, 1998	First Approved Version
Revision 1.1(draft)	June 25, 1999	This edition reflects issues arising from the April'99 IVI membership meeting.
Revision 2.0 (draft)	September 7, 1999	This edition refines the organization of the specification based on feedback at the July 1999 IVI Foundation meeting. Removed the Miscellaneous Extension Group Renamed the IviFgen Fundamental group to be theIviFgenBase group. Moved the SendSWTrigger function to the new IviFgenSoftwareTrigger extension group. Added the IviFgenArbFrequency, IviFgenInternalTrigger, and IviFgenBurst extension groups. Removed enumerations for floating point values Revised the IviFgenTrigger extension group. Added the OPERATION_MODE attribute to the IviFgenBase group
Revision 2.0	November 22, 1999	This edition reflects the changes made in the 2.0 draft revision, and has been approved by the IVI Foundation.
Revision 2.0a	May 22, 2001	This edition includes numerous changes to add COM API information to the spec. It also adds the C attribute hierarchy.
Revision 2.1vc1	June 29, 2001	This edition adds the standard IVI-C header file and the IVI-COM IviFgenEnglish.idl and IviFgenTypeLib.idl files. There are also several spelling, wording, and syntax corrections.
Revision 2.1vc2	September 17, 2001	This addition incorporates changes recommended during the review cycle (dates).

Table 1-1. IviFgen Class Specification Revisions

Revision 2.1vc3	October 17, 2001	This addition incorporates more changes recommended during the review cycle (dates), including removing inherent capabilities from the hierarchies.
Revision 2.1vc4	November 1, 2001	This addition incorporates changes from an Oct. 25 telephone conference and misc. email discussions
Revision 2.1vc5	December 18, 2001	Reflects all changes discussed at the December IVI Meeting.
Revision 3.0vc6	February 5, 2002	Reflects feedback from the January, 2002 review period. Based on feedback, decided that this was a major revision of the spec, and rev to 3.0 instead of 2.1.
Revision 3.0	April 22, 2002	Voted and approved version 3.0.
Revision 3.0	April 14, 2008	Editorial change to update the IVI Foundation contact information in the Important Information section to remove obsolete address information and refer only to the IVI Foundation web site.
Revision 3.0	April 2009	Editorial change to update repeated capabilities section to include both qualified and unqualified repeated capability names.
Revision 4.0	June 9, 2010	Incorporated IVI.NET
Revision 5.0	October 13, 2011	Added the following Extension Groups: IviFgenArbSeqDepth, IviFgenArbWfmSize64, IviFgenArbChannelWfm, IviFgenArbWfmBinary, IviFgenDataMarkers, IviFgenArbDataMask, IviFgenSparseMarkers, IviFgenSampleClock, IviFgenTerminalConfiguration, IviFgenStartTrigger, IviFgenStopTrigger, IviFgenHoldTrigger, IviFgenResumeTrigger, IviFgenAdvanceTrigger. Editorial IVI.NET change. Change references to process-wide locking to AppDomain-wide locking. Add an overload to the Create factory method that takes locking related parameters.
Revision 5.0	May 20, 2014	Editorial Change to update the name of `IVIFGEN_ATTR_MAX_SEQUENCE_DEPTH` to `IVIFGEN_ATTR_SEQUENCE_DEPTH_MAX` in section 33.5, and to change the sparse marker index COM type from LONG to __int64 in sections 27.3.1, 27.3.3, and 27.3.4.
Revision 5.0	May 15, 2014	Editorial change to explain that the IVI Shared Components do not support version 5 extensions for COM.
Revision 5.0	September 24, 2015	Editorial Change – Clarified the use of one-based index for C and COM, and zero-based index for .NET for repeated capabilities in sections 4.3.7, 25.2.7, and 27.2.6.

Table 1-1. IviFgen Class Specification Revisions

Revision 5.1	January 6, 2016	Minor Changes: - Updated base behavior model to not require generating an output after reset. - Modified so that arbitrary waveform handle, and arbitrary sequence handle attributes are not limited to IVI-C. Editorial Change: - Added purely editorial explanation describing how to handle gain.
Revision 5.2	February 26, 2016	Minor Changes: - When creating a waveform from an IWaveform object, require the waveform to be normalized to values between _1.0 and -1.0, and allow types other than IWaveform<Double>. - When specifically creating binary waveforms, use Int16[] and Int32[] rather than IWaveform<Int16> and IWaveform<Int32>, respectively.
Revision 5.2	May 13, 2016	Editorial Changes (.NET only): - Change Arbitrary.Sequence.Handle and Arbitrary.Waveform.Handle to Get/Set properties with channelName parameters. - Remove the Trigger.Source property. - Move Trigger.SendSoftwareTrigger() and Trigger.BurstCount to Trigger.Start. - Move Output.OperationMode and Output.ReferenceClockSource to the root Fgen interface (IIviFgen) Editorial Change (COM only) - Change Arbitrary.Sequence.Handle and Arbitrary.Waveform.Handle to have a channelName parameter.
Revision 5.2	October 14, 2016	Editorial Change – Modified header text for table 32.2 to indicate that the messages do not apply to .NET exceptions..

API Versions

Architecture	Drivers that comply with version 5.2 comply with all of the versions below
C	2.0 - 5.2
COM	5.1-5.2
.NET	5.2

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 IviFgen Specification

1.1 Introduction

This specification defines the IVI class for function generators. The IviFgen class is designed to support the typical function generator as well as common extended functionality found in more complex instruments. This section summarizes the *IviFgen 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:

- IviFgen Class Overview

- The definitions of terms and acronyms

- References

1.2 IviFgen Class Overview

This specification describes the IVI class for function generators. The IviFgen class is designed to support the typical function generator as well as common extended functionality found in more complex instruments. The IviFgen class conceptualizes a function generator as an instrument capable of generating an analog voltage waveform, and can be applied to a wide range of instruments. The output signal is typically functional in nature (for instance sinusoidal or square). Some instruments support the generation of arbitrary waveforms, which consist of user-specified data. If the function generator also supports the generation of arbitrary waveform sequences, the output signal can consist of a sequence of repeated arbitrary waveforms. The IviFgen class is divided into a base capability group and multiple extension capability groups. The base capability group configures a function generator for basic signal output. With the extensions, you can configure a function generator to generate a specific type of waveform. An instrument driver must support either the IviFgenStdFunc extension or the IviFgenArbWfm extension.

The IviFgenBase capabily group allows you to configure a function for basic signal output. This includes setting the output impedance, the operation mode, the reference clock source, and enabling or disabling output channels. You can also initiate and abort signal generation. The IviFgenBase capability group is described in Section 4, *IviFgenBase Capability Group*.

The IviFgenStdFunc extension allows you to produce standard periodic waveforms. With this extension you can configure the function generator to produce a standard waveform function, and control waveform parameters such as amplitude, offset, frequency, and start phase. This extension group is described in Section 5, *IviFgenStdFunc Extension Group*.

The IviFgenArbWfm extension allows you to generate user-defined periodic waveforms. With this extension you can configure the function generator to produce a user-specified waveform, and control arbitrary waveform parameters such as gain, offset, and sample rate. This extension group is described in Section 6, *IviFgenArbWfm Extension Group*.

In addition to the base capabilities, the IviFgen class defines extended capabilities for waveform generators that can:

- Set the frequency at which to generate the entire waveform

- Generate a sequence of arbitrary waveforms

- Use a number of different triggers with advanced settings

- Apply amplitude or frequency modulation to a signal

- Generate user-defined waveforms specified in a binary format

- Output data and sparse markers with advanced settings

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

1.3   References

The following documents and specifications are related to this specification:

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

1.4   Definitions of Terms and Acronyms

Terms of general interest are defined in *IVI-5.0: Glossary*.

This section does not define additional terms or acronyms.

2 IviFgen Class Capabilities

2.1 Introduction

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

2.2 IviFgen Group Names

The capability group names for the IviFgen 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. IviFgen Group Names

Group Name	Description
IviFgenBase	Base capabilities of the IviFgen specification. This group supports the ability to configure basic signal generation properties like output impedance and the reference clock source.
IviFgenStdFunc	Supports IviFgenBase capabilities and has the ability to generate standard periodic waveforms
IviFgenArbWfm	Supports IviFgenBase capabilities and has the ability to create and generate user-defined arbitrary waveforms.
IviFgenArbChannelWfm	Supports the IviFgenArbWfm extension and has the ability to create and generate user-defined arbitrary waveforms for specific channels.
IviFgenArbWfmBinary	Supports the IviFgenArbWfm extension and has the ability to create and generate user-defined arbitrary waveforms for specific channels in binary (16-bit or 32-bit integer) format.
IviFgenDataMarkers	Supports the IviFgenArbWfmBinary extension and has the ability to output particular bits of the waveform data as marker outputs.
IviFgenSparseMarkers	Supports the IviFgenArbWfm extension and has the ability to output markers synchronous with specified locations in waveforms.
IviFgenArbDataMask	Supports the IviFgenArbWfmBinary extension and has the ability to mask out bits of the output data.
IviFgenArbFrequency	Supports the IviFgenArbWfm extension and has the ability to specify the rate at which the function generator produces one cycle of an arbitrary waveform on a channel.
IviFgenArbSeq	Supports the IviFgenArbWfm extension and has the ability to create and generate sequences of user-defined arbitrary waveforms.

Table 2-1. IviFgen Group Names

`IviFgenArbSeqDepth`	Supports the IviFgenArbSeq extension and has the ability to create and generate sequences of sequences of user-defined arbitrary waveforms.
`IviFgenTrigger`	Supports IviFgenBase capabilities and has the ability to specify the source of trigger signals. This capability has been deprecated by the IviFgenStartTrigger extension.
`IviFgenInternalTrigger`	Supports the IviFgenTrigger capabilities and has the ability to generate triggers internally.
`IviFgenSoftwareTrigger`	Supports the IviFgenTrigger capabilities and has the ability to receive software triggers.
`IviFgenBurst`	Supports IviFgenBase capabilities and has the ability to generate discrete numbers of waveform cycles based on a trigger event.
`IviFgenModulateAM`	Supports IviFgenBase capabilities and has the ability to apply amplitude modulation to an output signal.
`IviFgenModulateFM`	Supports IviFgenBase capabilities and has the ability to apply frequency modulation to an output signal.
`IviFgenSampleClock`	Supports IviFgenBase capabilities and has the ability to select between the internal sample clock or an external clock for the generation of the output signal, and optionally to output the sample clock via a dedicated connector.
`IviFgenTerminalConfiguration`	Supports IviFgenBase capabilities and has the ability to specify whether the output terminals are single-ended or differential.
`IviFgenStartTrigger`	Supports the IviFgenBase extension and has the ability to specify the source and properties of start trigger signals.
`IviFgenStopTrigger`	Supports the IviFgenBase extension and has the ability to specify the source and properties of stop trigger signals.
`IviFgenHoldTrigger`	Supports the IviFgenBase extension and has the ability to specify the source and properties of hold trigger signals.
`IviFgenResumeTrigger`	Supports the IviFgenBase extension and has the ability to specify the source and properties of resume trigger signals.
`IviFgenAdvanceTrigger`	Supports the IviFgenArbWfm extension and has the ability to specify the source and properties of advance trigger signals.

2.3 Repeated Capability Names

The IviFgen Class Specification defines three repeated capabilities. 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*, and Section 5.9, *Repeated Capability Identifiers and Selectors*.

- Channel

- DataMarker

- SparseMarker

2.3.1 Channel

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

Note that some Fgen properties are channel dependent. In COM, these properties are implemented as properties which take a single "Channel Name" parameter. These kinds of properties are also called "parameterized properties".   In .NET, these properties are implemented as a pair of methods named Set<property name> and Get<property name>, where <property name> is the name of the corresponding COM property[1]. In this specification, the .NET Set<property name> and Get<property name> methods are documented in the "Attribute" section where the corresponding IVI-C/IVI-COM attribute/property is documented, to emphasize the fact that they represent the same property in the .NET API.

2.3.2 DataMarker

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

The DataMarker capability is used by the IviFgenDataMarker extension and will only be available if that extension is implemented.

2.3.3 SparseMarker

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

The SparseMarker capability is used by the IviFgenSparseMarker extension and will only be available if that extension is implemented.

---

[1] This is due to the fact that properties in CLS-compliant .NET may not take parameters.

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 IviFgen class is `Ivi.Fgen`.

2.6 .NET IviFgen Session Factory

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

```
IIviFgen Ivi.Fgen.Create(String name);

IIviFgen Ivi.Fgen.Create(String name,
              Boolean idQuery,
              Boolean reset);

IIviFgen Ivi.Fgen.Create(String name,
              Boolean idQuery,
              Boolean reset,
              String options);

IIviFgen Ivi.Fgen.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 IviFgen 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 pointer to the IIviFgen interface of the driver referenced by session.	IIviFgen

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 IviFgen instrument class API from the logical name "My LogicalName" use the following code:

```
IIviFgen fgen = IviFgen.Create("MyLogicalName");
```

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

3 General Requirements

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

3.1 Minimum Class Compliance

To be compliant with the IviFgen Class Specification, an IVI specific driver shall conform to all of the requirements for an IVI class-compliant specific driver specified in *IVI-3.1: Driver Architecture Specification*. In addition it shall implement the inherent capabilities that *IVI- 3.2: Inherent Capabilities Specification* defines, the IviFgenBase capability group, and at least one of the following capability extension groups:

- IviFgenStdFunc capability extension group

- IviFgenArbWfm capability extension group

3.1.1 Disable

Refer to *IVI-3.2: Inherent Capabilities Specification* for the prototype of this function.

The Disable function shall cause the Function Generator to apply the minimum amount of power possible at the output terminals. Setting the voltage to a value close to zero or physically disconnecting the function generator from the output terminals meets this requirement. Other techniques are also allowed.

3.2 Capability Group Compliance

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

3.3 IviFgen Version 5.0 Compliance and Compatibility

To accommodate arbitrary waveform generators with significant new functionality, the IviFgen class specification version 5.0 (or higher) adds a number of new extensions, methods, and properties. These changes were made with consistency and backwards compatibility as priorities. However, driver writers should be aware that there are some specific requirements when porting a driver to IviFgen class specification version 5.0 from previous versions of the class specification. This section highlights the new additions to version 5.0 and seeks to assist driver writers with the new requirements.

3.3.1 New Extensions

Several extensions are new to version 5.0 of the IviFgen specification. It is generally optional for drivers to implement these extensions when migrating to version 5.0. If a driver implements similar functionality to that of an extension, it is strongly recommended that the driver implement the extension. New extensions include IviFgenArbChannelWfm, IviFgenArbWfmBinary, IviFgenArbWfmSize64, IviFgenDataMarker, IviFgenArbDataMask, IviFgenSparseMarker, IviFgenSampleClock, IviFgenTerminalConfiguration, IviFgenStartTrigger, IviFgenStopTrigger, IviFgenHoldTrigger, IviFgenResumeTrigger, IviFgenAdvanceTrigger, and IviFgenArbSeqDepth.

3.3.2 New Repeated Capabilities

Several repeated capabilities are new to version 5.0 of the IviFgen class specification. These include DataMarker, and SparseMarker. These repeated capabilities are required only if the driver

implements the associated extensions. Refer to Section 2.3, *Repeated Capability Names* for more details.

3.3.3 New Create Waveform Functions

The IviFgenArbChannelWfm extension is new to version 5.0 of the IviFgen class specification. It allows the channel on which the waveform is generated to be specified at waveform creation. This potentially enables the driver to optimize waveform transfer to the arbitrary waveform generator. Arbitrary waveform generators that support the IviFgenArbChannelWfm extension must also support the IviFgenArbWfm extension. For more details, refer to Section 23, *IviFgenArbChannelWfm Extension Group*.

When implementing IviFgenArbWfm for an instrument that requires the channel to be known in order to download the waveform to the device, the driver must make other accommodations, such as storing the waveform in memory or on disk until Configure Waveform is called.

3.3.4 Waveform Sequence Changes

Version 5.0 adds the IviFgenArbSeqDepth extension so that more complex waveform sequences, i.e., sequences of sequences, can be supported. Drivers that allow sequences of sequences must implement the Max Sequence Depth attribute and return a value of 2 or greater. Note that additional requirements apply to drivers that support sequences of sequences, such as ensuring waveform handles and sequence handles are unique. These requirements are outlined in more detail in Section 28, *IviFgenArbSeqDepth Extension Group*.

3.3.5 IviFgenStartTrigger Extension

Prior to version 5.0, the IviFgen class specification defined a single trigger for starting a generation. Version 5.0 adds a number of new trigger types and expands the available options on triggers. This includes an IviFgenStartTrigger extension that deprecates the existing IviFgenTrigger extension. To ensure that applications based on previous versions of the IviFgen class specification continue to work with version 5.0, the class specification requires drivers that implement triggers to support the IviFgenTrigger extension. In other words, if a driver supports the IviFgenStartTrigger extension, it must also support the IviFgenTrigger extension. The converse is also true – if a driver compliant with version 5.0 supports the IviFgenTrigger extension, it must also support the IviFgenStartTrigger extension. For IVI-C drivers, this is accomplished by supporting both extensions. For IVI-COM drivers, this is accomplished by exporting two different top-level COM interface hierarchies. Refer to Section 3.3.6, *IVI-C Changes in Version 5.0* and Section 3.3.7, *IVI-COM Changes in Version 5.0* for more details.

When the initial IviFgen class specification was written, trigger sources were represented using enumeration values. The IVI Foundation now recommends that newer specifications use the string data type for trigger sources, as strings provide more vendor and user flexibility. Version 5.0 of the IviFgen class specification also adds a number of new trigger types. This includes the new IviFgenStartTrigger extension. Consistent with the IVI Foundation's recommendations, the sources for the new trigger types are specified using strings.

3.3.6 IVI-C Changes in Version 5.0

IVI-C drivers upgrading to version 5.0 of the IviFgen class specification must make the following changes:
1. If the driver supports sequences of sequences (Max Sequence Depth ≥2), it must ensure that waveform and sequence handles are unique.
2. If the driver supports the IviFgenTrigger extension, it must add support for the IviFgenStartTrigger extension and continue to support both extensions.

The IVI Shared Components currently support version 5.0 extensions for C.

3.3.7 IVI-COM Changes in Version 5.0

Version 5.0 introduces a new COM interface – IIviFgen2. The IIviFgen2 interface is an additional top-level interface that version 5.0 drivers will have to support when it becomes available. A driver compliant with version 5.0 of the IviFgen class specification must implement both IIviFgen and IIviFgen2 class interfaces on the main driver coclass. This will allow applications built with previous versions of the specification to access the old IIviFgen interface.

IIviFgen2 removes the Trigger Source property and adds the new properties and methods associated with version 5.0 of the class specification. The IIviFgen interface remains unchanged from previous versions of the class specification.

3.3.8 IVI.NET

Because the .NET API was first introduced with version 5.0, there are no compatibility issues with .NET.

The IVI.NET Shared Components currently support version 5.0 extensions.

4 IviFgenBase Capability Group

4.1 IviFgenBase Overview

The IviFgenBase capability group supports the most basic function generator capabilities. The user can configure the output impedance and reference clock source, and enable or disable the function generator's output channels.

This specification uses the following terms to describe the function generator's output stage: Output Impedance, System Impedance, and Output Voltage. These properties are defined in the following list:

Output Impedance – The impedance of the function generator at the output terminal. The user configures the output impedance with the Output Impedance attribute.

System Impedance – The impedance of the system attached to the output terminal.

Preload Voltage – The voltage the function generator generates internally.

Output Voltage – The voltage of the waveform at the output terminal.

These properties are shown in the following illustration.
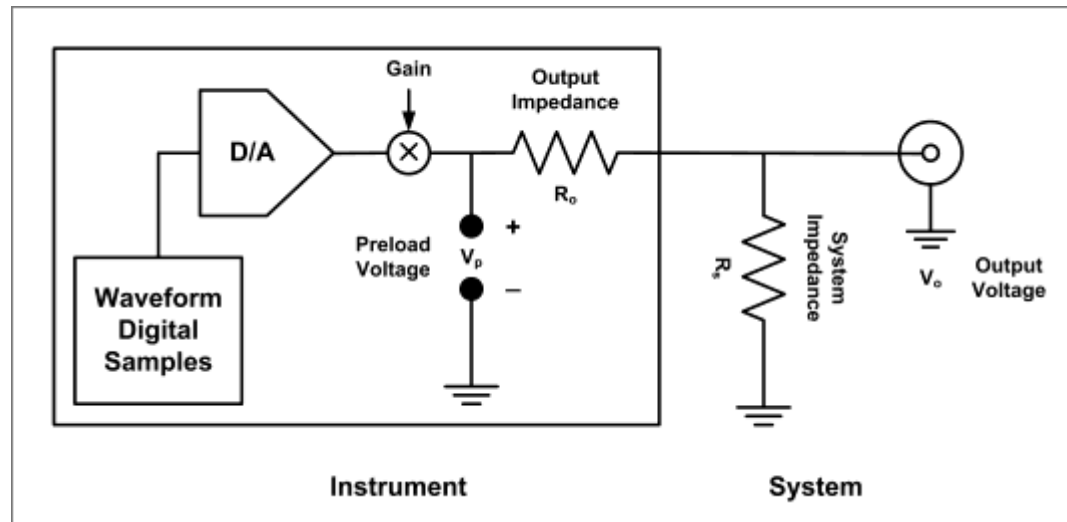
Figure 4-1. Illustration of Output Path Properties

The output impedance and system impedance form a voltage divider, which causes the output voltage to be less than the preload voltage. The relationship between the output voltage and the preload voltage is governed by the following equation,

$V_o = V_pR_s/(R_o + R_s)$

When the output impedance is equal to the system impedance, the equation reduces to

$V_o = V_p/2$.

When the output impedance is very small compared to the system impedance, the equation reduces to

$V_o = V_p$.

When the output impedance is a non-zero value, the instrument assumes that the output impedance and system impedance are equal, and produces a preload voltage that is twice the user-requested value. This causes the output voltage to be equal to the user-requested value.

When the output impedance is 0.0, the instrument assumes that the system impedance is very large, and produces a preload voltage that is equal to the user-requested value. This causes the output voltage to be equal to the user-requested value.

Note: *The IviFgenBase capability group does not support the generation of a specific kind of output signal. Instead, a function generator must support either the IviFgenStdFunc or IviFgenArbWfm Extension Groups. This organization is required because many function generators support only one of these extension groups. If a function generator supports more than one of these extensions, the output mode can be configured to specify which extension the function generator uses to produce a signal.*

4.2 IviFgenBase Attributes

The IviFgenBase capability group defines the following attributes:

- Output Count

- Operation Mode

- Output Enabled

- Output Impedance

- Output Mode

- Output Name (IVI-COM Only)

- Reference Clock Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

4.2.1 Output Count

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	None

.NET Property Name

> Output.Count

COM Property Name

> Output.Count

C Constant Name

> IVIFGEN_ATTR_CHANNEL_COUNT

Description

> Returns the number of available output 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.

4.2.2 Operation Mode

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Operation Mode

.NET Property Using Channel-Dependent Get and Set Methods

```
Ivi.Fgen.OperationMode GetOperationMode(String channelName)

void SetOperationMode(String channelName,
                           OperationMode mode)
```

.NET Enumeration Name

```
Ivi.Fgen.OperationMode
```

COM Property Name

```
Output.OperationMode(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenOperationModeEnum
```

C Constant Name

```
IVIFGEN_ATTR_OPERATION_MODE
```

Description

Specifies how the function generator produces output on a channel.

Defined Values

Name	Description	
	Language	*Identifier*
Operate Continuous	When in the Output Generation State, the function generator produces output continuously.	
	.NET	OperationMode.Continuous
	C	IVIFGEN_VAL_OPERATE_CONTINUOUS
	COM	IviFgenOperationModeContinuous
Operate Burst	When in the Output Generation State, the function generator produces a burst of waveform cycles based on a trigger condition. A burst consists of a discrete number of waveform cycles. The user uses the attribute of the IviFgenTrigger Extension Group to configure the trigger, and the attributes of the IviFgenBurst extension group to configure how the function generator produces bursts.	
	.NET	OperationMode.Burst
	C	IVIFGEN_VAL_OPERATE_BURST

	COM	IviFgenOperationModeBurst

.NET Exceptions

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

Compliance Notes

1. Specific drivers shall implement the Operate Continuous value.

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

Name	Required Capability Group
Operate Burst	IviFgenBurst, IviFgenTrigger

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

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

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

Refer to Section 16, Attribute Value Definitions, for the definitions of Operation Mode Specific Ext Base, IVIFGEN_VAL_OP_MODE_SPECIFIC_EXT_BASE and IVIFGEN_VAL_OP_MODE_CLASS_EXT_BASE.

4.2.3 Output Enabled

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViBoolean	R/W	Channel	None	Configure Output Enabled

.NET Property Using Channel-Dependent Get and Set methods

```
Boolean Output.GetEnabled(String channelName)

void Output.SetEnabled(String channelName, Boolean enabled)
```

COM Property Name

```
Output.Enabled(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_OUTPUT_ENABLED
```

Description

If set to True, the signal the function generator produces appears at the output connector. If set to False, the signal the function generator produces does not appear at the output connector.

.NET Exceptions

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

Compliance Notes

1. Specific drivers shall support True on all channels.

4.2.4 Output Impedance

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Output Impedance

.NET Property Using Channel-Dependent Get and Set methods

```
Double Output.GetImpedance (String channelName)

void Output.SetImpedance (String channelName, Double impedance)
```

COM Property Name

```
Output.Impedance(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_OUTPUT_IMPEDANCE
```

Description

Specifies the impedance of the output channel. The units are Ohms.

.NET Exceptions

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

4.2.5 Output Mode

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	N/A	None	Configure Output Mode

.NET Property Name

> Output.OutputMode

.NET EnumerationName

> Ivi.Fgen.OutputMode

COM Property Name

> Output.OutputMode

COM Enumeration Name

> IviFgenOutputModeEnum

C Constant Name

> IVIFGEN_ATTR_OUTPUT_MODE

Description

> Determines how the function generator produces waveforms. This attribute determines which
> extension group's functions and attributes are used to configure the waveform the function
> generator produces.

Defined Values

Name	Description	
	Language	Identifier
Output Function	The driver uses the attributes and functions of the IviFgenStdFunc extension group to configure the function generator's output signal.	
	.NET	OutputMode.Function
	C	IVIFGEN_VAL_OUTPUT_FUNC
	COM	IviFgenOutputModeFunction
Output Arbitrary	The driver uses the attributes and functions of the IviFgenArbWfm extension group to configure the function generator's output signal.	
	.NET	OutputMode.Arbitrary
	C	IVIFGEN_VAL_OUTPUT_ARB
	COM	IviFgenOutputModeArbitrary

Output Sequence	The driver uses the attributes and functions of the IviFgenArbSeq extension group to configure the function generator's output signal.		
		.NET	`Ivi.Fgen.OutputMode.Sequence`
		C	`IVIFGEN_VAL_OUTPUT_SEQ`
		COM	`IviFgenOutputModeSequence`

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

Name	Required Capability Group
Output Function	IviFgenStdFunc
Output Arbitrary	IviFgenArbWfm
Output Sequence	IviFgenArbSeq

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

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

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

Refer to Section 16, Attribute Value Definitions, for the definitions of Output Mode Specific Ext Base, `IVIFGEN_VAL_OUT_MODE_SPECIFIC_EXT_BASE` and `IVIFGEN_VAL_OUT_MODE_CLASS_EXT_BASE`.

4.2.6 Output Name (IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViChar[]	RO	N/A	None	GetChannelName (C Only)

.NET Prototype

    N/A

COM Property Name

    Output.Name([in] LONG Index);

C Prototype

    N/A

Description

This property returns the physical name defined by the specific driver for the output channel that corresponds to the one-based index that the user specifies. If the driver defines a qualified channel name, this property returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of Output Count, the property returns an empty string and returns an error.

4.2.7 Reference Clock Source

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32 (C/COM) ViString (.NET)	R/W	N/A	None	Configure Reference Clock Source

.NET Property Name

>   ReferenceClockSource

COM Property Name

>   Output.ReferenceClockSource

COM Enumeration Name

>   IviFgenReferenceClockSourceEnum

C Constant Name

>   IVIFGEN_ATTR_REF_CLOCK_SOURCE

Description

> Specifies the source of the reference clock. The function generator derives frequencies and sample rates that it uses to generate waveforms from the reference clock.
>
> In IVI.NET the source of the reference clock is a string. If an IVI driver supports a reference clock source and the reference clock source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3, then the IVI driver shall accept the standard string for that reference clock. This attribute is case insensitive, but case preserving. That is, the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new reference clock source strings for reference clock sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Defined Values

Name	Description	
	Language	Identifier
Reference Clock Internal	The function generator produces the reference clock signal internally.	
	C	IVIFGEN_VAL_REF_CLOCK_INTERNAL
	COM	IviFgenReferenceClockSourceInternal
Reference Clock External	The function generator receives the reference clock signal from an external source.	
	C	IVIFGEN_VAL_REF_CLOCK_EXTERNAL
	COM	IviFgenReferenceClockSourceExternal

Reference Clock RTSI Clock	The function generator receives the reference clock signal from the RTSI clock source.	
	C	IVIFGEN_VAL_REF_CLOCK_RTSI_CLOCK
	COM	IviFgenReferenceClockSourceRTSI

.NET 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 specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIFGEN_VAL_CLK_SRC_SPECIFIC_EXT_BASE.

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

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

Refer to Section 16, Attribute Value Definitions, for the definitions of Reference Clock Specific Ext Base, IVIFGEN_VAL_CLK_SRC_SPECIFIC_EXT_BASE and IVIFGEN_VAL_CLK_SRC_CLASS_EXT_BASE.

4.3   IviFgenBase Functions

In addition to the IVI inherent functions, the IviFgenBase capability group includes the following functions:

- Abort Generation

- Configure Operation Mode (IVI-C Only)

- Configure Output Enabled (IVI-C Only)

- Configure Output Impedance (IVI-C Only)

- Configure Output Mode (IVI-C Only)

- Configure Reference Clock Source (IVI-C Only)

- Get Channel Name (IVI-C Only)

- Initiate Generation

This section describes the behavior and requirements of each function.

4.3.1 Abort Generation

Description

Aborts a previously initiated signal generation. If the function generator is in the Output
Generation State, this function moves the function generator to the Configuration State. If the
function generator is already in the Configuration State, the function does nothing and returns
Success.

This specification requires that the user be able to configure the output of the function generator
regardless of whether the function generator is in the Configuration State or the Generation State.
This means that the user is not required to call Abort Generation prior to configuring the output of
the function generator.

Many function generators constantly generate an output signal, and do not require the user to abort
signal generation prior to configuring the instrument. If a function generator's output cannot be
aborted (i.e., the function generator cannot stop generating a signal) this function does nothing and
returns Success.

Some function generators require that the user abort signal generation prior to configuring the
instrument . The specific drivers for these types of instruments must compensate for this
restriction and allow the user to configure the instrument without requiring the user to call Abort
Generation. For these types of instruments, there is often a significant performance increase if the
user configures the output while the instrument is not generating a signal.

The user is not required to call Abort Generation or Initiate Generation. Whether the user chooses
to call these functions in an application program has no impact on interchangeability. The user
can choose to use these functions if they want to optimize their application for instruments that
exhibit increased performance when output configuration is performed while the instrument is not
generating a signal.

.NET Method Prototype

```
void AbortGeneration();
```

COM Method Prototype

```
HRESULT AbortGeneration();
```

C Prototype

```
ViStatus IviFgen_AbortGeneration (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Return Values (C/COM)

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

.NET Exceptions

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

4.3.2 Configure Operation Mode (IVI-C Only)

Description

Configures how the function generator produces output on a channel.

.NET Method Prototype

```
N/A
```

(Use `SetOperationMode`. See the Operation Mode attribute section.)

COM Method Prototype

```
N/A
```

(Use the `Output.OperationMode` property.)

C Prototype

```
ViStatus IviFgen_ConfigureOperationMode (ViSession Vi,
                                         ViConstString ChannelName,
                                         ViInt32 OperationMode);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the channel on which to configure the operation mode.	ViConstString
OperationMode	Specifies how the function generator produces output. The driver uses this value to set the Operation Mode attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

4.3.3 Configure Output Enabled (IVI-C Only)

Description

Configures whether the signal the function generator produces appears at a channel's output connector.

.NET Method Prototype

```
N/A
```

(Use `Output.SetEnabled`. See the Output Enabled attribute section.)

COM Method Prototype

```
N/A
```

(Use the `Output.Enabled` property.)

C Prototype

```
ViStatus IviFgen_ConfigureOutputEnabled (ViSession Vi,
                                         ViConstString ChannelName,
                                         ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the channel to enable or disable.	ViConstString
Enabled	Specifies whether the signal the function generator produces appears at the channel's output connector. The driver uses this value to set the Output Enabled Attribute. See the attribute description for more details.	ViBoolean

Return Values (C)

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

4.3.4 Configure Output Impedance (IVI-C Only)

Description

Sets the output terminal impedance of one of the function generator's channels.

.NET Method Prototype

```
N/A
```

(Use `Output.SetImpedance. See the Output Impedance attribute section.`)

COM Method Prototype

```
N/A
```

(Use the `Output.Impedance` property)

C Prototype

```
ViStatus IviFgen_ConfigureOutputImpedance (ViSession Vi,
                                           ViConstString ChannelName,
                                           ViReal64 Impedance);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the channel on which to configure the output impedance.	ViConstString
Impedance	Specifies the output terminal impedance. The driver uses this value to set the Output Impedance Attribute. See the attribute description for more details.	ViReal64

Return Values (C)

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

4.3.5 Configure Output Mode (IVI-C Only)

Description

Configures the output mode of the function generator. The output mode determines how the function generator produces waveforms.

.NET Method Prototype

```
N/A
```

(Use the `Output.OutputMode` property)

COM Method Prototype

```
N/A
```

(Use the `Output.OutputMode` property)

C Prototype

```
ViStatus IviFgen_ConfigureOutputMode (ViSession Vi,
                                      ViInt32 OutputMode);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
OutputMode	Specifies the output mode. The driver uses this value to set the Output Mode attribute. See the attribute description for more details.	ViConstString

Return Values (C)

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

4.3.6 Configure Reference Clock Source (IVI-C Only)

Description

Sets the source of the function generator's reference clock. The function generator uses the reference clock to derive frequencies and sample rates when generating output.

.NET Method Prototype

```
N/A
```

(Use `SetReferenceClockSource`. See the Reference Clock Source attribute section.)

COM Method Prototype

```
N/A
```

(Use the `Output.ReferenceClockSource` property)

C Prototype

```
ViStatus IviFgen_ConfigureRefClockSource (ViSession Vi,
                                          ViInt32 Source);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Source	Specifies the reference clock source. The driver uses this value to set the Reference Clock Source attribute. See the attribute description for more details.	ViConstString

Return Values (C)

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

4.3.7 Get Channel Name (IVI-C and IVI.NET Only)

Description

This function returns the physical name defined by the specific driver for the output channel that corresponds to the index that the user specifies. If the driver defines a qualified channel name, this property returns the qualified name. If the value that the user passes for the ChannelIndex parameter is less than one or greater than the value of the Channel Count, the function returns an empty string in the ChannelName parameter and returns an error.

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

For COM, Name is considered as a read-only property. Its IDL attributes include `propget`.

.NET Method Prototype

```
String Output.GetChannelName (Int32 index);
```

COM Method Prototype

Use the Output.Name property.

C Prototype

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

Parameters

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

Outputs	Description	Base Type
Name (C/COM)	A user-allocated buffer into which the driver stores the channel name. The caller may pass VI_NULL for this parameter if the name buffer size parameter is 0.	ViChar []
Return value (NET)	The channel name.	ViString

Return Values (C)

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

.NET Exceptions

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

4.3.8 Initiate Generation

Description

Initiates signal generation. If the function generator is in the Configuration State, this function moves the function generator to the Output Generation State. If the function generator is already in the Output Generation State, this function does nothing and returns Success.

This specification requires that the instrument be in the Generation State after the user calls the Initialize or Reset functions. This specification also requires that the user be able to configure the output of the function generator regardless of whether the function generator is in the Configuration State or the Generation State. This means that the user is only required to call Initiate Generation if they abort signal generation by calling Abort Generation.

Many function generators constantly generate an output signal, and do not require the user to initiate signal generation. If a function generator is always outputting the currently configured signal, this function does nothing and returns Success.

Some function generators require that the user abort signal generation prior to configuring the instrument and initiate signal generation after configuring the instrument. The specific drivers for these types of instruments must compensate for this restriction and allow the user to configure the instrument without requiring the user to call Abort Generation and Initiate Generation. For these types of instruments, there is often a significant performance increase if the user configures the output while the instrument is not generating a signal.

The user is not required to call Initiate Generation and Abort Generation. Whether the user chooses to call these functions in an application program has no impact on interchangeability. The user can choose to use these functions if they want to optimize their application for instruments that exhibit increased performance when output configuration is performed while the instrument is not generating a signal.

.NET Method Prototype

```
void InitiateGeneration();
```

COM Method Prototype

```
HRESULT InitiateGeneration();
```

C Prototype

```
ViStatus IviFgen_InitiateGeneration (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Return Values (C/COM)

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

.NET Exceptions

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

4.4 IviFgenBase Behavior Model

The following behavior model shows the relationship between the IviFgenBase capability group and function generator behavior.
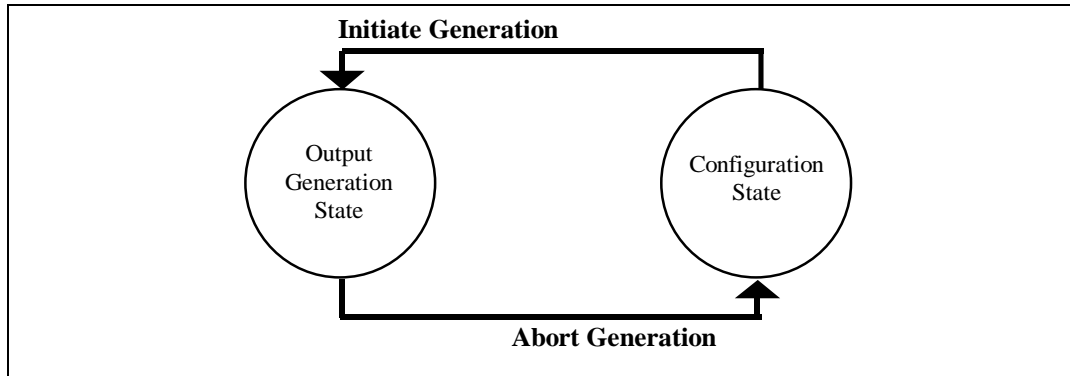
Figure 4-2. IviFgenBase Behavior Model

After the user calls the Initialize or Reset functions, the function generator enters either the Configuration State or the Output Generation State. Drivers are encouraged to go to the Configuration State in order to provide the application with explicit control of when an output is generated.

Calling Abort Generation moves the function generator to the Configuration State. Depending on the specific function generator, this might or might not cause signal generation to cease.

Note: *Any changes made to the instrument while in the Configuration State shall take place no later than when the user calls Initiate Generation, but can occur before then.*

Calling Initiate Generation moves the function generator back to the Output Generation State. The output signal then reflects all configuration changes made to the function generator while in the Configuration State.

Note: *Any changes made to the instrument while in the Output Generation State take place immediately.*

When the function generator is in the Output Generation State, it generates output based on the current value of the Operation Mode attribute. If the Operation Mode attribute is set to Operate Continuous, the function generator produces the waveform continuously.

If the Operation Mode attribute is set to a value other than Operate Continuous, the attributes and functions of a corresponding extension group control the operation of the function generator. Refer to the Behavior Model section of the corresponding extension group for details regarding instrument behavior.

This specification does not define when the function generator changes to a new operation mode if the value of the Operation Mode attribute changes while the function generator is in the Output Generation State.

5 IviFgenStdFunc Extension Group

5.1 IviFgenStdFunc Overview

The IviFgenStdFunc Extension Group supports function generators that can produce manufacturer-supplied periodic waveforms. The user can modify properties of the waveform such as frequency, amplitude, DC offset, and phase offset.

This extension affects instrument behavior when the Output Mode attribute is set to Output Function.

Instrument vendors typically have different definitions for the waveform properties. In order to achieve a consistent waveform description between different instrument vendors, this specification provides waveform property definitions that must be followed when developing instrument drivers. The definitions for these waveform properties are given in the following list:

Standard Waveform – The overall "shape" of one period of the standard waveform. This specification defines six waveform types: *Sine, Square, Triangle, Ramp Up, Ramp Down,* and *DC*. The following figure illustrates these waveform types:
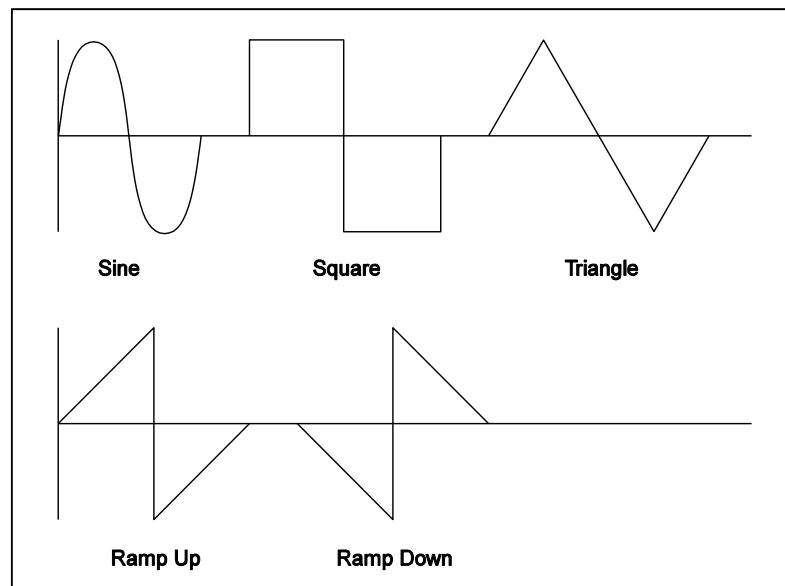
Figure 5-1. Diagrams of Standard Waveforms

Amplitude – The difference between the maximum and minimum waveform values, or the peak-to-peak voltage value.

DC Offset – The difference between the average of the maximum and minimum waveform values and the x-axis (0 volts). A positive DC offset places the middle of the waveform above the x-axis, while a negative DC offset places the middle of the waveform below the x-axis.

Frequency – The number of waveform cycles generated in one second.

Start Phase – Specifies the waveform's horizontal offset. The units are degrees of one waveform cycle. For example, a 180 degree phase offset means output generation begins half way through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle. It is therefore identical to a start phase of 0 degrees.

These properties are shown in the following illustration, which displays a phase offset of 315 degrees. Note that this is equivalent to a phase offset of –45 degrees.
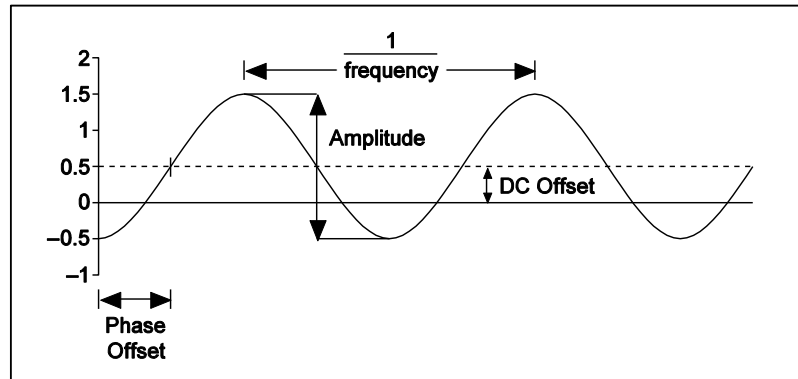
Figure 5-2. Illustration of Basic Waveform Properties

Duty Cycle – A square waveform requires an additional parameter to configure the duty cycle of the waveform. Duty cycle is defined as the percentage of time during one cycle for which the square wave is at its high value.

The following figure illustrates this concept.
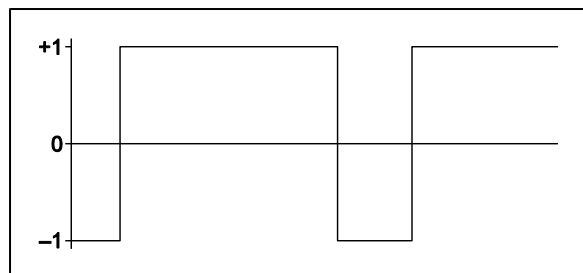
Figure 5-3. Square Waveform with 75 percent Duty Cycle

5.2 IviFgenStdFunc Attributes

The IviFgenStdFunc capability group defines the following attributes:

- Amplitude

- DC Offset

- Duty Cycle High

- Frequency

- Start Phase

- Waveform

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

5.2.1 Amplitude

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Standard Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double StandardWaveform.GetAmplitude(String channelName)

void StandardWaveform.SetAmplitude(String channelName, Double amplitude)
```

COM Property Name

```
StandardWaveform.Amplitude(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_AMPLITUDE
```

Description

Specifies the amplitude of the standard waveform the function generator produces. When the Waveform attribute is set to Waveform DC, this attribute does not affect signal output. The units are volts.

.NET Exceptions

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

5.2.2 DC Offset

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Standard Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double StandardWaveform.GetDCOffset (String channelName)

void StandardWaveform.SetDCOffset (String channelName, Double dcOffset)
```

COM Property Name

```
StandardWaveform.DCOffset(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_DC_OFFSET
```

Description

Specifies the DC offset of the standard waveform the function generator produces. If the Waveform attribute is set to Waveform DC, this attribute specifies the DC level the function generator produces. The units are volts.

.NET Exceptions

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

5.2.3 Duty Cycle High

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	None

.NET Property Using Channel-Dependent Get and Set methods

```
Double StandardWaveform.GetDutyCycleHigh (String channelName)

void StandardWaveform.SetDutyCycleHigh (String channelName,
                                        Double dutyCycleHigh)
```

COM Property Name

```
StandardWaveform.DutyCycleHigh(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_DUTY_CYCLE_HIGH
```

Description

Specifies the duty cycle for a square waveform. This attribute affects function generator behavior only when the Waveform attribute is set to Waveform Square. The value is expressed as a percentage.

Compliance Notes

1. If a specific driver does not support the Waveform Square value for the Waveform attribute, it need not implement the the Duty Cycle High 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 these methods.

5.2.4 Frequency

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Standard Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double StandardWaveform.GetFrequency (String channelName)

void StandardWaveform.SetFrequency (String channelName, Double frequency)
```

COM Property Name

```
StandardWaveform.Frequency(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_FREQUENCY
```

Description

Specifies the frequency of the standard waveform the function generator produces. When the Waveform attribute is set to Waveform DC, this attribute does not affect signal output. The units are Hertz.

.NET Exceptions

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

5.2.5 Start Phase

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Standard Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double StandardWaveform.GetStartPhase (String channelName)

void StandardWaveform.SetStartPhase (String channelName,
                                    Double startPhase)
```

COM Property Name

```
StandardWaveform.StartPhase(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_START_PHASE
```

Description

Specifies the start phase of the standard waveform the function generator produces. When the Waveform attribute is set to Waveform DC, this attribute does not affect signal output. The units are degrees.

.NET Exceptions

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

5.2.6 Waveform

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Standard Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Ivi.Fgen.StandardWaveform StandardWaveform.GetWaveformFunction
                             (String channelName)

void StandardWaveform.SetWaveformFunction
                             (String channelName,
                             Ivi.Fgen.StandardWaveform waveform)
```

.NET Enumeration Name

```
Ivi.Fgen.StandardWaveform
```

COM Property Name

```
StandardWaveform.Waveform(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenWaveformEnum
```

C Constant Name

```
IVIFGEN_ATTR_FUNC_WAVEFORM
```

Description

Specifies which standard waveform the function generator produces.

Defined Values

Name	Description	
	Language	*Identifier*
Waveform Sine	Configures the function generator to produce a sinusoid waveform.	
	.NET	StandardWaveform.Sine
	C	IVIFGEN_VAL_WFM_SINE
	COM	IviFgenWaveformSine
Waveform Square	Configures the function generator to produce a square waveform.	
	.NET	StandardWaveform.Square
	C	IVIFGEN_VAL_WFM_SQUARE
	COM	IviFgenWaveformSquare

Waveform Triangle	Configures the function generator to produce a triangular waveform.	
	.NET	StandardWaveform.Triangle
	C	IVIFGEN_VAL_WFM_TRIANGLE
	COM	IviFgenWaveformTriangle
Waveform Ramp Up	Configures the function generator to produce a positive ramp waveform.	
	.NET	StandardWaveform.RampUp
	C	IVIFGEN_VAL_WFM_RAMP_UP
	COM	IviFgenWaveformRampUp
Waveform Ramp Down	Configures the function generator to produce a negative ramp waveform.	
	.NET	StandardWaveform.RampDown
	C	IVIFGEN_VAL_WFM_RAMP_DOWN
	COM	IviFgenWaveformRampDown
Waveform DC	Configures the function generator to produce a constant voltage.	
	.NET	StandardWaveform.DC
	C	IVIFGEN_VAL_WFM_DC
	COM	IviFgenWaveformDC

.NET Exceptions

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

Compliance Notes

1. If a specific driver implements the defined value Waveform Square, the driver shall also implement the Duty Cycle High attribute.

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

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

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

Refer to Section 28, Attribute Value Definitions, for the definitions of Waveform Specific Ext Base, IVIFGEN_VAL_WFM_SPECIFIC_EXT_BASE and IVIFGEN_VAL_WFM_CLASS_EXT_BASE.

5.3 IviFgenStdFunc Functions

The IviFgenStdFunc capability group defines the following functions:

- Configure Standard Waveform

This section describes the behavior and requirements of each function.

5.3.1 Configure Standard Waveform

Description

This function configures the attributes of the function generator that affect standard waveform generation. These attributes are the Waveform, Amplitude, DC Offset, Frequency, and Start Phase.

When the Waveform parameter is set to Waveform DC, this function ignores the Amplitude, Frequency, and Start Phase parameters and does not set the Amplitude, Frequency, and Start Phase attributes.

.NET Method Prototype

```
void StandardWaveform.Configure (String channelName,
                                 StandardWaveformFunction waveformFunction,
                                 Double amplitude,
                                 Double dcOffset,
                                 Double frequency,
                                 Double startPhase);
```

COM Method Prototype

```
HRESULT StandardWaveform.Configure ([in] BSTR ChannelName,
                                    [in] IviFgenWaveformEnum Waveform,
                                    [in] DOUBLE Amplitude,
                                    [in] DOUBLE DCOffset,
                                    [in] DOUBLE Frequency,
                                    [in] DOUBLE StartPhase);
```

C Prototype

```
ViStatus IviFgen_ConfigureStandardWaveform (ViSession Vi,
                                            ViConstString ChannelName,
                                            ViInt32 Waveform,
                                            ViReal64 Amplitude,
                                            ViReal64 DCOffset,
                                            ViReal64 Frequency,
                                            ViReal64 StartPhase);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure a standard waveform.	ViConstString
Waveform (C/COM) waveformFunction (.NET)	Specifies the standard waveform. The driver uses this value to set the Waveform attribute. See the attribute description for more details.	ViInt32
Amplitude	Specifies the waveform amplitude. The driver uses this value to set the Amplitude attribute. See the attribute description for more details.	ViReal64

`DCOffset`	Specifies the waveform's DC offset. The driver uses this value to set the DC Offset attribute. See the attribute description for more details.	`ViReal64`
`Frequency`	Specifies the waveform frequency. The driver uses this value to set the Frequency attribute. See the attribute description for more details.	`ViReal64`
`StartPhase`	Specifies the waveform start phase. The driver uses this value to set the Start Phase attribute. See the attribute description for more details.	`ViReal64`

Return Values (C/COM)

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

.NET Exceptions

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

5.4   IviFgenStdFunc Behavior Model

The IviFgenStdFunc Extension Group follows the behavior model of the IviFgenBase capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenStdFunc settings.

5.5   IviFgenStdFunc Compliance Notes

1. If a specific driver implements the IviFgenStdFunc Extension Group, it shall implement the value Output Function for the Output Mode attribute.

2. If a specific driver does not support the Waveform Square value for the Waveform attribute, it need not implement the the Duty Cycle High attribute.

6 IviFgenArbWfm Extension Group

6.1 IviFgenArbWfm Overview

The IviFgenArbWfm Extension Group supports function generators capable of producing user-defined arbitrary waveforms. The user can modify parameters of the arbitrary waveform such as sample rate, waveform gain, and waveform offset. The IviFgenArbWfm extension group includes functions for creating, configuring, and generating arbitrary waveforms, and for returning information about arbitrary waveform creation.This extension affects instrument behavior when the Output Mode attribute is set to Output Arbitrary or Output Sequence.

Before a function generator can produce an arbitrary waveform, the user must configure some signal generation properties. This specification provides definitions for arbitrary waveform properties that must be followed when developing instrument drivers. The definition of an arbitrary waveform and its properties are given in the following list:

Arbitrary Waveform – A user-defined series of sequential data points, between −1.0 and 1.0 inclusive, that describe an output waveform.

Gain – The factor by which the function generator scales the arbitrary waveform data. For example, a gain value of 2.0 causes the waveform data to range from −2.0V to +2.0V.

Offset – The value the function generator adds to the scaled arbitrary waveform data. For example, scaled arbitrary waveform data that ranges from −1.0V to +1.0V is generated from 0.0V to 2.0V when the end user specifies a waveform offset of 1.0V.

☞ **Note:** *The offset is added to any inherent offset in the arbitrary waveform data.*

The following figure illustrates the definitions for arbitrary waveform properties.
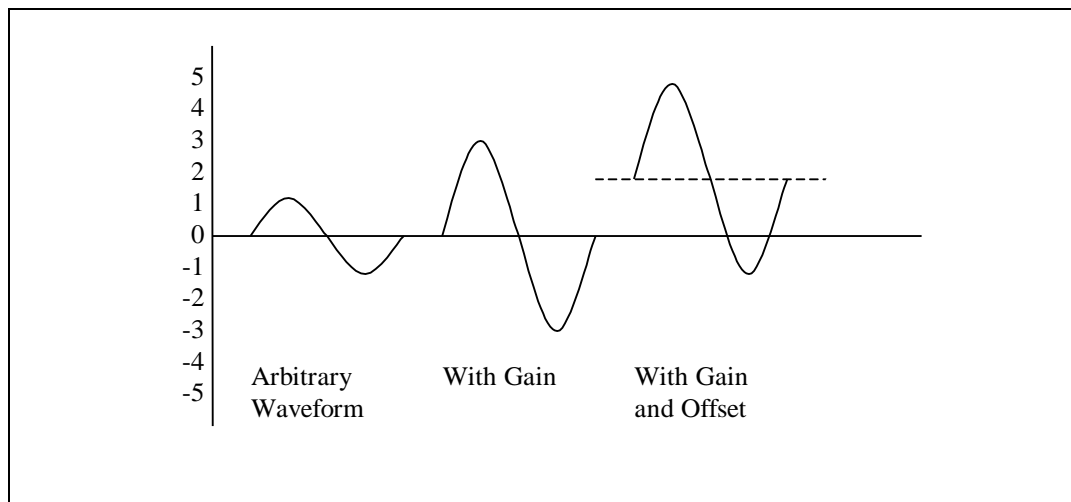
Figure 6-1. Examples of Arbitrary Waveforms

Sample Rate – The rate at which the function generator produces the points in an arbitrary waveform.

The following figure illustrates the definition for sample rate.
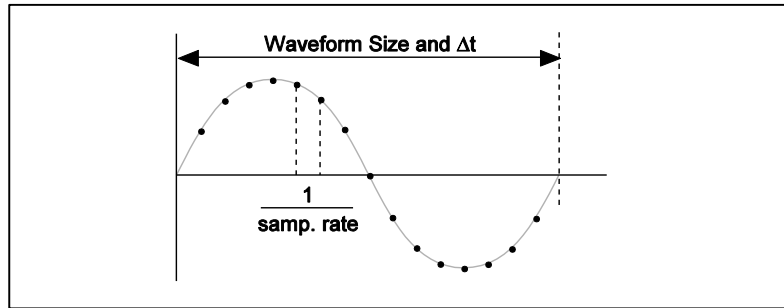
Figure 6-2. Size, Sample Rate, and Time Elements of a Waveform

The sample rate is the reciprocal of the amount of time for which points in the arbitrary waveform is generated. The frequency at which the function generator produces one cycle of an arbitrary waveform can be expressed by the equation:

```
Frequency = SampleRate/NumPoints
```

6.2   IviFgenArbWfm Attributes

The IviFgenArbWfm capability group defines the following attributes:

- Arbitrary Gain

- Arbitrary Offset

- Arbitrary Sample Rate

- Arbitrary Waveform Handle

- Number Waveforms Max

- Waveform Size Max

- Waveform Size Min

- Waveform Quantum

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

6.2.1 Arbitrary Gain

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Arbitrary Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double Arbitrary.GetGain (String channelName)

void Arbitrary.SetGain (String channelName, Double gain)
```

COM Property Name

```
Arbitrary.Gain(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ARB_GAIN
```

Description

Specifies the gain of the arbitrary waveform the function generator produces. This value is unitless.

For waveforms that are created using normalized floating point numbers (that is, those created using CreateChannelWaveform or CreateWaveform), the output voltage of the generator is the gain times the sample plus the offset.

For waveforms that are created using integers (that is, those created used Create, CreateChannelWaveformInt16, or CreateChannelWaveformInt32), the output voltage of the generator is:

$$Output\ Voltage = Offset + \frac{Sample * Gain}{2^{(SampleBitResolution - 1)} - 1}$$

.NET Exceptions

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

6.2.2 Arbitrary Offset

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Arbitrary Waveform

.NET Property Using Channel-Dependent Get and Set methods

```
Double Arbitrary.GetOffset (String channelName)

void Arbitrary.SetOffset (String channelName, Double offset)
```

COM Property Name

```
Arbitrary.Offset(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ARB_OFFSET
```

Description

Specifies the offset of the arbitrary waveform the function generator produces. The units are volts.

.NET Exceptions

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

6.2.3 Arbitrary Sample Rate

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N/A	None	Configure Sample Rate

.NET Property Name

> Arbitrary.SampleRate

COM Property Name

> Arbitrary.SampleRate

C Constant Name

> IVIFGEN_ATTR_ARB_SAMPLE_RATE

Description

> Specifies the sample rate of the arbitrary waveforms the function generator produces. The units are samples per second.

Compliance Notes

1. If the user has set the Arbitrary Frequency attribute in the IviFgenArbFrequency extension group such that the sample rates of multiple channels are not the same, querying this attribute may return an error code.

2. If the driver supports the IviFgenSampleClock extension group and the source is external, the user sets this property to specify the frequency of the external sample clock.

.NET 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.

6.2.4 Arbitrary Waveform Handle

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Arbitrary Waveform

.NET Property Name Using Channel-Dependent Get and Set methods

```
Int32 Arbitrary.Waveform.GetHandle (String channelName)

void Arbitrary.Waveform.SetHandle (String channelName,
                                   Int32 handle)
```

COM Property Name

```
Arbitrary.Waveform.Handle(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ARB_WAVEFORM_HANDLE
```

Description

Identifies which arbitrary waveform the function generator produces. You create arbitrary
waveforms with the Create Arbitrary Waveform function. This function returns a handle that
identifies the particular waveform. To configure the function generator to produce a specific
waveform, set this attribute to the waveform's handle.

Compliance Notes

1. This attribute shall accept only values that the Create Arbitrary Waveform, Create Channel
   Arbitrary Waveform, Create Channel Arbitrary Waveform Int16, and Create Channel
   Arbitrary Waveform Int32 functions returns.

6.2.5 Number Waveforms Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	Query Arbitrary Waveform Capabilities

.NET Property Name

> Arbitrary.Waveform.NumberWaveformsMax

COM Property Name

> Arbitrary.Waveform.NumberWaveformsMax

C Constant Name

> IVIFGEN_ATTR_MAX_NUM_WAVEFORMS

Description

> Returns the maximum number of arbitrary waveforms that the function generator allows.

.NET 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.

6.2.6 Waveform Size Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32 (C/COM)	RO	N/A	None	Query Arbitrary Waveform Capabilities
Int64 (.NET)	RO	N/A	None	N/A

.NET Property Name

> Arbitrary.Waveform.SizeMax

COM Property Name

> Arbitrary.Waveform.SizeMax

C Constant Name

> IVIFGEN_ATTR_MAX_WAVEFORM_SIZE

Description

> Returns the maximum number of points the function generator allows in an arbitrary waveform.

.NET 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.

6.2.7 Waveform Size Min

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32 (C/COM)	RO	N/A	None	Query Arbitrary Waveform Capabilities
Int64 (.NET)	RO	N/A	None	N/A

.NET Property Name

>   Arbitrary.Waveform.SizeMin

COM Property Name

>   Arbitrary.Waveform.SizeMin

C Constant Name

>   IVIFGEN_ATTR_MIN_WAVEFORM_SIZE

Description

>   Returns the minimum number of points the function generator allows in an arbitrary waveform.

.NET 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.

6.2.8 Waveform Quantum

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	Query Arbitrary Waveform Capabilities

.NET Property Name

> Arbitrary.Waveform.Quantum

COM Property Name

> Arbitrary.Waveform.Quantum

C Constant Name

> IVIFGEN_ATTR_WAVEFORM_QUANTUM

Description

> The size of each arbitrary waveform shall be a multiple of a quantum value. This attribute returns the quantum value the function generator allows. For example, if this attribute returns a value of 8, all waveform sizes must be a multiple of 8.

.NET 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 IviFgenArbWfm Functions

The IviFgenArbWfm extension defines the following functions:

- Clear Arbitrary Waveform

- Configure Arbitrary Waveform

- Configure Sample Rate (IVI-C Only)

- Create Arbitrary Waveform

- Query Arbitrary Waveform Capabilities (IVI-C Only)

This section describes the behavior and requirements of each function.

6.3.1 Clear Arbitrary Waveform

Description

Removes a previously created arbitrary waveform from the function generator's memory and invalidates the waveform's handle.

If the waveform cannot be cleared because it is currently being generated, or it is specified as part of an existing arbitrary waveform sequence, this function returns the Waveform In Use error.

.NET Method Prototype

```
void Arbitrary.Waveform.Clear (Int32 handle);
```

COM Method Prototype

```
HRESULT Arbitrary.Waveform.Clear ([in] LONG Handle);
```

C Prototype

```
ViStatus IviFgen_ClearArbWaveform (ViSession Vi,
                                   ViInt32 Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Handle	Specifies the handle that identifies the arbitrary waveform to clear. A value of All Waveforms causes the function generator to clear all user-defined waveforms from its memory. Defined Values: All Waveforms	ViInt32

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
Waveform In Use	The function generator is currently configured to produce the specified waveform or the waveform is part of an existing sequence.

.NET Exceptions

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

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

Exception Class	Description
WaveformInUseException	The function generator is currently configured to produce the specified waveform or the waveform is part of an existing sequence.

Compliance Notes

1. Specific drivers shall implement the All Waveforms value for the Handle parameter.

6.3.2 Configure Arbitrary Waveform

Description

Configures the attributes of the function generator that affect arbitrary waveform generation. These attributes are the arbitrary waveform handle, gain, and offset.

.NET Method Prototype

```
void Arbitrary.Waveform.Configure (String channelName,
                                   Int32 handle,
                                   Double gain,
                                   Double offset);
```

COM Method Prototype

```
HRESULT Arbitrary.Waveform.Configure ([in] BSTR ChannelName,
                                      [in] LONG Handle,
                                      [in] DOUBLE Gain,
                                      [in] DOUBLE Offset);
```

C Prototype

```
ViStatus IviFgen_ConfigureArbWaveform (ViSession Vi,
                                        ViConstString ChannelName,
                                        ViInt32 Handle,
                                        ViReal64 Gain,
                                        ViReal64 Offset);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the arbitrary waveform.	ViConstString
Handle	Specifies the handle that identifies the arbitrary waveform to produce. The driver uses this value to set the Arbitrary Waveform Handle attribute. See the attribute description for more details.	ViInt32
Gain	Specifies the arbitrary waveform gain. The driver uses this value to set the Arbitrary Gain attribute. See the attribute description for more details.	ViReal64
Offset	Specifies the arbitrary waveform offset. The driver uses this value to set the Arbitrary Offset attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

Completion Codes	Description
Invalid Waveform Channel	The waveform was created on a different channel than the one for which it is being configured.

.NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown,
and warning events that may be raised, by this method.
The table below specifies additional class-defined exceptions for this method.

Exception Class	Description
InvalidWaveformChannelException	The waveform was created on a different channel than the one for which it is being configured.

6.3.3 Configure Sample Rate (IVI-C Only)

Description

Configures the function generator's sample rate.

.NET Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.SampleRate` property.)

COM Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.SampleRate` property.)

C Prototype

```
ViStatus IviFgen_ConfigureSampleRate (ViSession Vi,
                                      ViReal64 SampleRate);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
SampleRate	Specifies the sample rate. The driver uses this value to set the Arbitrary Sample Rate attribute. See the attribute description for more details.	ViReal64

Return Values (C)

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

6.3.4 Create Arbitrary Waveform (IVI-C and IVI-COM Only)

Description

Creates an arbitrary waveform and returns a handle that identifies that waveform. You pass a waveform handle as the `waveformHandle` parameter of the Configure Arbitrary Waveform function to produce that waveform. You also use the handles this function returns to create a sequence of arbitrary waveforms with the Create Arbitrary Sequence function.

If the function generator cannot store any more arbitrary waveforms, this function returns the error No Waveforms Available.

For IVI.NET, the functions related to arbitrary waveform creation are included in the IviFgenArbChannelWfm and IviFgenArbWfmBinary extension groups.

.NET Method Prototype

N/A

(Use the `Arbitrary.Waveform.CreateChannelWaveform` methods.)

COM Method Prototype

```
HRESULT Arbitrary.Waveform.Create ([in] SAFEARRAY(double) *Data,
                                   [out,retval] LONG *Handle);
```

C Prototype

```
ViStatus IviFgen_CreateArbWaveform (ViSession Vi,
                                    ViInt32 Size,
                                    ViReal64 Data[],
                                    ViInt32 *Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Size	Specifies the number of points in the Data array.	ViInt32
Data	Specifies the array of data to use for the new arbitrary waveform. The array must have at least as many elements as the value in the Size parameter. The array's elements must be normalized between -1.00 and +1.00.	ViReal64[]

Outputs	Description	Base Type
Handle (C/ COM)	Returns the handle that identifies the new arbitrary waveform.	ViInt32
Return Value (NET)	Returns the handle that identifies the new arbitrary waveform.	ViInt32

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
No Waveforms Available	The function generator's arbitrary waveform memory is full

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method. The *IVI-3-18: IVI.NET Utility Classes and Interfaces Specification* defines additional waveform-related exceptions that may be thrown by this method.

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

Exception Class	Description
NoWaveformsAvailable	The function generator's arbitrary waveform memory is full

6.3.5 Query Arbitrary Waveform Capabilities (IVI-C Only)

Description

Returns the attributes of the function generator that are related to creating arbitrary waveforms. These attributes are the maximum number of waveforms, waveform quantum, minimum waveform size, and maximum waveform size.

.NET Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.NumberWaveformsMax`, `Quantum`, `SizeMin`, and `SizeMax` properties.)

COM Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.NumberWaveformsMax`, `Quantum`, `SizeMin`, and `SizeMax` properties.)

C Prototype

```
ViStatus IviFgen_QueryArbWfmCapabilities (ViSession Vi,
                                          ViInt32 *MaxNumWfms,
                                          ViInt32 *WfmQuantum,
                                          ViInt32 *MinWfmSize,
                                          ViInt32 *MaxWfmSize);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Outputs	Description	Base Type
MaxNumWfms	Returns the maximum number of arbitrary waveforms that the function generator allows. The driver obtains this value from the Number Waveforms Max attribute. See the attribute description for more details.	ViInt32
WfmQuantum	Returns the quantum value the function generator uses. The driver returns this value from the Waveform Quantum attribute. See the attribute description for more details.	ViInt32
MinWfmSize	Returns the minimum number of points the function generator allows in a waveform. The driver obtains this value from the Waveform Size Min attribute. See the attribute description for more details.	ViInt32
MaxWfmSize	Returns the maximum number of points the function generator allows in a waveform. The driver obtains this value from the Waveform Size Max attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

6.4 IviFgenArbWfm Behavior Model

The IviFgenArbWfm Extension Group follows the behavior model of the IviFgenBase capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenArbWfm settings.

6.5 IviFgenArbWfm Compliance Notes

1. If a specific driver implements the IviFgenArbWfm Extension Group, it shall implement the value Output Arbitrary for the Output Mode attribute.

7 IviFgenArbFrequency Extension Group

7.1 IviFgenArbFrequency Overview

The IviFgenArbFrequency extension group supports function generators capable of producing arbitrary waveforms that allow the user to set the rate at which an entire waveform buffer is generated. In order to support this extension, a driver must first support the IviFgenArbWfm extension group. This extension uses the IviFgenArbWfm extension group's attributes of Arbitrary Waveform Handle, Arbitrary Gain, and Arbitrary Offset to configure an arbitrary waveform.

This extension affects instrument behavior when the Output Mode attribute is set to Output Arbitrary.

7.2 IviFgenArbFrequency Attributes

The IviFgenArbFrequency capability group defines the following attribute:

- Arbitrary Frequency

This section describes the behavior and requirements of the attribute. The actual value for the attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

7.2.1 Arbitrary Frequency

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	Channel	None	Configure Arb Frequency

.NET Property Using Channel-Dependent Get and Set methods

```
Double Arbitrary.Waveform.GetFrequency (String channelName)

void Arbitrary.Waveform.SetFrequency (String channelName,
                                      Double frequency)
```

COM Property Name

```
Arbitrary.Waveform.Frequency(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ARB_FREQUENCY
```

Description

Specifies the rate in Hertz at which an entire arbitrary waveform is generated.

.NET Exceptions

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

7.3 IviFgenArbFrequency Functions

The IviFgenArbFrequency extension defines the following functions:

- Configure Arbitrary Frequency (IVI-C Only)

This section describes the behavior and requirements of each function.

7.3.1 Configure Arbitrary Frequency (IVI-C Only)

Description

Configures the rate at which the function generator produces an entire arbitrary waveform.

.NET Method Prototype

```
N/A
```

(Use Arbitrary.Waveform.SetFrequency. See the Arbitrary Frequency attribute section.)

COM Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.Frequency` property.)

C Prototype

```
ViStatus IviFgen_ConfigureArbFrequency (ViSession Vi,
                                        ViConstString ChannelName,
                                        ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the arbitrary waveform frequency.	ViConstString
Frequency	Specifies the rate at which an entire arbitrary waveform is generated. The driver uses this value to set the Arbitrary Frequency attribute. See the attribute description for more details.	ViReal64

Return Values (C)

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

7.4 IviFgenArbFrequency Behavior Model

The IviFgenArbFrequency Extension Group follows the behavior model of the IviFgenArbWfm capability group. The only modification to the behavior model from the IviFgenArbWfm capability group is the ability to configure IviFgenArbFrequency settings.

7.5 IviFgenArbFrequency Compliance Notes

1. If a specific driver implements the IviFgenArbFrequency Extension Group, it shall also implement the IviFgenArbWfm Extension Group.

8 IviFgenArbSeq Extension Group

8.1 IviFgenArbSeq Overview

The IviFgenArbSeq extension group supports function generators capable of producing sequences of arbitrary waveforms. In order to support this extension, a driver must first support the IviFgenArbWfm extension group. This extension uses the IviFgenArbWfm extension group's attributes of sample rate, gain, and offset to configure a sequence. The IviFgenArbSeq extension group includes functions for creating, configuring, and generating sequences, and for returning information about arbitrary sequence creation.

This extension affects instrument behavior when the Output Mode attribute is set to Output Sequence.

This specification defines an arbitrary sequence as a list of arbitrary waveforms to produce. Each waveform in the sequence is repeated a discrete number of times before producing the next waveform. When generating an arbitrary sequence, the waveform properties of *Gain, Offset* and *Sample Rate* defined in Section 6.1*, IviFgenArbWfm Overview* apply to all waveforms in the sequence.

The following figure illustrates the definition of an arbitrary sequence.
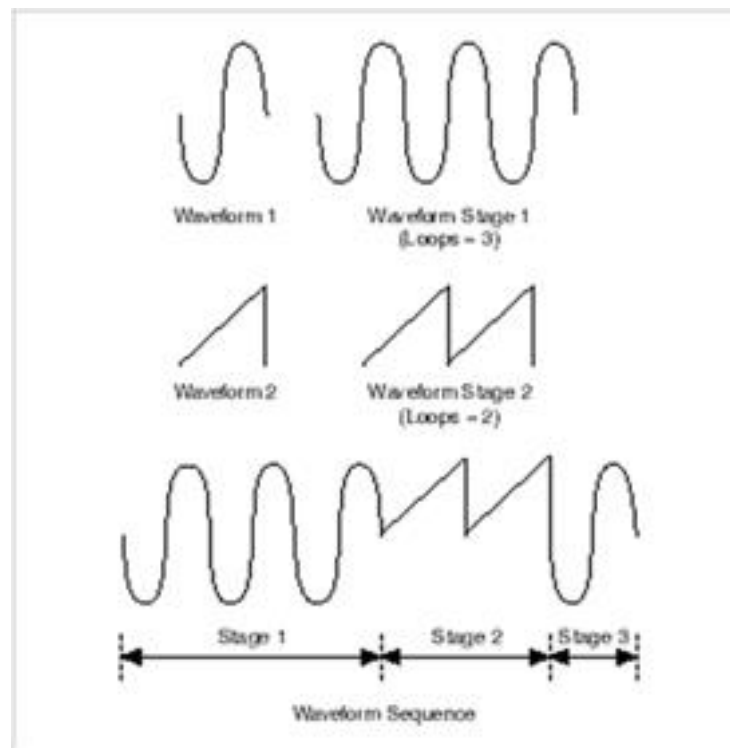
Figure 8-1. Waveform Sequencing

An arbitrary waveform sequence is constructed by specifying the waveforms to generate and the number of cycles of each waveform to generate. In the example diagram above, a sinusoid waveform is generated for three cycles, and a positive ramp waveform is generated for two cycles. After the last positive ramp waveform cycle is generated, the sequence is repeated.

8.2   IviFgenArbSeqAttributes

The IviFgenArbSeqAttributes capability group defines the following attributes:

- Arbitrary Sequence Handle

- Number Sequences Max

- Loop Count Max

- Sequence Length Max

- Sequence Length Min

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

8.2.1 Arbitrary Sequence Handle

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Arbitrary Sequence

.NET Property Name Using Channel-Dependent Get and Set methods

```
Int32 Arbitrary.Sequence.GetHandle (String channelName)

void Arbitrary.Sequence.SetHandle (String channelName,
                                   Int32 handle)
```

COM Property Name

```
Arbitrary.Sequence.Handle(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ARB_SEQUENCE_HANDLE
```

Description

Identifies which arbitrary sequence the function generator produces. You create arbitrary sequences with the Create Arbitrary Sequence function. This function returns a handle that identifies the particular sequence. To configure the function generator to produce a specific sequence, set this attribute to the sequence's handle.

Compliance Notes

1. This attribute shall accept only values that have been returned from the Create Arbitrary Sequence function.

8.2.2 Number Sequences Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	Query Arbitrary Sequence Capabilities

.NET Property Name

Arbitrary.Sequence.NumberSequencesMax

COM Property Name

Arbitrary.Sequence.NumberSequencesMax

C Constant Name

IVIFGEN_ATTR_MAX_NUM_SEQUENCES

Description

Returns the maximum number of arbitrary sequences that the function generator allows.

.NET Exceptions

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

8.2.3 Loop Count Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	Query Arbitrary Sequence Capabilities

.NET Property Name

Arbitrary.Sequence.LoopCountMax

COM Property Name

Arbitrary.Sequence.LoopCountMax

C Constant Name

IVIFGEN_ATTR_MAX_LOOP_COUNT

Description

Returns the maximum number of times that the function generator can repeat a waveform in a sequence.

.NET Exceptions

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

8.2.4 Sequence Length Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N A	None	Query Arbitrary Sequence Capabilities

.NET Property Name

> Arbitrary.Sequence.LengthMax

COM Property Name

> Arbitrary.Sequence.LengthMax

C Constant Name

> IVIFGEN_ATTR_MAX_SEQUENCE_LENGTH

Description

> Returns the maximum number of arbitrary waveforms that the function generator allows in an
> arbitrary sequence.

.NET Exceptions

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

8.2.5 Sequence Length Min

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N A	None	Query Arbitrary Sequence Capabilities

.NET Property Name

      Arbitrary.Sequence.LengthMin

COM Property Name

      Arbitrary.Sequence.LengthMin

C Constant Name

      IVIFGEN_ATTR_MIN_SEQUENCE_LENGTH

Description

Returns the minimum number of arbitrary waveforms that the function generator allows in an arbitrary sequence.

.NET Exceptions

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

8.3 IviFgenArbSeq Functions

The IviFgenArbSeq extension defines the following functions:

- Clear Arbitrary Memory

- Clear Arbitrary Sequence

- Configure Arbitrary Sequence

- Create Arbitrary Sequence

- Query Arbitrary Sequence Capabilities (IVI-C Only)

This section describes the behavior and requirements of each function.

8.3.1 Clear Arbitrary Memory

Description

Removes all previously created arbitrary waveforms and sequences from the function generator's memory and invalidates all waveform and sequence handles.

If a waveform cannot be cleared because it is currently being generated, this function returns the error Waveform In Use.

If a sequence cannot be cleared because it is currently being generated, this function returns the error Sequence In Use.

.NET Method Prototype

```
void Arbitrary.ClearMemory ();
```

COM Method Prototype

```
HRESULT Arbitrary.ClearMemory ();
```

C Prototype

```
ViStatus IviFgen_ClearArbMemory (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Return Values (C/COM)

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

Completion Codes	Description
Waveform In Use	The function generator is currently configured to produce the specified waveform.
Sequence In Use	The function generator is currently configured to produce the specified sequence.

.NET Exceptions

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

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

Exception Class	Description
WaveformInUseException	The function generator is currently configured to produce the specified waveform.
SequenceInUseException	The function generator is currently configured to produce the specified sequence.

8.3.2 Clear Arbitrary Sequence

Description

Removes a previously created arbitrary sequence from the function generator's memory and invalidates the sequence's handle.

If the sequence cannot be cleared because it is currently being generated, this function returns the error Sequence In Use.

.NET Method Prototype

```
void Arbitrary.Sequence.Clear (Int32 handle);
```

COM Method Prototype

```
HRESULT Arbitrary.Sequence.Clear ([in] LONG Handle);
```

C Prototype

```
ViStatus IviFgen_ClearArbSequence (ViSession Vi,
                                   ViInt32 Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Handle	Specifies the handle that identifies the arbitrary sequence to clear. A value of All Sequences causes the function generator to clear all user-defined waveforms from its memory. Defined Values: All Sequences	ViInt32

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
Sequence In Use	The function generator is currently configured to produce the specified sequence.

.NET Exceptions

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

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

Exception Class	Description
SequenceInUseException	The function generator is currently configured to produce the specified sequence.

Compliance Notes

1. Specific drivers shall implement the All Sequences value.

8.3.3 Configure Arbitrary Sequence

Description

Configures the attributes of the function generator that affect arbitrary sequence generation. These attributes are the arbitrary sequence handle, gain, and offset.

.NET Method Prototype

```
void Arbitrary.Sequence.Configure (String channelName,
                                   Int32 handle,
                                   Double gain,
                                   Double offset);
```

COM Method Prototype

```
HRESULT Arbitrary.Sequence.Configure ([in] BSTR ChannelName,
                                      [in] LONG Handle,
                                      [in] DOUBLE Gain,
                                      [in] DOUBLE Offset);
```

C Prototype

```
ViStatus IviFgen_ConfigureArbSequence (ViSession Vi,
                                        ViConstString ChannelName,
                                        ViInt32 Handle,
                                        ViReal64 Gain,
                                        ViReal64 Offset);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the arbitrary waveform.	ViConstString
Handle	Specifies the handle that identifies the arbitrary sequence to produce. The driver uses this value to set the Arbitrary Sequence Handle attribute. See the attribute description for more details.	ViInt32
Gain	Specifies the arbitrary waveform gain. The driver uses this value to set the Arbitrary Gain attribute. See the attribute description for more details.	ViReal64
Offset	Specifies the arbitrary waveform offset. The driver uses this value to set the Arbitrary Offset attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

.NET Exceptions

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

8.3.4 Create Arbitrary Sequence

Description

Creates an arbitrary waveform sequence from an array of waveform handles and a corresponding array of loop counts. The function returns a handle that identifies the sequence. You pass a sequence handle to the `Handle` parameter of the *Configure Arbitrary Sequence* function to produce that sequence.

If the function generator cannot store any more arbitrary sequences, this function returns the error No Sequences Available.

.NET Method Prototype

```
Int32 Arbitrary.Sequence.Create (Int32[] waveformHandle,
                                 Int32[] loopCount);
```

COM Method Prototype

```
HRESULT Arbitrary.Sequence.Create ([in] SAFEARRAY(LONG) *WfmHandle,
                                   [in] SAFEARRAY(LONG) *LoopCount,
                                   [out,retval] LONG *Handle);
```

C Prototype

```
ViStatus IviFgen_CreateArbSequence (ViSession Vi,
                                    ViInt32 Length,
                                    ViInt32 WfmHandle[],
                                    ViInt32 LoopCount[],
                                    ViInt32 *Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Length	Specifies the number of arbitrary waveforms in the new arbitrary sequence.	ViInt32
WfmHandle	Specifies the array of waveform handles for the new arbitrary sequence. The array must have at least as many elements as the value in the Length parameter (IVI-C only). Each WfmHandle array element has a corresponding LoopCount array element that specifies how many times that waveform is repeated. If the generator supports sequences of sequences (indicated by a Max Sequence Depth greater than 1), the user can pass sequence handles in this parameter as well.	ViInt32[]

LoopCount	Specifies the array of loop counts for the new arbitrary sequence. The array must have at least as many elements as the value in the length parameter (IVI-C only). Each LoopCount array element corresponds to a WfmHandle array element and indicates how many times to repeat that waveform. Each element of the LoopCount array must be less than or equal to the maximum number of loop counts the function generator allows. The function generator's maximum loop count is stored in the Loop Count Max attribute.	ViInt32[]

Outputs	Description	Base Type
Handle (C/COM)	Returns the handle that identifies the new arbitrary sequence.	ViInt32
Return Value (.NET)	Returns the handle that identifies the new arbitrary sequence.	ViInt32

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
No Sequences Available	The function generator's arbitrary sequence memory is full.

.NET Exceptions

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

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

Exception Class	Description
NoSequencesAvailableException	The function generator's arbitrary sequence memory is full.

8.3.5 Query Arbitrary Sequence Capabilities (IVI-C Only)

Description

Returns the attributes of the function generator that are related to creating arbitrary sequences. These attributes are the maximum number of sequences, minimum sequence length, maximum sequence length, and maximum loop count.

.NET Method Prototype

```
N/A
```

(Use the `Arbitrary.Sequence.NumberSequencesMax`, `LengthMin`, `LengthMax`, and `LoopCountMax` properties.)

COM Method Prototype

```
N/A
```

(Use the `Arbitrary.Sequence.NumberSequencesMax`, `LengthMin`, `LengthMax`, and `LoopCountMax` properties.)

C Prototype

```
ViStatus IviFgen_QueryArbSeqCapabilities (ViSession Vi,
                                          ViInt32 *MaxNumSeqs,
                                          ViInt32 *MinSeqLength,
                                          ViInt32 *MaxSeqLength,
                                          ViInt32 *MaxLoopCount);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Outputs	Description	Base Type
MaxNumSeqs	Returns the maximum number of sequences the function generator allows. The driver obtains this value from the Number Sequences Max attribute. See the attribute description for more details.	ViInt32
MinSeqLength	Returns the minimum number of waveforms the function generator allows in a sequence. The driver obtains this value from the Sequence Length Min attribute. See the attribute description for more details.	ViInt32
MaxSeqLength	Returns the maximum number of waveforms the function generator allows in a sequence. The driver obtains this value from the Sequence Length Max attribute. See the attribute description for more details.	ViInt32
MaxLoopCount	Returns the function generator's maximum loop count. The driver obtains this value from the Loop Count Max attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

8.4　IviFgenArbSeq Behavior Model

The IviFgenArbSeq Extension Group follows the behavior model of the IviFgenArbWfm capability group. The only modification to the behavior model from the IviFgenArbWfm capability group is the ability to configure IviFgenArbSeq settings.

8.5　IviFgenArbSeq Compliance Notes

1. If a specific driver implements the IviFgenArbSeq Extension Group, it shall also implement the IviFgenArbWfm Extension Group.

2. If a specific driver implements the IviFgenArbSeq Extension Group, it shall implement the value Output Sequence for the Output Mode attribute.

9 IviFgenTrigger Extension Group

9.1 IviFgenTrigger Overview

The IviFgenTrigger Extension Group supports function generators capable of configuring a trigger. This trigger source is used by other extension groups like IviFgenBurst to determine when to produce output generation. This extension group has been deprecated by the IviFgenStartTrigger Extension group. Drivers that support the IviFgenTrigger Extension group shall also support the IviFgenStartTrigger Extension group in order to be compliant with version 5.0 or later of the IviFgen class specification.

This extension affects instrument behavior when the Operation Mode attribute is set to Operate Burst.

9.2 IviFgenTrigger Attributes

The IviFgenTrigger capability group defines the following attribute:

* Trigger Source

This section describes the behavior and requirements of this attribute. The actual value for the attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

9.2.1 Trigger Source (C and COM Only)

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Trigger Source

.NET Property

> See section 10.2.3, *Start Trigger Source*.

COM Property Name

> Trigger.Source(BSTR ChannelName)

COM Enumeration Name

> IviFgenTriggerSourceEnum

C Constant Name

> IVIFGEN_ATTR_TRIGGER_SOURCE

Description

> Specifies the trigger source. After the function generator receives a trigger from this source, it produces a signal.

Defined Values

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

Name	Description	
	Language	**Identifier**
Internal Trigger	The function generator does not produce an output signal until it receives a trigger from its internal trigger source.	
	C	IVIFGEN_VAL_INTERNAL_TRIGGER
	COM	IviFgenTriggerSourceInternal
External Trigger	The function generator does not produce an output signal until it receives a trigger on its external trigger input terminal.	
	C	IVIFGEN_VAL_EXTERNAL
	COM	IviFgenTriggerSourceExternal
Software Trigger	The function generator does not produce an output signal until the Send Software Trigger function executes. Refer to the Standardized Cross Class Capabilities specification for a complete description of this value and the Send Software Trigger function.	

	C	IVIFGEN_VAL_SOFTWARE_TRIG
	COM	IviFgenTriggerSourceSoftware
TTL0 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL0 line.	
	C	IVIFGEN_VAL_TTL0
	COM	IviFgenTriggerSourceTTL0
TTL1 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL1 line.	
	C	IVIFGEN_VAL_TTL1
	COM	IviFgenTriggerSourceTTL1
TTL2 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL2 line.	
	C	IVIFGEN_VAL_TTL2
	COM	IviFgenTriggerSourceTTL2
TTL3 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL3 line.	
	C	IVIFGEN_VAL_TTL3
	COM	IviFgenTriggerSourceTTL3
TTL4 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL4 line.	
	C	IVIFGEN_VAL_TTL4
	COM	IviFgenTriggerSourceTTL4
TTL5 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL5 line.	
	C	IVIFGEN_VAL_TTL5
	COM	IviFgenTriggerSourceTTL5
TTL6 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL6 line.	
	C	IVIFGEN_VAL_TTL6
	COM	IviFgenTriggerSourceTTL6
TTL7 Trigger	The function generator does not produce an output signal until it receives a trigger from the TTL7 line.	
	C	IVIFGEN_VAL_TTL7
	COM	IviFgenTriggerSourceTTL7
ECL0 Trigger	The function generator does not produce an output signal until it receives a trigger from the ECL0 line.	
	C	IVIFGEN_VAL_ECL0
	COM	IviFgenTriggerSourceECL0
ECL1 Trigger	The function generator does not produce an output signal until it receives a trigger from the ECL1 line.	
	C	IVIFGEN_VAL_ECL1

		COM	`IviFgenTriggerSourceECL1`
PXI Star Trigger	The function generator does not produce an output signal until it receives a trigger from the PXI Star bus.		
		C	`IVIFGEN_VAL_PXI_STAR`
		COM	`IviFgenTriggerSourcePXIStar`
RTSI0 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI 0 line.		
		C	`IVIFGEN_VAL_RTSI_0`
		COM	`IviFgenTriggerSourceRTSI0`
RTSI1 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI 1 line.		
		C	`IVIFGEN_VAL_RTSI_1`
		COM	`IviFgenTriggerSourceRTSI1`
RTSI2 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI 2 line.		
		C	`IVIFGEN_VAL_RTSI_2`
		COM	`IviFgenTriggerSourceRTSI2`
RTSI3 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI 3 line.		
		C	`IVIFGEN_VAL_RTSI_3`
		COM	`IviFgenTriggerSourceRTSI3`
RTSI4 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI4 line.		
		C	`IVIFGEN_VAL_RTSI_4`
		COM	`IviFgenTriggerSourceRTSI4`
RTSI5 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI5 line.		
		C	`IVIFGEN_VAL_RTSI_5`
		COM	`IviFgenTriggerSourceRTSI5`
RTSI6 Trigger	The function generator does not produce an output signal until it receives a trigger from the RTSI6 line.		
		C	`IVIFGEN_VAL_RTSI_6`
		COM	`IviFgenTriggerSourceRTSI6`

.NET Exceptions

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

Compliance Notes

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

Name	Required Capability Group
Internal Trigger	IviFgenInternalTrigger
Software Trigger	IviFgenSoftwareTrigger

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

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

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

Refer to Section 30, *IviFgen Attribute Value Definitions*, for the definitions of Trigger Source Specific Ext Base, IVIFGEN_VAL_TRIG_SRC_SPECIFIC_EXT_BASE and IVIFGEN_VAL_TRIG_SRC_CLASS_EXT_BASE.

9.3   IviFgenTrigger Functions

The IviFgenTrigger extension defines the following functions:

- Configure Trigger Source (IVI-C Only)

This section describes the behavior and requirements of each function.

9.3.1 Configure Trigger Source (IVI-C Only)

Description

Configures the function generator's trigger source attribute.

.NET Method Prototype

```
N/A
```

(Use Trigger.SetSource. See the Trigger Source attribute section.)

COM Method Prototype

```
N/A
```

(Use the Trigger.Source property.)

C Prototype

```
ViStatus IviFgen_ConfigureTriggerSource (ViSession Vi,
                                         ViConstString ChannelName,
                                         ViInt32 Source);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the trigger signal. The driver uses this value to set the Trigger Source attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

9.4 IviFgenTrigger Behavior Model

The IviFgenTrigger Extension Group follows the behavior model of the IviFgenBase capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenTrigger settings.

9.5 IviFgenTrigger Compliance Notes

1. If a specific driver implements the IviFgenTrigger Extension Group, it shall also implement the IviFgenStartTrigger Extension Group.

10 IviFgenStartTrigger Extension Group

10.1 IviFgenStartTrigger Overview

The IviFgenStartTrigger Extension Group supports function generators capable of configuring a start trigger. A start trigger initiates generation of a waveform or sequence. This Extension group deprecates the IviFgenTrigger extension group. Drivers that implement this extension group shall implement the IviFgenTrigger extension group as well, to ensure that applications based on previous versions of the IviFgen class specification continue to work with version 5.0.

Setting the Start Trigger Source attribute to a value other than None enables the start trigger. To disable the start trigger, set the Start Trigger Source to None.

10.2 IviFgenStartTrigger Attributes

The IviFgenStartTrigger extension group defines the following attributes:

- Start Trigger Delay
- Start Trigger Slope
- Start Trigger Source
- Start Trigger Threshold

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

10.2.1 Start Trigger Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

> PrecisionTimeSpan Trigger.Start.GetDelay(String channelName)
>
> void Trigger.Start.SetDelay(String channelName, PrecisionTimeSpan delay)

COM Property Name

> Trigger.Start.Delay(BSTR ChannelName)

COM Enumeration Name

> N/A

C Constant Name

> IVIFGEN_ATTR_START_TRIGGER_DELAY

Description

> Specifies an additional length of time to delay from the start trigger to the first point in the
> waveform generation. 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 these methods.

10.2.2 Start Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	Configure Start Trigger

.NET Property Using Channel-Dependent Get and Set methods

    TriggerSlope Trigger.Start.GetSlope(String channelName)

    void Trigger.Start.SetSlope(String channelName, TriggerSlope slope)

.NET Enumeration Name

    TriggerSlope

COM Property Name

    Trigger.Start.Slope(BSTR ChannelName)

COM Enumeration Name

    IviFgenTriggerSlopeEnum

C Constant Name

    IVIFGEN_ATTR_START_TRIGGER_SLOPE

Description

Specifies the slope of the trigger that starts the generator.

Defined Values

Name	Description		
	Language	*Identifier*	
Positive	The generator triggers on a positive slope.		
	.NET	Positive	
	C	IVIFGEN_VAL_TRIGGER_POSITIVE	
	COM	IviFgenTriggerSlopePositive	
Negative	The generator triggers on a negative slope.		
	.NET	Negative	
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE	
	COM	IviFgenTriggerSlopNegative	
Either	The generator triggers on either a positive or negative slope.		
	.NET	Either	
	C	IVIFGEN_VAL_TRIGGER_EITHER	
	COM	IviFgenTriggerSlopeEither	

.NET Exceptions

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

10.2.3 Start Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	Channel	None	Configure Start Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
String Trigger.Start.GetSource(String channelName)

void Trigger.Start.SetSource(String channelName, String source)
```

COM Property Name

```
Trigger.Start.Source(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_START_TRIGGER_SOURCE
```

Description

Specifies the source of the start trigger.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in Section 29, *IviFgen Attribute ID Definitions*.

.NET Exceptions

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

10.2.4 Start Trigger Threshold

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

> Double Trigger.Start.GetThreshold(String channelName)
>
> void Trigger.Start.SetThreshold(String channelName, Double threshold)

COM Property Name

> Trigger.Start.Threshold(BSTR ChannelName)

COM Enumeration Name

> N/A

C Constant Name

> IVIFGEN_ATTR_START_TRIGGER_THRESHOLD

Description

> Specifies the voltage threshold for the start trigger. The units are volts.

.NET Exceptions

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

10.3 IviFgenStartTrigger Functions

The IviFgenStartTrigger extension group defines the following function:

- Configure Start Trigger

This section describes the behavior and requirements of this function.

10.3.1 Configure Start Trigger

Description

This function configures the start trigger properties.

.NET Method Prototype

```
void Trigger.Start.Configure (String channelName,
                  String source,
                  TriggerSlope slope);
```

COM Method Prototype

```
HRESULT Trigger.Start.Configure (BSTR ChannelName,
                  [in] BSTR Source,
                  [in] IviFgenTriggerSlopeEnum Slope);
```

C Prototype

```
ViStatus IviFgen_ConfigureStartTrigger (ViSession Vi,
                  ViConstString ChannelName,
                  ViConstString Source,
                  ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the start trigger. The driver uses this value to set the Start Trigger Source attribute. See the attribute description for more details.	ViConstString
Slope	Specifies the slope of the start trigger. This value sets the Start Trigger Slope attribute.	ViInt32

Defined Values for the Slope Parameter

Name	*Description*	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE

	COM	IviFgenTriggerSlopNegative
Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

Defined Values for the Source Parameter

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

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.

10.4 IviFgenStartTrigger Behavior Model

The IviFgenStartTrigger extension group follows the same behavior model as the IviFgenTrigger capability group described in Section 9.4 *IviFgenTrigger Behavior Model.*

10.5 IviFgenStartTrigger Compliance Notes

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

11 IviFgenStopTrigger Extension Group

11.1 IviFgenStopTrigger Overview

The IviFgenStopTrigger Extension Group supports function generators capable of configuring a stop trigger. A stop trigger terminates any generation and has the same effect as calling the AbortGeneration function.

Setting the Stop Trigger Source attribute to a value other than None enables the stop trigger. To disable the stop trigger, set the Stop Trigger Source to None.

11.2 IviFgenStopTrigger Attributes

The IviFgenStopTrigger extension group defines the following attributes:

- Stop Trigger Delay

- Stop Trigger Slope

- Stop Trigger Source

- Stop Trigger Threshold

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

11.2.1 Stop Trigger Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

        PrecisionTimeSpan Trigger.Stop.GetDelay(String channelName)

        void Trigger.Stop.SetDelay(String channelName, PrecisionTimeSpan delay)

COM Property Name

        Trigger.Stop.Delay(BSTR ChannelName)

COM Enumeration Name

    N/A

C Constant Name

        IVIFGEN_ATTR_STOP_TRIGGER_DELAY

Description

Specifies an additional length of time to delay from the stop trigger to the termination of the generation. 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 these methods.

11.2.2 Stop Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	Configure Stop Trigger

.NET Property Using Channel-Dependent Get and Set methods

        TriggerSlope Trigger.Stop.GetSlope(String channelName)

        void Trigger.Stop.SetSlope(String channelName, TriggerSlope slope)

.NET Enumeration Name

        TriggerSlope

COM Property Name

        Trigger.Stop.Slope(BSTR ChannelName)

COM Enumeration Name

        IviFgenTriggerSlopeEnum

C Constant Name

        IVIFGEN_ATTR_STOP_TRIGGER_SLOPE

Description

        Specifies the slope of the stop trigger.

Defined Values

Name	Description	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE
	COM	IviFgenTriggerSlopNegative
Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

.NET Exceptions

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

11.2.3 Stop Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	Channel	None	Configure Stop Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
String Trigger.Stop.GetSource(String channelName)

void Trigger.Stop.SetSource(String channelName, String source)
```

COM Property Name

```
Trigger.Stop.Source(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_STOP_TRIGGER_SOURCE
```

Description

Specifies the source of the stop trigger.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in Section 30, *IviFgen Attribute Value Definitions*.

.NET Exceptions

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

11.2.4 Stop Trigger Threshold

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

```
Double Trigger.Stop.GetThreshold(String channelName)

void Trigger.Stop.SetThreshold(String channelName, Double threshold)
```

COM Property Name

```
Trigger.Stop.Threshold(BSTR ChannelName)
```

COM Enumeration Name

N/A

C Constant Name

```
IVIFGEN_ATTR_STOP_TRIGGER_THRESHOLD
```

Description

Specifies the voltage threshold for the stop trigger. The units are volts.

.NET Exceptions

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

11.3 IviFgenStopTrigger Functions

The IviFgenStopTrigger extension group defines the following function:

- Configure Stop Trigger

- Send Software Stop Trigger

This section describes the behavior and requirements of this function.

11.3.1 Configure Stop Trigger

Description

This function configures the stop trigger properties.

.NET Method Prototype

```
void Trigger.Stop.Configure (String channelName,
                String source,
                TriggerSlope slope);
```

COM Method Prototype

```
HRESULT Trigger.Stop.Configure (BSTR ChannelName,
                [in] BSTR Source,
                [in] IviFgenTriggerSlopeEnum Slope);
```

C Prototype

```
ViStatus IviFgen_ConfigureStopTrigger (ViSession Vi,
                ViConstString ChannelName,
                ViConstString Source,
                ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the stop trigger. The driver uses this value to set the Stop Trigger Source attribute. See the attribute description for more details.	ViConstString
Slope	Specifies the slope of the trigger that stops the generator. This value sets the Stop Trigger Slope attribute.	ViInt32

Defined Values for the Slope Parameter

Name	Description		
		Language	*Identifier*
Positive	The generator triggers on a positive slope.		
		.NET	Positive
		C	IVIFGEN_VAL_TRIGGER_POSITIVE
		COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.		
		.NET	Negative
		C	IVIFGEN_VAL_TRIGGER_NEGATIVE
		COM	IviFgenTriggerSlopNegative

Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

Defined Values for the Source Parameter

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

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.

11.3.2 Send Software Stop Trigger

Description

This function sends a software-generated stop trigger to the instrument.

.NET Method Prototype

```
void Trigger.Stop.SendSoftwareTrigger();
```

COM Method Prototype

```
HRESULT Trigger.Stop.SendSoftwareTrigger ();
```

C Prototype

```
ViStatus IviFgen_SendSoftwareStopTrigger (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. This function can also return this additional standard cross-class status code:

- Trigger Not Software

Trigger Not Software is defined in *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 method. This method can also throw this additional standard cross-class exception:

- TriggerNotSoftwareException

TriggerNotSoftwareException is defined in *IVI-3.2: Inherent Capabilities Specification*.

11.4 IviFgenStopTrigger Behavior Model

The IviFgenStopTrigger extension group follows the same behavior model as the IviFgenTrigger capability group described in Section 9.4 IviFgenTrigger Behavior Model.

11.5 IviFgenStopTrigger Compliance Notes

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

12 IviFgenHoldTrigger Extension Group

12.1 IviFgenHoldTrigger Overview

The IviFgenHoldTrigger Extension Group supports function generators capable of configuring a hold trigger. A hold trigger pauses generation. From the paused state, a resume trigger resumes generation; a stop trigger terminates generation; start trigger behavior is vendor defined.

Setting the Hold Trigger Source attribute to a value other than None enables the hold trigger. To disable the hold trigger, set the Hold Trigger Source to None.

12.2 IviFgenHoldTrigger Attributes

The IviFgenHoldTrigger extension group defines the following attributes:

- Hold Trigger Delay
- Hold Trigger Slope
- Hold Trigger Source
- Hold Trigger Threshold

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

12.2.1 Hold Trigger Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

> PrecisionTimeSpan Trigger.Hold.GetDelay(String channelName)
>
> void Trigger.Hold.SetDelay(String channelName, PrecisionTimeSpan delay)

COM Property Name

> Trigger.Hold.Delay(BSTR ChannelName)

COM Enumeration Name

> N/A

C Constant Name

> IVIFGEN_ATTR_HOLD_TRIGGER_DELAY

Description

> Specifies an additional length of time to delay from the hold trigger to the pause of the generation.
> 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 these methods.

12.2.2 Hold Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	Configure Hold Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
TriggerSlope Trigger.Hold.GetSlope(String channelName)

void Trigger.Hold.SetSlope(String channelName, TriggerSlope slope)
```

.NET Enumeration Name

```
TriggerSlope
```

COM Property Name

```
Trigger.Hold.Slope(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenTriggerSlopeEnum
```

C Constant Name

```
IVIFGEN_ATTR_HOLD_TRIGGER_SLOPE
```

Description

Specifies the slope of the hold trigger.

Defined Values

Name	*Description*	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	`Positive`
	C	`IVIFGEN_VAL_TRIGGER_POSITIVE`
	COM	`IviFgenTriggerSlopePositive`
Negative	The generator triggers on a negative slope.	
	.NET	`Negative`
	C	`IVIFGEN_VAL_TRIGGER_NEGATIVE`
	COM	`IviFgenTriggerSlopNegative`
Either	The generator triggers on either a positive or negative slope.	
	.NET	`Either`
	C	`IVIFGEN_VAL_TRIGGER_EITHER`
	COM	`IviFgenTriggerSlopeEither`

.NET Exceptions

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

12.2.3 Hold Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	Channel	None	Configure Hold Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
String Trigger.Hold.GetSource(String channelName)

void Trigger.Hold.SetSource(String channelName, String source)
```

COM Property Name

```
Trigger.Hold.Source(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_HOLD_TRIGGER_SOURCE
```

Description

Specifies the source of the hold trigger.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in Section 30, *IviFgen Attribute Value Definitions*.

.NET Exceptions

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

12.2.4 Hold Trigger Threshold

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

```
Double Trigger.Hold.GetThreshold(String channelName)

void Trigger.Hold.SetThreshold(String channelName, Double threshold)
```

COM Property Name

```
Trigger.Hold.Threshold(BSTR ChannelName)
```

COM Enumeration Name

N/A

C Constant Name

```
IVIFGEN_ATTR_HOLD_TRIGGER_THRESHOLD
```

Description

Specifies the voltage threshold for the hold trigger. The units are volts.

.NET Exceptions

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

12.3 IviFgenHoldTrigger Functions

The IviFgenHoldTrigger extension group defines the following function:

- Configure Hold Trigger

- Send Software Hold Trigger

This section describes the behavior and requirements of this function.

12.3.1 Configure Hold Trigger

Description

This function configures the hold trigger properties.

.NET Method Prototype

```
void Trigger.Hold.Configure(String channelName,
              String source,
              TriggerSlope slope);
```

COM Method Prototype

```
HRESULT Trigger.Hold.Configure(BSTR ChannelName,
              [in] BSTR Source,
              [in] IviFgenTriggerSlopeEnum Slope);
```

C Prototype

```
ViStatus IviFgen_ConfigureHoldTrigger(ViSession Vi,
              ViConstString ChannelName,
              ViConstString Source,
              ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the hold trigger. The driver uses this value to set the Hold Trigger Source attribute. See the attribute description for more details.	ViConstString
Slope	Specifies the slope of the hold trigger. This value sets the Hold Trigger Slope attribute.	ViInt32

Defined Values for the Slope Parameter

Name	Description	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE
	COM	IviFgenTriggerSlopNegative

Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

Defined Values for the Source Parameter

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

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.

12.3.2 Send Software Hold Trigger

Description

This function sends a software-generated hold trigger to the instrument.

.NET Method Prototype

```
void Trigger.Hold.SendSoftwareTrigger();
```

COM Method Prototype

```
HRESULT Trigger.Hold.SendSoftwareTrigger ();
```

C Prototype

```
ViStatus IviFgen_SendSoftwareHoldTrigger (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.
This function can also return this additional standard cross-class status code:
• Trigger Not Software

.NET 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. This method can also throw this additional standard cross-class exception:

• TriggerNotSoftwareException

TriggerNotSoftwareException is defined in *IVI-3.2: Inherent Capabilities Specification*.

12.4 IviFgenHoldTrigger Behavior Model

The IviFgenHoldTrigger extension group follows the same behavior model as the IviFgenTrigger capability group described in Section 9.4 *IviFgenTrigger Behavior Model.*

12.5 IviFgenHoldTrigger Compliance Notes

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

13 IviFgenResumeTrigger Extension Group

13.1 IviFgenResumeTrigger Overview

The IviFgenResumeTrigger Extension Group supports function generators capable of configuring a resume trigger. A resume trigger resumes generation after it has been paused by a hold trigger, starting with the next point.

Setting the Resume Trigger Source attribute to a value other than None enables the resume trigger. To disable the resume trigger, set the Resume Trigger Source to None.

13.2 IviFgenResumeTrigger Attributes

The IviFgenResumeTrigger extension group defines the following attributes:

- Resume Trigger Delay

- Resume Trigger Slope

- Resume Trigger Source

- Resume Trigger Threshold

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

13.2.1 Resume Trigger Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

   PrecisionTimeSpan Trigger.Resume.GetDelay(String channelName)

   void Trigger.Resume.SetDelay(String channelName, PrecisionTimeSpan delay)

COM Property Name

   Trigger.Resume.Delay(BSTR ChannelName)

COM Enumeration Name

   N A

C Constant Name

   IVIFGEN_ATTR_RESUME_TRIGGER_DELAY

Description

   Specifies an additional length of time to delay from the resume trigger to the resumption of the generation. 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 these methods.

13.2.2 Resume Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	Configure Resume Trigger

.NET Property Using Channel-Dependent Get and Set methods

    TriggerSlope Trigger.Resume.GetSlope(String channelName)

    void Trigger.Resume.SetSlope(String channelName, TriggerSlope slope)

.NET Enumeration Name

    TriggerSlope

COM Property Name

    Trigger.Resume.Slope(BSTR ChannelName)

COM Enumeration Name

    IviFgenTriggerSlopeEnum

C Constant Name

    IVIFGEN_ATTR_RESUME_TRIGGER_SLOPE

Description

Specifies the slope of the resume trigger.

Defined Values

Name	*Description*	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE
	COM	IviFgenTriggerSlopNegative
Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

.NET Exceptions

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

13.2.3 Resume Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	Channel	None	Configure Resume Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
String Trigger.Resume.GetSource(String channelName)

void Trigger.Resume.SetSource(String channelName, String source)
```

COM Property Name

```
Trigger.Resume.Source(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_RESUME_TRIGGER_SOURCE
```

Description

Specifies the source of the resume trigger.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in Section 30, *IviFgen Attribute Value Definitions*.

.NET Exceptions

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

13.2.4 Resume Trigger Threshold

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

```
Double Trigger.Resume.GetThreshold(String channelName)

void Trigger.Resume.SetThreshold(String channelName, Double threshold)
```

COM Property Name

```
Trigger.Resume.Threshold(BSTR ChannelName)
```

COM Enumeration Name

N/A

C Constant Name

```
IVIFGEN_ATTR_RESUME_TRIGGER_THRESHOLD
```

Description

Specifies the voltage threshold for the resume trigger. The units are volts.

.NET Exceptions

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

13.3 IviFgenResumeTrigger Functions

The IviFgenResumeTrigger extension group defines the following function:

- Configure Resume Trigger

- Send Software Resume Trigger

This section describes the behavior and requirements of this function.

13.3.1 Configure Resume Trigger

Description

This function configures the resume trigger properties.

.NET Method Prototype

```
void Trigger.Resume.Configure (String channelName,
                    String source,
                    TriggerSlope slope);
```

COM Method Prototype

```
HRESULT Trigger.Resume.Configure (BSTR ChannelName,
                    [in] BSTR Source,
                    [in] IviFgenTriggerSlopeEnum Slope);
```

C Prototype

```
ViStatus IviFgen_ConfigureResumeTrigger (ViSession Vi,
                    ViConstString ChannelName,
                    ViConstString Source,
                    ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the resume trigger. The driver uses this value to set the Resume Trigger Source attribute. See the attribute description for more details.	ViConstString
Slope	Specifies the slope of the resume trigger. This value sets the Resume Trigger Slope attribute.	ViInt32

Defined Values for the Slope Parameter

Name	Description		
	Language	*Identifier*	
Positive	The generator triggers on a positive slope.		
	.NET	Positive	
	C	IVIFGEN_VAL_TRIGGER_POSITIVE	
	COM	IviFgenTriggerSlopePositive	
Negative	The generator triggers on a negative slope.		
	.NET	Negative	
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE	
	COM	IviFgenTriggerSlopNegative	

Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

Defined Values for the Source Parameter

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

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.

13.3.2 Send Software Resume Trigger

Description

This function sends a software-generated resume trigger to the instrument.

.NET Method Prototype

void Trigger.Resume.SendSoftwareTrigger();

COM Method Prototype

HRESULT Trigger.Resume.SendSoftwareTrigger ();

C Prototype

ViStatus IviFgen_SendSoftwareResumeTrigger (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.
This function can also return this additional standard cross-class status code:

• Trigger Not Software

.NET 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. This method can also throw this additional standard cross-class exception:

- TriggerNotSoftwareException

TriggerNotSoftwareException is defined in *IVI-3.2: Inherent Capabilities Specification*.

13.4 IviFgenResumeTrigger Behavior Model

The IviFgenResumeTrigger extension group follows the same behavior model as the IviFgenTrigger capability group described in Section 9.4 *IviFgenTrigger Behavior Model.*

13.5 IviFgenResumeTrigger Compliance Notes

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

14 IviFgenAdvanceTrigger Extension Group

14.1 IviFgenAdvanceTrigger Overview

The IviFgenAdvanceTrigger Extension Group supports function generators capable of configuring an advance trigger. An advance trigger advances generation to the end of the current waveform, where generation proceeds according to the current configuration.

Setting the Advance Trigger Source attribute to a value other than None enables the advance trigger. To disable the advance trigger, set the Advance Trigger Source to None.

14.2 IviFgenAdvanceTrigger Attributes

The IviFgenAdvanceTrigger extension group defines the following attributes:

- Advance Trigger Delay
- Advance Trigger Slope
- Advance Trigger Source
- Advance Trigger Threshold

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

14.2.1 Advance Trigger Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

> PrecisionTimeSpan Trigger.Advance.GetDelay(String channelName)
>
> void Trigger.Advance.SetDelay(String channelName,
>   PrecisionTimeSpan delay)

COM Property Name

> Trigger.Advance.Delay(BSTR ChannelName)

COM Enumeration Name

> N/A

C Constant Name

> IVIFGEN_ATTR_ADVANCE_TRIGGER_DELAY

Description

> Specifies an additional length of time to delay from the advance trigger to the advancing to the end of the current waveform. 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 these methods.

14.2.2 Advance Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	Configure Advance Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
TriggerSlope Trigger.Advance.GetSlope(String channelName)

void Trigger.Advance.SetSlope(String channelName, TriggerSlope slope)
```

.NET Enumeration Name

TriggerSlope

COM Property Name

```
Trigger.Advance.Slope(BSTR ChannelName)
```

COM Enumeration Name

IviFgenTriggerSlopeEnum

C Constant Name

IVIFGEN_ATTR_ADVANCE_TRIGGER_SLOPE

Description

Specifies the slope of the advance trigger.

Defined Values

Name	Description	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE
	COM	IviFgenTriggerSlopNegative
Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

.NET Exceptions

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

14.2.3 Advance Trigger Source

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	Channel	None	Configure Advance Trigger

.NET Property Using Channel-Dependent Get and Set methods

```
String Trigger.Advance.GetSource(String channelName)

void Trigger.Advance.SetSource(String channelName, String source)
```

COM Property Name

```
Trigger.Advance.Source(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_ADVANCE_TRIGGER_SOURCE
```

Description

Specifies the source of the advance trigger.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

Compliance Notes

1. The specific driver need not support all of the trigger source values defined in Section 30, *IviFgen Attribute Value Definitions*.

.NET Exceptions

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

14.2.4 Advance Trigger Threshold

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64	R/W	Channel	Down	None

.NET Property Using Channel-Dependent Get and Set methods

```
Double Trigger.Advance.GetThreshold(String channelName)

void Trigger.Advance.SetThreshold(String channelName, Double threshold)
```

COM Property Name

```
Trigger.Advance.Threshold(BSTR ChannelName)
```

COM Enumeration Name

N/A

C Constant Name

```
IVIFGEN_ATTR_ADVANCE_TRIGGER_THRESHOLD
```

Description

Specifies the voltage threshold for the advance trigger. The units are volts.

.NET Exceptions

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

14.3 IviFgenAdvanceTrigger Functions

The IviFgenAdvanceTrigger extension group defines the following function:

- Configure Advance Trigger

- Send Software Advance Trigger

This section describes the behavior and requirements of this function.

14.3.1 Configure Advance Trigger

Description

This function configures the advance trigger properties.

.NET Method Prototype

```
void Trigger.Advance.Configure (String channelName,
                  String source,
                  TriggerSlope slope);
```

COM Method Prototype

```
HRESULT Trigger.Advance.Configure (BSTR ChannelName,
                  [in] BSTR Source,
                  [in] IviFgenTriggerSlopeEnum Slope);
```

C Prototype

```
ViStatus IviFgen_ConfigureAdvanceTrigger (ViSession Vi,
                  ViConstString ChannelName,
                  ViConstString Source,
                  ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Source	Specifies the source of the advance trigger. The driver uses this value to set the Advance Trigger Source attribute. See the attribute description for more details.	ViConstString
Slope	Specifies the slope of the advance trigger. This value sets the Advance Trigger Slope attribute.	ViInt32

Defined Values for the Source Parameter

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

Defined Values for the Slope Parameter

Name	Description	
	Language	*Identifier*
Positive	The generator triggers on a positive slope.	
	.NET	Positive
	C	IVIFGEN_VAL_TRIGGER_POSITIVE
	COM	IviFgenTriggerSlopePositive
Negative	The generator triggers on a negative slope.	
	.NET	Negative
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE

	COM	IviFgenTriggerSlopNegative
Either	The generator triggers on either a positive or negative slope.	
	.NET	Either
	C	IVIFGEN_VAL_TRIGGER_EITHER
	COM	IviFgenTriggerSlopeEither

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.

14.3.2 Send Software Advance Trigger

Description

This function sends a software-generated advance trigger to the instrument.

.NET Method Prototype

void Trigger.Advance.SendSoftwareTrigger();

COM Method Prototype

HRESULT Trigger.Advance.SendSoftwareTrigger ();

C Prototype

ViStatus IviFgen_SendSoftwareAdvanceTrigger (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.
This function can also return this additional standard cross-class status code:

• Trigger Not Software

.NET 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. This method can also throw this additional standard cross-class exception:

• TriggerNotSoftwareException

TriggerNotSoftwareException is defined in *IVI-3.2: Inherent Capabilities Specification*.

14.4 IviFgenAdvanceTrigger Behavior Model

The IviFgenAdvanceTrigger extension group follows the same behavior model as the IviFgenTrigger capability group described in Section 9.4 *IviFgenTrigger Behavior Model.*

14.5 IviFgenAdvanceTrigger Compliance Notes

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

15 IviFgenInternalTrigger Extension Group

15.1 IviFgenInternalTrigger Overview

The IviFgenInternalTrigger Extension Group supports function generators that can generate output based on an internally generated trigger signal. The user can configure the rate at which internal triggers are generated.

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

15.2 IviFgenInternalTrigger Attributes

The IviFgenInternalTrigger capability group defines the following attribute:

- Internal Trigger Rate

This section describes the behavior and requirements of this attribute. The actual value for the attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

15.2.1 Internal Trigger Rate

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N A	None	Configure Internal Trigger Rate

.NET Property Name

Trigger.InternalRate

COM Property Name

Trigger.InternalRate

C Constant Name

IVIFGEN_ATTR_INTERNAL_TRIGGER_RATE

Description

Specifies the rate at which the function generator's internal trigger source produces a trigger, in triggers per second.

.NET Exceptions

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

15.3 IviFgenInternalTrigger Functions

The IviFgenInternalTrigger extension defines the following functions:

- Configure Internal Trigger Rate (IVI-C Only)

This section describes the behavior and requirements of each function.

15.3.1 Configure Internal Trigger Rate (IVI-C Only)

Description

Configures the function generator's internal trigger rate.

.NET Method Prototype

```
N/A
```

(Use the `Trigger.Rate` property.)

COM Method Prototype

```
N/A
```

(Use the `Trigger.Rate` property.)

C Prototype

```
ViStatus IviFgen_ConfigureInternalTriggerRate (ViSession Vi,
                                               ViReal64 Rate);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Rate	Specifies the rate at which the function generator's internal trigger source produces triggers. The driver uses this value to set the Internal Trigger Rate attribute. See the attribute description for more details.	ViReal64

Return Values (C)

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

15.4 IviFgenInternalTrigger Behavior Model

The IviFgenInternalTrigger Extension Group follows the behavior model of the IviFgenTrigger capability group. The only modification to the behavior model from the IviFgenTrigger capability group is the ability to configure IviFgenInternalTrigger settings.

15.5 IviFgenInternalTrigger Compliance Notes

1. If a specific driver implements the IviFgenInternalTrigger Extension Group, it shall implement the IviFgenTrigger and IviFgenStartTrigger Extension Groups.

2. If a specific driver implements the IviFgenInternalTrigger Extension Group, it shall implement the Internal Trigger value for the Trigger Source attribute.

3. If a specific driver implements the IviFgenInternalTrigger Extension Group, it shall implement the Internal Trigger string for the Start Trigger Source attribute.

16 IviFgenSoftwareTrigger Extension Group

16.1 IviFgenSoftwareTrigger Overview

The IviFgenSoftwareTrigger Extension Group supports function generators that can generate output based on a software trigger signal. The user can send a software trigger to cause signal output to occur.

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

16.2 IviFgenSoftwareTrigger Functions

The IviFgenSoftwareTrigger extension defines the following functions:

- Send Software Trigger

This section describes the behavior and requirements of this function.

16.2.1 Send Software Trigger

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

In the Fgen specification, Send Software Trigger applies to start triggers. In the IVI-C and IVI-COM hierarchies, Send Software Trigger is a child of the Trigger node. In the IVI.NET hierarchy, Send Software Trigger is a child of the Trigger.Start node. Other trigger types have their own Send Software Trigger methods – for example, Send Software Resume Trigger.

16.3 IviFgenSoftwareTrigger Behavior Model

The IviFgenSoftwareTrigger Extension Group follows the behavior model of the IviFgenTrigger capability group. The only modification to the behavior model from the IviFgenTrigger capability group is the ability to send software triggers.

16.4 IviFgenSoftwareTrigger Compliance Notes

1. If a specific driver implements the IviFgenSoftwareTrigger Extension Group, it shall implement the IviFgenTrigger and IviFgenStartTrigger Extension Groups.

2. If a specific driver implements the IviSoftwareTrigger Extension Group, it shall implement the value Software Trigger for the Trigger Source attribute.

3. If a specific driver implements the IviSoftwareTrigger Extension Group, it shall implement the Software Trigger string for the Start Trigger Source attribute.

17 IviFgenBurst Extension Group

17.1 IviFgenBurst Overview

The IviFgenBurst Extension Group supports function generators capable of generating a discrete number of waveform cycles based on a trigger. The trigger is configured with the IviFgenTrigger or IviFgenStartTrigger extension group. The user can specify the number of waveform cycles to generate when a trigger event occurs.

For standard and arbitrary waveforms, a cycle is one period of the waveform. For arbitrary sequences, a cycle is one complete progression through the generation of all iterations of all waveforms in the sequence.

This extension affects instrument behavior when the Operation Mode attribute is set to Operate Burst.

17.2 IviFgenBurst Attributes

The IviFgenBurst capability group defines the following attribute:

- Burst Count

This section describes the behavior and requirements of this attribute. The actual value for the attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

17.2.1 Burst Count

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure Burst Count

.NET Property Using Channel-Dependent Get and Set methods

```
Int32 Trigger.Start.GetBurstCount (String channelName)

void Trigger.Start.SetBurstCount (String channelName, Int32 burstCount)
```

COM Property Name

```
Trigger.BurstCount(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_BURST_COUNT
```

Description

Specifies the number of waveform cycles that the function generator produces after it receives a trigger.

.NET Exceptions

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

17.3 IviFgenBurst Functions

The IviFgenBurst extension defines the following functions:

- Configure Burst Count (IVI-C Only)

This section describes the behavior and requirements of each function.

17.3.1 Configure Burst Count (IVI-C Only)

Description

Configures the burst count attribute.

.NET Method Prototype

```
N/A
```

(Use Trigger.SetBurstCount. See the Burst Count attribute section.)

COM Method Prototype

```
N/A
```

(Use the Output.BurstCount property.)

C Prototype

```
ViStatus IviFgen_ConfigureBurstCount (ViSession Vi,
                                      ViConstString ChannelName,
                                      ViInt32 Count);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the trigger.	ViConstString
Count	Specifies the burst count. The driver uses this value to set the Burst Count attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

17.4 IviFgenBurst Behavior Model

The behavior model for the IviFgenBurst extension capability group leverages the behavior model for the IviFgenBase extension group. The only difference is what happens while the function generator is in the Output Generation State. The following diagram shows the Output Generation State for the IviFgenBurst extension group.
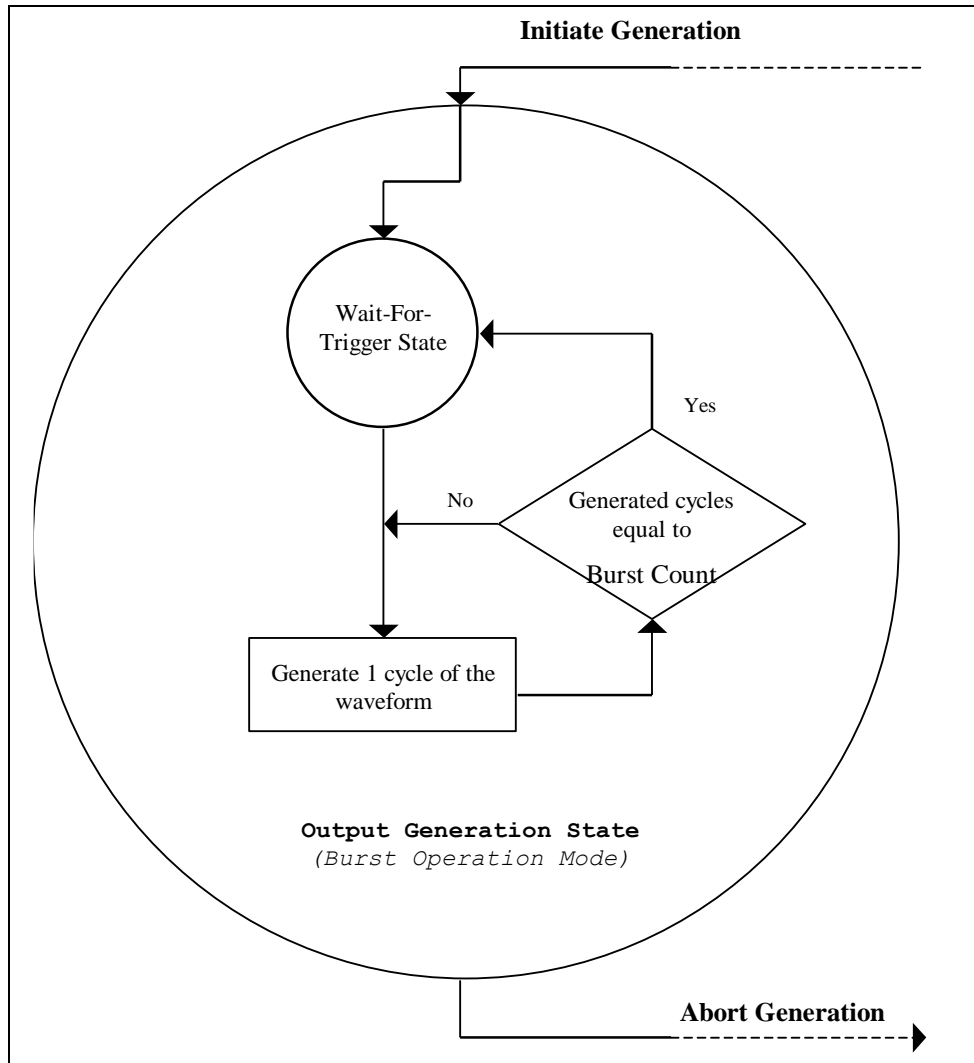
Figure 12-1. IviFgenBurst Behavior Model

When the function generator enters the Output Generation State and the Operation attribute is set to Operate Burst the function generator enters the Wait-For-Trigger state. The function generator exits the Wait-For-Trigger state when it receives a start trigger from the source that the Trigger Source or Start Trigger Source attribute specifies.

After the function generator exits the Wait-For-Trigger state it generates one complete cycle of the currently configured waveform. The Output Mode attribute determines the type of waveform that the function generator produces. The waveform could be a standard function, an arbitrary waveform, or an arbitrary sequence.

After the function generator produces each waveform cycle, the function generator compares the total number of waveform cycles that have been generated since the function generator exited the Wait-For-Trigger state with the value of the Burst Count attribute. If the total number of waveforms generated is less than the value of the Burst Count attribute, the function generator produces another waveform cycle. If the total number of waveforms generated is equal to the value of the Burst Count attribute, the function generator returns to the Wait-For-Trigger state.

After the function generator begins to generate waveform cycles, it ignores all start trigger signals until Burst Count waveform cycles have been generated.

17.5 IviFgenBurst Compliance Notes

1. If an instrument driver implements the IviFgenBurst Extension Group, it shall implement the IviFgenTrigger and the IviFgenStartTrigger Extension Groups.

2. If an instrument driver implements the IviFgenBurst Extension Group, it shall implement the Operate Burst value for the Operation Mode attribute

18 IviFgenModulateAM Extension Group

18.1 IviFgenModulateAM Overview

The IviFgenModulateAM Extension Group supports function generators that can apply amplitude modulation to an output signal. The user can enable or disable amplitude modulation, and specify the source of the modulating waveform. If the function generator supports an internal modulating waveform source, the user can specify the waveform, frequency, and modulation depth.

Amplitude modulation is accomplished by varying the amplitude of a carrier waveform according to the amplitude of a modulating waveform. The general equation for applying amplitude modulation to a waveform is,

$$AM(t) = [M(t) + 1] \times C(t),$$

where $C(t)$ is the carrier waveform, $M(t)$ is the modulating waveform, and $AM(t)$ is the modulated signal.

This specification provides modulating waveform property definitions that must be followed when developing specific instrument drivers. The carrier waveform is defined as the waveform the function generator produces without any modulation. You configure the carrier waveform with the IviFgenStdFunc, IviFgenArbWfm, or IviFgenArbSeq capability groups.

The modulating waveform is defined by the following properties.

Waveform – The overall "shape" of one period of the modulating waveform. This specification defines five modulating waveforms: *Sine, Square, Triangle, Ramp Up,* and *Ramp Down.*

Frequency – The number of modulating waveform cycles generated in one second.

Modulation Depth – The extent to which the modulating waveform affects the amplitude of the carrier waveform. This value is expressed as a percentage.

At the maximum peak of the modulating waveform, the amplitude of the output signal is equal to (100.0 + Modulation Depth) percent of the carrier signal amplitude. At the minimum peak of the modulating waveform, the amplitude of the output signal is equal to (100.0 – Modulation Depth) percent of the carrier signal amplitude. At a modulation depth of 0 percent, the modulating waveform has no affect on the carrier waveform. At a modulation depth of 100 percent, the amplitude of the output signal varies between 0.0V and twice the amplitude of the carrier signal.

The following diagrams illustrate the effect of amplitude modulation on a carrier signal, and the effect on the output signal of varying the modulation depth:
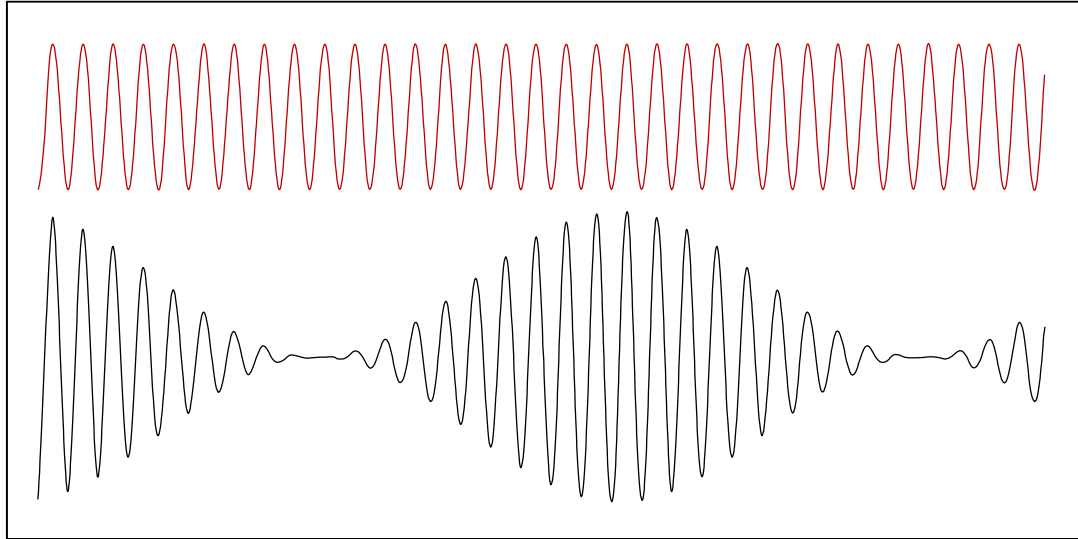
Figure 18-1. One kHz Carrier Sine Wave and the result of modulating the carrier waveform with a sine waveform at 100 percent depth.

18.2 IviFgenModulateAM Attributes

The IviFgenModulateAM capability group defines the following attributes:

- AM Enabled

- AM Internal Depth

- AM Internal Frequency

- AM Internal Waveform

- AM Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

18.2.1 AM Enabled

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViBoolean	R/W	Channel	None	Configure AM Enabled

.NET Property Using Channel-Dependent Get and Set methods

```
Boolean AM.GetEnabled (String channelName)

void AM.SetEnabled (String channelName, Boolean enabled)
```

COM Property Name

```
AM.Enabled(BSTR ChannelName)
```

C Constant Name

```
IVIFGEN_ATTR_AM_ENABLED
```

Description

Specifies whether the function generator applies amplitude modulation to the signal that the function generator produces with the IviFgenStdFunc, IviFgenArbWfm, or IviFgenArbSeq capability groups. If set to True, the function generator applies amplitude modulation to the output signal. If set to False, the function generator does not apply amplitude modulation to the output signal.

Compliance Notes

1. Instrument drivers shall support the values True and 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 these methods.

18.2.2 AM Internal Depth

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N A	None	Configure AM Internal

.NET Property Name

> AM.InternalDepth

COM Property Name

> AM.InternalDepth

C Constant Name

> IVIFGEN_ATTR_AM_INTERNAL_DEPTH

Description

> Specifies the extent of modulation the function generator applies to the carrier waveform when the AM Source attribute is set to AM Internal. The unit is percentage.

> This attribute affects the behavior of the instrument only when the AM Source attribute is set to AM Internal.

.NET Exceptions

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

18.2.3 AM Internal Frequency

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N A	None	Configure AM Internal

.NET Property Name

AM.InternalFrequency

COM Property Name

AM.InternalFrequency

C Constant Name

IVIFGEN_ATTR_AM_INTERNAL_FREQUENCY

Description

Specifies the frequency of the internal modulating waveform source. The units are Hertz.

This attribute affects the behavior of the instrument only when the AM Source attribute is set to AM Internal.

.NET Exceptions

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

18.2.4 AM Internal Waveform

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	N A	None	Configure AM Internal

.NET Property Name

> AM.InternalWaveformFunction

.NET Enumeration Name

> Ivi.Fgen.StandardWaveform

COM Property Name

> AM.InternalWaveform

COM Enumeration Name

> IviFgenAMInternalWaveformEnum

C Constant Name

> IVIFGEN_ATTR_AM_INTERNAL_WAVEFORM

Description

> Specifies the waveform of the internal modulating waveform source.

> This attribute affects the behavior of the instrument only when the AM Source attribute is set to AM Internal.

Defined Values

Name	Description		
		Language	*Identifier*
AM Internal Sine	The function generator uses a sinusoid waveform as the modulating waveform.		
	.NET	StandardWaveform.Sine	
	C	IVIFGEN_VAL_AM_INTERNAL_SINE	
	COM	IviFgenAMInternalWaveformSine	
AM Internal Square	The function generator uses a square waveform as the modulating waveform.		
	.NET	StandardWaveform.Square	
	C	IVIFGEN_VAL_AM_INTERNAL_SQUARE	
	COM	IviFgenAMInternalWaveformSquare	
AM Internal Triangle	The function generator uses a triangle waveform as the modulating waveform.		
	.NET	StandardWaveform.Triangle	
	C	IVIFGEN_VAL_AM_INTERNAL_TRIANGLE	

	COM	`IviFgenAMInternalWaveformTriangle`
AM Internal Ramp Up	The function generator uses a positive ramp waveform as the modulating waveform.	
	.NET	`StandardWaveform.RampUp`
	C	`IVIFGEN_VAL_AM_INTERNAL_RAMP_UP`
	COM	`IviFgenAMInternalWaveformRampUp`
AM Internal Ramp Down	The function generator uses a negative ramp waveform as the modulating waveform.	
	.NET	`StandardWaveform.RampDown`
	C	`IVIFGEN_VAL_AM_INTERNAL_RAMP_DOWN`
	COM	`IviFgenAMInternalWaveformRampDown`

.NET 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.

If a program attempts to set this property to `Ivi.Fgen.StandardWaveform.DC,` the driver shall throw a Value Not Supported (`System.NotSupportedException`) exception.

Compliance Notes

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

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

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

Refer to Section 30, *IviFgen Attribute Value Definitions*, for the definitions of AM Internal Waveform Specific Ext Base, `IVIFGEN_VAL_AM_INTERNAL_WFM_SPECIFIC_EXT_BASE` and `IVIFGEN_VAL_AM_INTERNAL_WFM_CLASS_EXT_BASE`.

18.2.5 AM Source

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure AM Source

.NET Property Using Channel-Dependent Get and Set methods

```
Ivi.Fgen.AMSource AM.GetSource (String channelName)

void AM.SetSource (String channelName, Ivi.Fgen.AMSource source)
```

.NET Enumeration Name

```
Ivi.Fgen.AMSource
```

COM Property Name

```
AM.Source(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenAMSourceEnum
```

C Constant Name

```
IVIFGEN_ATTR_AM_SOURCE
```

Description

Specifies the source of the signal that the function generator uses as the modulating waveform.

This attribute affects instrument behavior only when the AM Enabled attribute is set to True.

Defined Values

Name	Description		
		Language	*Identifier*
AM Internal	The function generator uses an internally generated waveform as the modulating waveform. You use the AM Internal Depth, AM Internal Waveform, and AM Internal Frequency attributes to configure the internally generated modulating waveform.		
		.NET	`AMSource.Internal`
		C	`IVIFGEN_VAL_AM_INTERNAL`
		COM	`IviFgenAMSourceInternal`
AM External	The function generator uses a waveform from an external source as the modulating waveform.		
		.NET	`AMSource.External`
		C	`IVIFGEN_VAL_AM_EXTERNAL`

| | | COM | `IviFgenAMSourceExternal` |

.NET Exceptions

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

Compliance Notes

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

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

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

Refer to Section 30, *IviFgen Attribute Value Definitions*, for the definitions of AM Source Specific Ext Base, `IVIFGEN_VAL_AM_SOURCE_SPECIFIC_EXT_BASE` and `IVIFGEN_VAL_AM_SOURCE_CLASS_EXT_BASE`.

18.3 IviFgenModulateAM Functions

The IviFgenModulateAM extension defines the following functions:

- Configure AM Enabled (IVI-C Only)

- Configure AM Internal

- Configure AM Source (IVI-C Only)

This section describes the behavior and requirements of each function.

18.3.1 Configure AM Enabled (IVI-C Only)

Description

Configures whether the function generator applies amplitude modulation to a channel.

.NET Method Prototype

```
N/A
```

(Use AM.SetEnabled. See the AM Enabled attribute section.)

COM Method Prototype

```
N/A
```

(Use the `AM.Enabled` property.)

C Prototype

```
ViStatus IviFgen_ConfigureAMEnabled (ViSession Vi,
                                     ViConstString ChannelName,
                                     ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to enable amplitude modulation.	ViConstString
Enabled	Specifies whether the function generator applies ampitude modulation to a channel. The driver uses this value to set the AM Enabled attribute. See the attribute description for more details.	ViBoolean

Return Values (C)

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

18.3.2 Configure AM Internal

Description

Configures the attributes that control the function generator's internal amplitude modulating waveform source. These attributes are the modulation depth, waveform, and frequency.

.NET Method Prototype

```
void AM.ConfigureInternal (Double depth,
                           StandardWaveform waveformFunction,
                           Double frequency);
```

COM Method Prototype

```
HRESULT AM.ConfigureInternal ([in] DOUBLE Depth,
                              [in] IviFgenAMInternalWaveformEnum Waveform,
                              [in] DOUBLE Frequency);
```

C Prototype

```
ViStatus IviFgen_ConfigureAMInternal (ViSession Vi,
                                      ViReal64 Depth,
                                      ViInt32 Waveform,
                                      ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Depth	Specifies the internal modulation depth. The driver uses this value to set the AM Internal Depth attribute. See the attribute description for more details.	ViReal64
Waveform (C/COM) waveformFunction (.NET)	Specifies the waveform the function generator uses for the internal modulating waveform source. The driver uses this value to set the AM Internal Waveform attribute. See the attribute description for more details.	ViInt32
Frequency	Specifies the frequency of the internal modulating waveform source. The driver uses this value to set the AM Internal Frequency attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

.NET Exceptions

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

18.3.3 Configure AM Source (IVI-C Only)

Description

Configures the source of the AM modulating waveform.

.NET Method Prototype

```
N/A
```

(Use AM.SetSource. See the AM Source attribute section.)

COM Method Prototype

```
N/A
```

(Use the AM.Source property.)

C Prototype

```
ViStatus IviFgen_ConfigureAMSource (ViSession Vi,
                                    ViConstString ChannelName
                                    ViInt32 Source);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the AM modulating waveform source.	ViConstString
Source	Specifies the source of the modulating waveform. The driver uses this value to set the AM Source attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

18.4 IviFgenModulateAM Behavior Model

The IviFgenModulateAM Extension Group follows the behavior model of the IviFgenBase capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenModulateAM settings.

18.5 IviFgenModulateAM Compliance Notes

1. If an instrument driver does not implement the value AM Internal for the AM Source attributes, it need not implement the AM Internal Waveform, AM Internal Depth, and AM Internal Frequency attributes, or the Configure AM Internal function.

19 IviFgenModulateFM Extension Group

19.1 IviFgenModulateFM Overview

The IviFgenModulateFM Extension Group supports function generators that can apply frequency modulation to an output signal. The user can enable or disable frequency modulation, and specify the source of the modulating waveform. If the function generator supports an internal modulating waveform source, the user can specify the waveform type, frequency, and peak frequency deviation.

Frequency modulation is accomplished by varying the frequency of a carrier waveform according to the amplitude of a modulating waveform. The general equation for a frequency modulated waveform is,

$$FM(t) = C[t + (M(t))],$$

where C(t) is the carrier waveform, M(t) is the modulating waveform, and FM(t) is the frequency modulated signal.

This specification provides modulating waveform property definitions that must be followed when developing specific instrument drivers. The carrier waveform is defined as the waveform the function generator produces without any modulation. You configure the carrier waveform with the IviFgenStdFunc, IviFgenArbWfm, or IviFgenArbSeq capability groups.   The modulating waveform is defined by the following properties:

Waveform Type – The overall "shape" of one period of the modulating waveform. This specification defines five modulation waveform types: *Sine, Square, Triangle, Ramp Up,* and *Ramp Down*.

Frequency – The number of modulating waveform cycles generated in one second.

Peak Frequency Deviation – The variation of frequency the modulating waveform applies to the carrier waveform. This value is expressed in hertz. At 0 hertz deviation, the modulating waveform has no effect on the carrier waveform. As frequency deviation increases, the frequency variation in the modulated waveform increases.

At the maximum peak of the modulating waveform, the frequency of the output signal is equal to the frequency of the carrier signal plus the Peak Frequency Deviation. At the minimum peak of the modulating waveform, the frequency of the output signal is equal to the frequency of the carrier signal minus the Peak Frequency Deviation.

The following diagrams illustrate the effect of frequency modulation on a carrier signal, and the effect on the output signal of varying the peak frequency deviation.
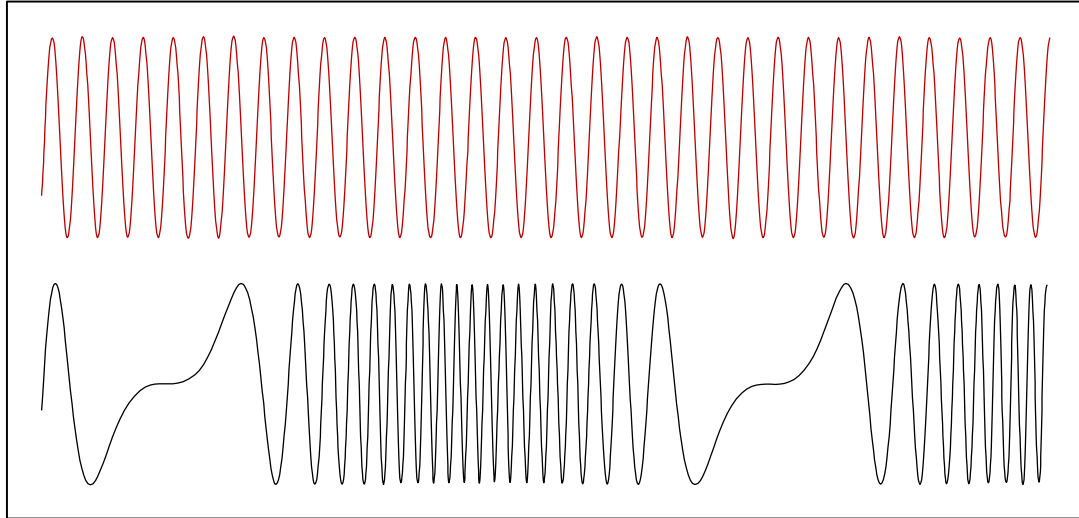
Figure 19-1. 1 kHz Carrier Sine Wave and a Frequency Modulated Wave at 500 Hz Peak Deviation

19.2 IviFgenModulateFM Attributes

The IviFgenModulateFM capability group defines the following attributes:

- FM Enabled

- FM Internal Deviation

- FM Internal Frequency

- FM Internal Wavefrom

- FM Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

19.2.1 FM Enabled

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViBoolean	R/W	Channel	None	Configure FM Enabled

.NET Property Using Channel-Dependent Get and Set methods

    Boolean FM.GetEnabled (String channelName)

    void FM.SetEnabled (String channelName, Boolean enabled)

COM Property Name

    FM.Enabled(BSTR ChannelName)

C Constant Name

    IVIFGEN_ATTR_FM_ENABLED

Description

Specifies whether the function generator applies amplitude modulation to the carrier waveform. If set to True, the function generator applies frequency modulation to the output signal. If set to False, the function generator does not apply frequency modulation to the output signal.

Compliance Notes

1.   Specific drivers shall support the values True and 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 these methods.

19.2.2 FM Internal Deviation

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N/A	None	Configure FM Internal

.NET Property Name

FM.InternalDeviation

COM Property Name

FM.InternalDeviation

C Constant Name

IVIFGEN_ATTR_FM_INTERNAL_DEVIATION

Description

Specifies the maximum frequency deviation, in Hertz, that the function generator applies to the carrier waveform when the FM Source attribute is set to FM Internal.

This attribute affects the behavior of the instrument only when the FM Source attribute is set to FM Internal.

.NET Exceptions

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

19.2.3 FM Internal Frequency

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	N/A	None	Configure FM Internal

.NET Property Name

> FM.InternalFrequency

COM Property Name

> FM.InternalFrequency

C Constant Name

> IVIFGEN_ATTR_FM_INTERNAL_FREQUENCY

Description

> Specifies the frequency of the internal modulating waveform source. The units are hertz.

> This attribute affects the behavior of the instrument only when the FM Source attribute is set to FM Internal.

.NET Exceptions

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

19.2.4 FM Internal Waveform

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	N A	None	Configure FM Internal

.NET Property Name

FM.InternalWaveformFunction

.NET Enumeration Name

Ivi.Fgen.StandardWaveform

COM Property Name

FM.InternalWaveform

COM Enumeration Name

IviFgenFMInternalWaveformEnum

C Constant Name

IVIFGEN_ATTR_FM_INTERNAL_WAVEFORM

Description

Specifies the waveform of the internal modulating waveform source.

This attribute affects the behavior of the instrument only when the FM Source attribute is set to FM Internal.

Defined Values

Name	Description	
	Language	*Identifier*
FM Internal Sine	The function generator uses a sinusoid waveform as the modulating waveform.	
	.NET	StandardWaveform.Sine
	C	IVIFGEN_VAL_FM_INTERNAL_SINE
	COM	IviFgenFMInternalWaveformSine
FM Internal Square	The function generator uses a square waveform as the modulating waveform.	
	.NET	StandardWaveform.Square
	C	IVIFGEN_VAL_FM_INTERNAL_SQUARE
	COM	IviFgenFMInternalWaveformSquare
FM Internal Triangle	The function generator uses a triangle waveform as the modulating waveform.	

	.NET	StandardWaveform.Triangle
	C	IVIFGEN_VAL_FM_INTERNAL_TRIANGLE
	COM	IviFgenFMInternalWaveformTriangle
FM Internal Ramp Up	The function generator uses a positive ramp waveform as the modulating waveform.	
	.NET	StandardWaveform.RampUp
	C	IVIFGEN_VAL_FM_INTERNAL_RAMP_UP
	COM	IviFgenFMInternalWaveformRampUp
FM Internal Ramp Down	The function generator uses a negative ramp waveform as the modulating waveform.	
	.NET	StandardWaveform.RampDown
	C	IVIFGEN_VAL_FM_INTERNAL_RAMP_DOWN
	COM	IviFgenFMInternalWaveformRampDown

.NET 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.

If a program attempts to set this property to Ivi.Fgen.StandardWaveform.DC, the driver shall throw a Value Not Supported (System.NotSupportedException) exception.

Compliance Notes

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

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

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

Refer to Section 30, *IviFgen Attribute Value Definitions*, for the definitions of FM Internal Waveform, IVIFGEN_VAL_FM_INTERNAL_WFM_SPECIFIC_EXT_BASE and IVIFGEN_VAL_FM_INTERNAL_WFM_CLASS_EXT_BASE.

19.2.5 FM Source

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	Configure FM Source

.NET Property Using Channel-Dependent Get and Set methods

```
FMSource FM.GetSource (String channelName)

void FM.SetSource (String channelName, FMSource source)
```

.NET Enumeration Name

```
Ivi.Fgen.FMSource
```

COM Property Name

```
FM.Source(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenFMSourceEnum
```

C Constant Name

```
IVIFGEN_ATTR_FM_SOURCE
```

Description

Specifies the source of the signal that the function generator uses as the modulating waveform.

This attribute affects instrument behavior only when the FM Enabled attribute is set to True.

Defined Values

Name	Description	
	Language	*Identifier*
FM Internal	The function generator uses an internally generated waveform as the modulating waveform. You use the FM Internal Deviation, FM Internal Waveform, and FM Internal Frequency attributes to configure the internally generated modulating waveform.	
	.NET	Ivi.Fgen.FMSource.Internal
	C	IVIFGEN_VAL_FM_INTERNAL
	COM	IviFgenFMSourceInternal
FM External	The function generator uses a waveform from an external source as the modulating waveform.	
	.NET	Ivi.Fgen.FMSource.External
	C	IVIFGEN_VAL_FM_EXTERNAL

	COM	IviFgenFMSourceExternal

.NET Exceptions

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

Compliance Notes

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

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

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

Refer to Section 30, *IviFgen Attribute Value Definitions*, for the definitions of FM Source Specific Ext Base, IVIFGEN_VAL_FM_SOURCE_SPECIFIC_EXT_BASE and IVIFGEN_VAL_FM_SOURCE_CLASS_EXT_BASE.

19.3 IviFgenModulateFM Functions

The IviFgenModulateFM extension defines the following functions:

- Configure FM Enabled (IVI-C Only)

- Configure FM Internal

- Configure FM Source (IVI-C Only)

This section describes the behavior and requirements of each function.

19.3.1 Configure FM Enabled (IVI-C Only)

Description

Configures whether the function generator applies frequency modulation to a channel.

.NET Method Prototype

```
N/A
```

(Use FM.SetEnabled. See the FM Enabled attribute section.)

COM Method Prototype

```
N/A
```

(Use the FM.Enabled property.)

C Prototype

```
ViStatus IviFgen_ConfigureFMEnabled (ViSession Vi,
                                     ViConstString ChannelName,
                                     ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to enable amplitude modulation.	ViConstString
Enabled	Specifies whether the function generator applies frequency modulation to a channel. The driver uses this value to set the FM Enabled attribute. See the attribute description for more details.	ViBoolean

Return Values (C)

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

19.3.2 Configure FM Internal

Description

Configures the attributes that control the function generator's internal frequency modulating waveform source. These attributes are the modulation peak deviation, waveform, and frequency.

.NET Method Prototype

```
void FM.ConfigureInternal (Double deviation,
                           StandardWaveform waveformFunction,
                           Double frequency);
```

COM Method Prototype

```
HRESULT FM.ConfigureInternal ([in] double Deviation,
                              [in] IviFgenFMInternalWaveformEnum Waveform,
                              [in] double Frequency);
```

C Prototype

```
ViStatus IviFgen_ConfigureFMInternal (ViSession Vi,
                                      ViReal64 Deviation,
                                      ViInt32 Waveform,
                                      ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Deviation	Specifies the internal modulation deviation. The driver uses this value to set the FM Internal Deviation attribute. See the attribute description for more details.	ViReal64
Waveform (C/COM) waveformFunction (.NET)	Specifies the waveform the function generator uses for the internal modulating waveform source. The driver uses this value to set the FM Internal Waveform attribute. See the attribute description for more details.	ViInt32
Frequency	Specifies the frequency of the internal modulating waveform source. The driver uses this value to set the FM Internal Frequency attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

.NET Exceptions

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

19.3.3 Configure FM Source (IVI-C Only)

Description

Configures the source of the FM modulating waveform.

.NET Method Prototype

```
N/A
```

(Use FM.SetSource. See the FM Source attribute section.)

COM Method Prototype

```
N/A
```

(Use the FM.Source property.)

C Prototype

```
ViStatus IviFgen_ConfigureFMSource (ViSession Vi,
                                    ViConstString ChannelName
                                    ViInt32 Source);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to configure the FM modulating waveform source..	ViConstString
Source	Specifies the source of the modulating waveform. The driver uses this value to set the FM Source attribute. See the attribute description for more details.	ViInt32

Return Values (C)

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

19.4 IviFgenModulateFM Behavior Model

The IviFgenModulateFM Extension Group follows the behavior model of the IviFgenBase capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenModulateFM settings.

19.5 IviFgenModulateFM Compliance Notes

1. If a specific driver does not implement the value FM Internal for the FM Source attribute, it need not implement the FM Internal Waveform, FM Internal Deviation, and FM Internal Frequency attributes, or the Configure FM Internal function.

20 IviFgenSampleClock Extension Group

20.1 IviFgenSampleClock Overview

The IviFgenSampleClock extension group supports arbitrary waveform generators with the ability to use (or provide) an external sample clock. Note that when using an external sample clock, the Arbitrary Sample Rate attribute must be set to the corresponding frequency of the external sample clock.

20.2 IviFgenSampleClock Attributes

The IviFgenSampleClock extension group defines the following attributes:

- Sample Clock Source

- Sample Clock Output Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

20.2.1 Sample Clock Source

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

.NET Property Name

> SampleClock.Source

.NET Enumeration Name

> SampleClockSource

COM Property Name

> SampleClock.Source

COM Enumeration Name

> IviFgenSampleClockSourceEnum

C Constant Name

> IVIFGEN_ATTR_SAMPLE_CLOCK_SOURCE

Description

> Specifies the clock used for the waveform generation. Note that when using an external sample clock, the Arbitrary Sample Rate attribute must be set to the corresponding frequency of the external sample clock.

Defined Values

Name	*Description*		
	Language	*Identifier*	
Internal	The internal sample clock is used.		
	.NET	Internal	
	C	IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_INTERNAL	
	COM	IviFgenSampleClockSourceInternal	
External	An external sample clock is used.		
	.NET	External	
	C	IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_EXTERNAL	
	COM	IviFgenSampleClockSourceExternal	

.NET Exceptions

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

20.2.2 Sample Clock Output Enabled

Data Type	Access	Applies To	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Sample Clock Output Enabled (IVI-C only)

.NET Property Name

SampleClock.OutputEnabled

COM Property Name

SampleClock.OutputEnabled

COM Enumeration Name

N A

C Constant Name

IVIFGEN_ATTR_SAMPLE_CLOCK_OUTPUT_ENABLED

Description

Specifies whether or not the sample clock appears at the sample clock output of the generator.

.NET Exceptions

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

20.3 IviFgenSampleClock Functions

The IviFgenSampleClock extension group defines the following functions:

- Configure Sample Clock (IVI-C only)
- Configure Sample Clock Output Enabled (IVI-C only)

This section describes the behavior and requirements of this function.

20.3.1 Configure Sample Clock (IVI-C Only)

Description

Configures the generator's sample clock.

.NET Method Prototype

(N/A)

(use the SampleClock.Source property)

COM Method Prototype

(N/A)

(use the SampleClock.Source property)

C Prototype

ViStatus IviFgen_ConfigureSampleClock (ViSession Vi, ViInt32 Source);

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Source	Specifies the source of the sample clock signal. The driver uses this value to set the Sample Clock Source attribute. See the attribute description for more details.	ViInt32

Defined Values for the Source Parameter

Name	*Description*	
	Language	*Identifier*
Internal	The internal sample clock is used.	
	C	IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_INTERNAL
	COM	IviFgenSampleClockSourceInternal
External	An external sample clock is used.	
	C	IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_EXTERNAL
	COM	IviFgenSampleClockSourceExternal

Return Values

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

20.3.2 Configure Sample Clock Output Enabled (IVI-C Only)

Description

Configures whether or not the sample clock appears at the sample clock output of the generator.

.NET Method Prototype

(N A)

(use the SampleClock.OutputEnabled property)

COM Method Prototype

(N A)

(use the SampleClock.OutputEnabled property)

C Prototype

ViStatus IviFgen_ConfigureSampleClockOutputEnabled(ViSession Vi,
                              ViBoolean Enabled);

Parameters

Inputs	Description	Base Type
Vi	Instrument handle.	ViSession
Enabled	Specifies whether or not the sample clock appears at the sample clock output of the generator. This value sets the Sample Clock Output Enabled attribute. See the attribute description for more details.	ViBoolean

Return Values

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

20.4 IviFgenSampleClock Behavior Model

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

20.5 IviFgenSampleClock Compliance Notes

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

21 IviFgenTerminalConfiguration Extension Group

21.1 IviFgenTerminalConfiguration Overview

The IviFgenTerminalConfiguration extension group supports function generators with the ability to specify whether the output terminals are single-ended or differential.

21.2 IviFgenTerminalConfiguration Attributes

The IviFgenTerminalConfiguration extension group defines the following attributes:

- Terminal Configuration

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

21.2.1 Terminal Configuration

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Channel	None	N/A

.NET Property Using Channel-Dependent Get and Set methods

```
TerminalConfiguration Output.GetTerminalConfiguration(String channelName)

void Output.SetTerminalConfiguration(String channelName,
                                     TerminalConfiguration configuration)
```

.NET Enumeration Name

```
TerminalConfiguration
```

COM Property Name

```
Output.TerminalConfiguration(BSTR ChannelName)
```

COM Enumeration Name

```
IviFgenTerminalConfigurationEnum
```

C Constant Name

```
IVIFGEN_ATTR_OUTPUT_TERMINAL_CONFIGURATION
```

Description

Determines whether the generator will run in single-ended or differential mode, and whether the output gain and offset values will be analyzed based on single-ended or differential operation.

Defined Values

Name	Description	
	Language	*Identifier*
Terminal Configuration Single Ended	Output terminal is single-ended.	
	.NET	SingleEnded
	C	IVIFGEN_VAL_TERMINAL_CONFIGURATION_SINGLE_ENDED
	COM	IviFgenTerminalConfigurationSingleEnded
Terminal Configuration Differential	Output terminal is differential.	
	.NET	Differential
	C	IVIFGEN_VAL_TERMINAL_CONFIGURATION_DIFFERENTIAL
	COM	IviFgenTerminalConfigurationDifferential

.NET Exceptions

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

21.3 IviFgenTerminalConfiguration Functions

The IviFgenTerminalConfiguration extension group defines no functions.

21.4 IviFgenTerminalConfiguration Behavior Model

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

21.5 IviFgenTerminalConfiguration Compliance Notes

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

22 IviFgenArbWfmSize64 Extension Group (IVI-C and IVI-COM Only)

22.1 IviFgenArbWfmSize64 Overview

The IviFgenArbWfmSize64 Extension Group supports function generators capable of producing user-defined arbitrary waveforms of sizes larger than can be represented by a 32-bit integer.

22.2 IviFgenArbWfmSize64 Attributes

The IviFgenArbWfmSize64 extension group defines the following attributes:

- Waveform Size Max 64
- Waveform Size Min 64

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

22.2.1 Waveform Size Max 64 (IVI-C and IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt64	RO	N A	None	Query Arbitrary Waveform Capabilities 64

.NET Property Name

N/A

(Use the `Arbitrary.Waveform.SizeMax` property.)

COM Property Name

`Arbitrary.Waveform.SizeMax64`

C Constant Name

`IVIFGEN_ATTR_MAX_WAVEFORM_SIZE64`

Description

Returns the maximum number of points the function generator allows in an arbitrary waveform.

.NET Exceptions

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

22.2.2 Waveform Size Min 64 (IVI-C and IVI-COM Only)

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt64	RO	N A	None	Query Arbitrary Waveform Capabilities 64

.NET Property Name

N/A

(Use the `Arbitrary.Waveform.SizeMin` property.)

COM Property Name

`Arbitrary.Waveform.SizeMin64`

C Constant Name

`IVIFGEN_ATTR_MIN_WAVEFORM_SIZE64`

Description

Returns the minimum number of points the function generator allows in an arbitrary waveform.

.NET Exceptions

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

22.3 IviFgenArbWfmSize64 Functions

The IviFgenArbWfmSize64 extension group defines the following function:

- Query Arbitrary Waveform Capabilities 64

This section describes the behavior and requirements of this function.

22.3.1 Query Arbitrary Waveform Capabilities 64 (IVI-C Only)

Description

Returns the attributes of the function generator that are related to creating arbitrary waveforms. These attributes are the maximum number of waveforms, waveform quantum, minimum waveform size, and maximum waveform size.

.NET Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.NumberWaveformsMax, Quantum, SizeMin,` and `SizeMax` properties.)

COM Method Prototype

```
N/A
```

(Use the `Arbitrary.Waveform.NumberWaveformsMax, Quantum, SizeMin64,` and `SizeMax64` properties.)

C Prototype

```
ViStatus IviFgen_QueryArbWfmCapabilities64 (ViSession Vi,
                                            ViInt32 *MaxNumWfms,
                                            ViInt32 *WfmQuantum,
                                            ViInt64 *MinWfmSize,
                                            ViInt64 *MaxWfmSize);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Outputs	Description	Base Type
MaxNumWfms	Returns the maximum number of arbitrary waveforms that the function generator allows. The driver obtains this value from the Number Waveforms Max attribute. See the attribute description for more details.	ViInt32
WfmQuantum	Returns the quantum value the function generator uses. The driver returns this value from the Waveform Quantum attribute. See the attribute description for more details.	ViInt32
MinWfmSize	Returns the minimum number of points the function generator allows in a waveform. The driver obtains this value from the Waveform Size Min 64 attribute. See the attribute description for more details.	ViInt64
MaxWfmSize	Returns the maximum number of points the function generator allows in a waveform. The driver obtains this value from the Waveform Size Max 64 attribute. See the attribute description for more details.	ViInt64

Return Values (C)

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

22.4 IviFgenArbWfmSize64 Behavior Model

The IviFgenArbWfmSize64 extension group follows the same behavior model as the IviFgenArbWfm capability group described in Section 4.4, *IviFgenBase Behavior Model.*

22.5 IviFgenArbWfmSize64 Compliance Notes

For a specific driver to comply with the IviFgenArbWfmSize64 extension, it shall be compliant with the IviFgenArbWfm capability group and it shall implement all of the attributes and functions listed in this section.

23 IviFgenArbChannelWfm Extension Group

23.1 IviFgenArbChannelWfm Overview

The IviFgenArbChannelWfm Extension Group supports single channel and multichannel function generators capable of producing user-defined arbitrary waveforms for specific output channels. The IviFgenArbChannelWfm extension group includes functions for creating, configuring, and generating arbitrary waveforms.

23.2 IviFgenArbChannelWfm Attributes

The IviFgenArbChannelWfm extension group defines no attributes.

23.3 IviFgenArbChannelWfm Functions

The IviFgenArbChannelWfm extension group defines the following function:

- Create Channel Arbitrary Waveform

This section describes the behavior and requirements of this function.

23.3.1 Create Channel Arbitrary Waveform

Description

Creates a channel-specific arbitrary waveform and returns a handle that identifies that waveform. You pass a waveform handle as the `waveformHandle` parameter of the Configure Arbitrary Waveform function to produce that waveform. You also use the handles this function returns to create a sequence of arbitrary waveforms with the Create Arbitrary Sequence function.

The waveform must be normalized to values between -1.0 and +1.0, where -1.0 and +1.0 correspond to full-scale before gain and offset are applied. Drivers are not required to check every value in the waveform for values outside of this range. For IWaveform objects, values returned by IWaveform.GetScaled(), or equivalent values, shall be used, and -1.0 and +1.0 correspond to negative and positive full-scale outputs.

If the instrument has multiple channels, it is possible to create multi-channel waveforms: the channel names are passed as a comma-separated list of channel names, and the waveform arrays are concatenated into a single array. In this case, all waveforms must be of the same length.

If the function generator cannot store any more arbitrary waveforms, this function returns the error No Waveforms Available.

For IVI.NET, the Arbitrary Sample Rate is set to the inverse of the Waveform IntervalPerPoint. Observation: Setting the Arbitrary Sample Rate after calling Create Channel Arbitrary Waveform (either by directly setting the attribute or by calling Create Channel Arbitrary Waveform again) will overwrite the value of Sample Rate established by the initial Create Channel Arbitrary Waveform call.

.NET Method Prototype

```
Int32 Arbitrary.Waveform.CreateChannelWaveform (
                                 String channelName,
                                 IWaveform<T> waveform);
```

COM Method Prototype

```
HRESULT Arbitrary.Waveform.CreateChannelWaveform (
                                 [in] BSTR ChannelName,
                                 [in] SAFEARRAY(double) *Data,
                                 [out,retval] LONG *Handle);
```

C Prototype

```
ViStatus IviFgen_CreateChannelArbWaveform (ViSession Vi,
                                 ViConstString ChannelName,
                                 ViInt64 Size,
                                 ViReal64 Data[],
                                 ViInt32 *Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to create the arbitrary waveform, or a comma-separated list of channel names when creating a multi-channel waveform.	ViConstString

Size	Specifies the number of points in the Data array.	ViInt64
Data (C/COM)	Specifies the array of data to use for the new arbitrary waveform. The array must have at least as many elements as the value in the Size parameter. The array's elements must be normalized between -1.00 and +1.00. When creating a multi-channel waveform, this array is the concatenation of the waveform arrays for each channel. In this case, all waveforms must be of the same length.	ViReal64[]
waveform (.NET)	Specifies the waveform to use for the new arbitrary waveform. When creating a multi-channel waveform, this waveform object is the concatenation of the waveform for each channel. In this case, all waveforms must be of the same length.	IWaveform<T>

Outputs	Description	Base Type
Handle (C/COM) Return Value (.NET)	Returns the handle that identifies the new arbitrary waveform.	ViInt32

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 Codes	Description
No Waveforms Available	The function generator's arbitrary waveform memory is full.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method. The table below specifies additional class-defined exceptions for this method.

Completion Codes	Description
NoWaveformsAvailableException	The function generator's arbitrary waveform memory is full.

23.4 IviFgenArbChannelWfm Behavior Model

The IviFgenArbChannelWfm extension group follows the same behavior model as the IviFgenArbWfm capability group described in Section 4.4, *IviFgenBase Behavior Model.*

23.5 IviFgenArbChannelWfm Compliance Notes

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

24 IviFgenArbWfmBinary Extension Group

24.1 IviFgenArbWfmBinary Overview

The IviFgenArbWfmBinary Extension Group supports multichannel function generators capable of producing user-defined arbitrary waveforms that can be specified in binary format. The IviFgenArbWfmBinary extension group includes functions for creating, configuring, and generating arbitrary waveforms.

24.2 IviFgenArbWfmBinary Attributes

The IviFgenArbWfmBinary extension group defines the following attributes:

- Binary Alignment
- Sample Bit Resolution

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

24.2.1 Binary Alignment

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N A	None	N/A

.NET Property Name

      Arbitrary.BinaryAlignment

.NET Enumeration Name

      BinaryAlignment

COM Property Name

      Arbitrary.BinaryAlignment

COM Enumeration Name

      IviFgenBinaryAlignmentEnum

C Constant Name

      IVIFGEN_ATTR_BINARY_ALIGNMENT

Description

Identifies whether the arbitrary waveform generator treats binary data provided to the Create Channel Arbitrary Waveform Int16 or Create Channel Arbitrary Waveform Int32 functions as left-justified or right-justified. Binary Alignment is only relevant if the generator supports bit-depths less than the size of the binarydata type of the create waveform function being used. For a 16-bit or a 32-bit generator, this function can return either value.

Defined Values

Name	Description	
	Language	*Identifier*
Binary Alignment Left	The driver uses the left-most bit (most significant bit) of the 16-bit or 32-bit data as the most significant bit of the data.	
	.NET	Left
	C	IVIFGEN_VAL_BINARY_ALIGNMENT_LEFT
	COM	IviFgenBinaryAlignmentLeft
Binary Alignment Right	The driver uses the right-most bit (least significant bit) of the 16-bit or 32-bit data as the least significant bit of the data.	
	.NET	Right
	C	IVIFGEN_VAL_BINARY_ALIGNMENT_RIGHT
	COM	IviFgenBinaryAlignmentRight

.NET 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.

24.2.2 Sample Bit Resolution

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N A	None	N/A

.NET Property Name

Arbitrary.SampleBitResolution

COM Property Name

Arbitrary.SampleBitResolution

C Constant Name

IVIFGEN_ATTR_SAMPLE_BIT_RESOLUTION

Description

Returns the number of significant bits that the generator supports in an arbitrary waveform. Together with the binary alignment, this allows the user to know the range and resolution of the integers in the waveform.

.NET Exceptions

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

24.3 IviFgenArbWfmBinary Functions

The IviFgenArbWfmBinary extension group defines the following functions:

- Create Channel Arbitrary Waveform Int16

- Create Channel Arbitrary Waveform Int32

This section describes the behavior and requirements of these functions.

24.3.1 Create Channel Arbitrary Waveform Int16

Description

Creates a channel-specific arbitrary waveform and returns a handle that identifies that waveform. Data is passed in as 16-bit binary data. If the arbitrary waveform generator supports formats less than 16 bits, call the BinaryAlignment property to determine whether to left or right justify the data before passing it to this call. You pass a waveform handle as the `waveformHandle` parameter of the Configure Arbitrary Waveform function to produce that waveform. You also use the handles this function returns to create a sequence of arbitrary waveforms with the Create Arbitrary Sequence function.

If the instrument has multiple channels, it is possible to create multi-channel waveforms: the channel names are passed as a comma-separated list of channel names, and the waveform arrays are concatenated into a single array. In this case, all waveforms must be of the same length.

If the function generator cannot store any more arbitrary waveforms, this function returns the error No Waveforms Available.

.NET Method Prototype

```
Int32 Arbitrary.Waveform.CreateChannelWaveform (
                                    String channelName,
                                    Int16[] waveform);
```

COM Method Prototype

```
HRESULT Arbitrary.Waveform.CreateChannelWaveformInt16 (
                                    [in] BSTR ChannelName,
                                    [in] SAFEARRAY(short) *Data,
                                    [out,retval] LONG *Handle);
```

C Prototype

```
ViStatus IviFgen_CreateChannelArbWaveform16 (ViSession Vi,
                                    ViConstString ChannelName,
                                    ViInt64 Size,
                                    ViInt16 Data[],
                                    ViInt32 *Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to createe the arbitrary waveform, or a comma-separated list of channel names when creating a multi-channel waveform.	ViConstString
Size	Specifies the number of points in the Data array.	ViInt64

Data (C/COM)	Specifies the array of data to use for the new arbitrary waveform. The array must have at least as many elements as the value in the Size parameter. When creating a multi-channel waveform, this array is the concatenation of the waveform arrays for each channel. In this case, all waveforms must be of the same length.	ViInt16[]
waveform (.NET)	Specifies the waveform to use for the new arbitrary waveform. When creating a multi-channel waveform, this waveform object is the concatenation of the waveform for each channel. In this case, all waveforms must be of the same length.	Int16[]

Outputs	Description	Base Type
Handle (C/COM) Return Value (.NET)	Returns the handle that identifies the new arbitrary waveform.	ViInt32

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 Codes	Description
No Waveforms Available	The function generator's arbitrary waveform memory is full.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method. The table below specifies additional class-defined exceptions for this method.

Completion Codes	Description
NoWaveformsAvailableException	The function generator's arbitrary waveform memory is full.

24.3.2 Create Channel Arbitrary Waveform Int32

Description

Creates a channel-specific arbitrary waveform and returns a handle that identifies that waveform. Data is passed in as 32-bit binary data. If the arbitrary waveform generator supports formats less than 32 bits, call the BinaryAlignment property to determine whether to left or right justify the data before passing it to this call. You pass a waveform handle as the `waveformHandle` parameter of the Configure Arbitrary Waveform function to produce that waveform. You also use the handles this function returns to create a sequence of arbitrary waveforms with the Create Arbitrary Sequence function.

If the instrument has multiple channels, it is possible to create multi-channel waveforms: the channel names are passed as a comma-separated list of channel names, and the waveform arrays are concatenated into a single array. In this case, all waveforms must be of the same length.

If the function generator cannot store any more arbitrary waveforms, this function returns the error No Waveforms Available.

.NET Method Prototype

```
Int32 Arbitrary.Waveform.CreateChannelWaveform (
                                  String channelName,
                                  Int32[] waveform);
```

COM Method Prototype

```
HRESULT Arbitrary.Waveform.CreateChannelWaveformInt32 (
                                  [in] BSTR ChannelName,
                                  [in] SAFEARRAY(long) *Data,
                                  [out,retval] LONG *Handle);
```

C Prototype

```
ViStatus IviFgen_CreateChannelArbWaveform32 (ViSession Vi,
                                  ViConstString ChannelName,
                                  ViInt64 Size,
                                  ViInt32 Data[],
                                  ViInt32 *Handle);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The channel on which to createe the arbitrary waveform, or a comma-separated list of channel names when creating a multi-channel waveform.	ViConstString
Size	Specifies the number of points in the Data array.	ViInt64

Data (C/COM)	Specifies the array of data to use for the new arbitrary waveform. The array must have at least as many elements as the value in the Size parameter. When creating a multi-channel waveform, this array is the concatenation of the waveform arrays for each channel. In this case, all waveforms must be of the same length.	ViInt32[]
waveform (.NET)	Specifies the waveform to use for the new arbitrary waveform. When creating a multi-channel waveform, this waveform object is the concatenation of the waveform for each channel. In this case, all waveforms must be of the same length.	Int32[]

Outputs	Description	Base Type
Handle (C/COM) Return Value (.NET)	Returns the handle that identifies the new arbitrary waveform.	ViInt32

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 Codes	Description
No Waveforms Available	The function generator's arbitrary waveform memory is full.

.NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method. The table below specifies additional class-defined exceptions for this method.

Completion Codes	Description
NoWaveformsAvailableException	The function generator's arbitrary waveform memory is full.

24.4 IviFgenArbWfmBinary Behavior Model

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

24.5 IviFgenArbWfmBinary Compliance Notes

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

25 IviFgenDataMarker Extension Group

25.1 IviFgenDataMarker Overview

The IviFgenDataMarker Extension Group supports arbitrary waveform generators that can output particular bits of waveform data as a marker output. The user can choose which bit (the 2nd bit for example) gets output, where the output goes, and various analog characteristics of the marker output. Data markers are repeated capabilities to allow the user to output multiple bits to different ouputs simultaneously. The user can also use the DataMask property to ensure that the data marker does not get output with the main waveform output.

Setting the Data Marker Destination attribute to a value other than None enables the data marker. To disable the data marker, set the Data Marker Destination to None.

25.2 IviFgenDataMarker Attributes

The IviFgenDataMarker capability group defines the following attributes:

- Data Marker Amplitude

- Data Marker Bit Position

- Data Marker Count

- Data Marker Delay

- Data Marker Destination

- Data Marker Item (IVI-COM only)

- Data Marker Name (IVI-COM only)

- Data Marker Polarity

- Data Marker Source Channel

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

25.2.1 Data Marker Amplitude

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	DataMarker	None	None

.NET Property Name

DataMarkers[].Amplitude

COM Property Name

DataMarkers.Item().Amplitude

C Constant Name

IVIFGEN_ATTR_DATAMARKER_AMPLITUDE

Description

Specifies the amplitude of the data marker output. The units are volts.

.NET Exceptions

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

25.2.2 Data Marker Bit Position

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	Data Marker	None	Configure Data Marker

.NET Property Name

DataMarkers[].BitPosition

COM Property Name

DataMarkers.Item().BitPosition

C Constant Name

IVIFGEN_ATTR_DATAMARKER_BIT_POSITION

Description

Specifies the bit position of the binary representation of the waveform data that will be output as a data marker. A value of 0 indicates the least significant bit.

.NET Exceptions

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

25.2.3 Data Marker Count

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

.NET Property Name

Data Markers.Count

This property is inherited from `IIviRepeatedCapabilityCollection`.

COM Property Name

Data Markers.Count

C Constant Name

IVIFGEN_ATTR_DATAMARKER_COUNT

Description

Returns the number of data markers.

.NET Exceptions

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

25.2.4 Data Marker Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	DataMarker	Down	None

.NET Property Name

DataMarkers[].Delay

COM Property Name

DataMarkers.Item().Delay

C Constant Name

IVIFGEN_ATTR_DATAMARKER_DELAY

Description

Specifies the amount of delay applied to the data marker output with respect to the analog data output. A value of zero indicates the marker is aligned with the analog data output.   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.

25.2.5 Data Marker Destination

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	DataMarker	None	Configure Data Marker

.NET Property Name

DataMarkers[].Destination

COM Property Name

DataMarkers.Item().Destination

C Constant Name

IVIFGEN_ATTR_DATAMARKER_DESTINATION

Description

Specifies the destination terminal for the data marker output.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

.NET Exceptions

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

Compliance Notes

1. The specific driver need not support all of the destination values defined in Section 30, *IviFgen Attribute Value Definitions*.

25.2.6 Data Marker Item (IVI-COM and IVI.NET Only)

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

.NET Property Name

DataMarkers[String name]

This indexer is inherited from IIviRepeatedCapabilityCollection. The name parameter uniquely identifies a particular data marker in the data markers collection.

COM Property Name

DataMarkers.Item([in] BSTR Name);

C Constant Name

N/A

Description

Data Marker Item uniquely identifies a data marker in the data markers collection. It returns an interface pointer which can be used to control the attributes and other functionality of that data marker.
The Item property takes a data marker name. If the user passes an invalid value for the data marker 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.

25.2.7 Data Marker Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

DataMarkers[].Name

This property is inherited from `IIviRepeatedCapabilityIdentification`.

COM Property Name

DataMarkers.Name ([in] LONG Index);

C Constant Name

N/A.

(Use the GetDataMarkerName function.)

Description

This attribute returns the repeated capability identifier defined by specific driver for the data marker that corresponds to the index that the user specifies. If the driver defines a qualified Data Marker name, this property returns the qualified name.

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

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

If the value that the user passes for the Index parameter is less than one or greater than the value of the Data Marker Count, the attribute returns an empty string for the value and returns an error.

.NET Exceptions

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

25.2.8 Data Marker Polarity

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

.NET Property Name

> DataMarkers[].Polarity

.NET Enumeration Name

> MarkerPolarity

COM Property Name

> DataMarkers.Item().Polarity

COM Enumeration Name

> IviFgenMarkerPolarityEnum

C Constant Name

> IVIFGEN_ATTR_DATAMARKER_POLARITY

Description

> Specifies the polarity of the data marker output.

Defined Values

Name	Description		
		Language	Identifier
Active High	When the data bit is 1 (high), the marker output level is high.		
		.NET	ActiveHigh
		C	IVIFGEN_VAL_MARKER_POLARITY_ACTIVE_HIGH
		COM	IviFgenMarkerPolarityActiveHigh
Active Low	When the data bit is 1 (high), the marker output level is low.		
		.NET	ActiveLow
		C	IVIFGEN_VAL_MARKER_POLARITY_ACTIVE_LOW
		COM	IviFgenMarkerPolarityActiveLow

.NET Exceptions

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

25.2.9 Data Marker Source Channel

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViString	R/W	DataMarker	None	Configure Data Marker

.NET Property Name

       DataMarkers[].SourceChannel

COM Property Name

       DataMarkers.Item().SourceChannel

C Constant Name

       IVIFGEN_ATTR_DATAMARKER_SOURCE_CHANNEL

Description

       Specifies the channel whose data bit will be output as a marker.

.NET Exceptions

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

25.3 IviFgenDataMarker Functions

The IviFgenDataMarker extension defines the following functions:

- Configure Data Marker

- Get Data Marker Name (IVI-C Only)

- Disable All Data Markers

This section describes the behavior and requirements of each function.

25.3.1 Configure Data Marker

Description

Configures some of the common data marker attributes.

.NET Method Prototype

```
void DataMarkers[].Configure (String sourceChannel,
                              Int32 bitPosition,
                              String destination);
```

COM Method Prototype

```
HRESULT DataMarkers.Configure ([in] BSTR Name,
                               [in] BSTR SourceChannel,
                               [in] LONG BitPosition,
                               [in] BSTR Destination);
```

C Prototype

```
ViStatus IviFgen_ConfigureDataMarker (ViSession Vi,
                                      ViConstString Name,
                                      ViConstString SourceChannel,
                                      ViInt32 BitPosition,
                                      ViConstString Destination);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Name	The name of the data marker.	ViConstString
SourceChannel	Specifies the channel whose data bit will be output as a marker. The driver uses this value to set the Source Channel attribute. Refer to the attribute description for more details.	ViConstString
BitPosition	Specifies the bit position of the waveform data bit that will be output as a data marker. A value of 0 indicates the least significant bit. The driver uses this value to set the Bit Position attribute. Refer to the attribute description for more details.	ViInt32
Destination	Specifies the destination terminal for the data marker. The driver uses this value to set the Destination attribute. Refer to the attribute description for more details.	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.

25.3.2 Get Data Marker Name (IVI-C Only)

Description

This function returns the repeated capability identifier for the data marker that corresponds to the one-based index that the user specifies. If the value that the user passes for the `Index` parameter is less than one or greater than the value of the DataMarker Count, the function returns an empty string in the `Name` parameter and returns an error.

.NET Method Prototype

N/A.

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

COM Method Prototype

N/A.

(Use the `DataMarkers.Name` property.)

C Prototype

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

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Index	A one-based index that defines which name to return	ViInt32
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the `Name` parameter	ViInt32

Outputs	Description	Base Type
Name	A user-allocated buffer into which the driver stores the channel name. The caller may pass `VI_NULL` for this parameter if the `NameBufferSize` parameter is 0.	ViChar[]

Return Values

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

25.3.3 Disable All Data Markers

Description

Disables all of the data markers by setting their Data Marker Destination attribute to None.

.NET Method Prototype

```
void DataMarkers.Clear();
```

COM Method Prototype

```
HRESULT DataMarkers.DisableAll();
```

C Prototype

```
ViStatus IviFgen_DisableAllDataMarkers (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.

25.4 IviFgenDataMarker Behavior Model

The IviFgenDataMarker Extension Group follows the behavior model of the
IviFgenArbWfmBinary capability group. The only modification to the behavior model from the
IviFgenArbWfmBinary capability group is the ability to configure IviFgenDataMarker settings.

25.5 IviFgenDataMarker Compliance Notes

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

26 IviFgenArbDataMask Extension Group

26.1 IviFgenArbDataMask Overview

The IviFgenArbDataMask extension group supports arbitrary waveform generators with the ability to mask out bits of the output data.

26.2 IviFgenArbDataMask Attributes

The IviFgenArbDataMask extension group defines the following attributes:

- Data Mask

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

26.2.1 Data Mask

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	R/W	N/A	None	N/A

.NET Property Name

> Arbitrary.DataMask

COM Property Name

> Arbitrary.DataMask

C Constant Name

> IVIFGEN_ATTR_OUTPUT_DATA_MASK

Description

> Determines which bits of the output data are masked out. This is especially useful when combined with Data Markers so that the bits embedded with the data to be used for markers are not actually output by the generator.
>
> A value of 1 for a particular bit indicates that the data bit should be output. A value of 0 indicates that the data bit should be masked out. For example, if the value of this property is 0xFFFFFFFF (all bits are 1), no masking is applied.

.NET 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.

26.3 IviFgenArbDataMask Functions

The IviFgenArbDataMask extension group defines no functions.

26.4 IviFgenArbDataMask Behavior Model

The IviFgenArbDataMask extension group follows the same behavior model as the IviFgenArbWfmBinary capability group. The only modification to the behavior model from the IviFgenArbWfmBinary capability group is the ability to configure IviFgenArbDataMask settings.

26.5 IviFgenArbDataMask Compliance Notes

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

27 IviFgenSparseMarker Extension Group

27.1 IviFgenSparseMarker Overview

The IviFgenSparseMarker Extension Group supports arbitrary waveform generators that can output signals, known as markers, associated with specified samples in the output data. Unlike data markers, sparse markers are not stored as part of the waveform data, but rather provided as a list of particular samples of the waveform on which the marker should be output. The user can choose which waveform and sample number the output is associated with, where the output goes, and various analog characteristics of the marker output. Sparse markers are repeated capabilities to allow the user to specify multiple markers to different outputs simultaneously.

Setting the Sparse Marker Destination attribute to a value other than None enables the sparse marker. To disable the sparse marker, set the Sparse Marker Destination to None.

27.2 IviFgenSparseMarker Attributes

The IviFgenSparseMarker capability group defines the following attributes:

- Sparse Marker Amplitude

- Sparse Marker Count

- Sparse Marker Delay

- Sparse Marker Destination

- Sparse Marker Item (IVI-COM only)

- Sparse Marker Name (IVI-COM only)

- Sparse Marker Polarity

- Sparse Marker Waveform Handle

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

27.2.1 Sparse Marker Amplitude

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViReal64	R/W	SparseMarker	None	None

.NET Property Name

    SparseMarkers[].Amplitude

COM Property Name

    SparseMarkers.Item().Amplitude

C Constant Name

    IVIFGEN_ATTR_SPARSEMARKER_AMPLITUDE

Description

Specifies the amplitude of the sparse marker output. The units are volts.

.NET Exceptions

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

27.2.2 Sparse Marker Count

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

.NET Property Name

SparseMarkers.Count

This property is inherited from IIviRepeatedCapabilityCollection.

COM Property Name

Sparse Markers. Count

C Constant Name

IVIFGEN_ATTR_SPARSEMARKER_COUNT

Description

Returns the number of sparse markers.

.NET Exceptions

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

27.2.3 Sparse Marker Delay

Data Type	Access	Applies To	Coercion	High Level Functions
ViReal64 (C/COM) PrecisionTimeSpan (.NET)	R/W	SparseMarker	Down	None

.NET Property Name

 SparseMarkers[].Delay

COM Property Name

 SparseMarkers.Item().Delay

C Constant Name

 IVIFGEN_ATTR_SPARSEMARKER_DELAY

Description

Specifies the amount of delay applied to the sparse marker output with respect to the analog data output. A value of zero indicates the marker is aligned with the analog data output. 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.

27.2.4 Sparse Marker Destination

Data Type	Access	Applies To	Coercion	High Level Functions
ViString	R/W	SparseMarker	None	Configure Sparse Marker

.NET Property Name

SparseMarkers[].Destination

COM Property Name

SparseMarkers.Item().Destination

C Constant Name

IVIFGEN_ATTR_SPARSEMARKER_DESTINATION

Description

Specifies the destination terminal for the sparse marker output.

Defined Values

The defined values for this attribute are given in Section 30, *IviFgen Attribute Value Definitions*.

.NET 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

The specific driver need not support all of the destination values defined in Section 30, *IviFgen Attribute Value Definitions*.

27.2.5 Sparse Marker Item (IVI-COM and IVI.NET Only)

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

.NET Property Name

> SparseMarkers[String name]

> This indexer is inherited from IIviRepeatedCapabilityCollection. The name parameter uniquely identifies a particular sparse marker in the sparse markers collection.

COM Property Name

> SparseMarkers.Item ([in] BSTR Name);

C Constant Name

> N A

Description

> Sparse Marker Item uniquely identifies a sparse marker in the sparse markers collection. It returns an interface pointer which can be used to control the attributes and other functionality of that sparse marker.

> The Item property takes a sparse marker name. If the user passes an invalid value for the sparse marker 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.

27.2.6 Sparse Marker Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

SparseMarkers[].Name

This property is inherited from IIviRepeatedCapabilityIdentification.

COM Property Name

SparseMarkers.Name ([in] LONG Index);

C Constant Name

N/A.

(Use the Get Sparse Marker Name function.)

Description

This attribute returns the repeated capability identifier defined by specific driver for the sparse marker that corresponds to the index that the user specifies. If the driver defines a qualified Sparse Marker name, this property returns the qualified name.

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

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

If the value that the user passes for the Index parameter is less than one or greater than the value of the Sparse Marker Count, the attribute returns an empty string for the value and returns an error.

.NET Exceptions

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

27.2.7 Sparse Marker Polarity

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

.NET Property Name

> SparseMarkers[].Polarity

.NET Enumeration Name

> MarkerPolarity

COM Property Name

> SparseMarkers.Item().Polarity

COM Enumeration Name

> IviFgenMarkerPolarityEnum

C Constant Name

> IVIFGEN_ATTR_SPARSEMARKER_POLARITY

Description

> Specifies the polarity of the sparse marker output.

Defined Values

Name	Description		
		Language	*Identifier*
Active High	When the data bit is 1 (high), the marker output level is high.		
		.NET	ActiveHigh
		C	IVIFGEN_VAL_MARKER_POLARITY_ACTIVE_HIGH
		COM	IviFgenMarkerPolarityActiveHigh
Active Low	When the data bit is 1 (high), the marker output level is low.		
		.NET	ActiveLow
		C	IVIFGEN_VAL MARKER_POLARITY_ACTIVE_LOW
		COM	IviFgenMarkerPolarityActiveLow

.NET Exceptions

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

27.2.8   Sparse Marker Waveform Handle

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	SparseMarker	None	Configure Sparse Marker

.NET Property Name

       SparseMarkers[].WaveformHandle

COM Property Name

       SparseMarkers.Item().WfmHandle

C Constant Name

       IVIFGEN_ATTR_SPARSEMARKER_WFMHANDLE

Description

       Specifies the waveform whose indexes the sparse marker refers to.

.NET Exceptions

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

27.3 IviFgenSparseMarker Functions

The IviFgenSparseMarker extension defines the following functions:

- Configure Sparse Marker

- Get Sparse Marker Name (IVI-C Only)

- Get Sparse Marker Indexes

- Set Sparse Marker Indexes

- Disable All Sparse Markers

This section describes the behavior and requirements of each function.

27.3.1 Configure Sparse Marker

Description

Configures some of the common sparse marker attributes.

.NET Method Prototype

```
void SparseMarkers[].Configure (Int32 waveformHandle,
                                Int64[] indexes
                                String destination);
```

COM Method Prototype

```
HRESULT SparseMarkers.Configure ([in] BSTR Name,
                            [in] LONG WfmHandle,
                            [in] SAFEARRAY(__int64) *Indexes
                            [in] BSTR Destination);
```

C Prototype

```
ViStatus IviFgen_ConfigureSparseMarker (ViSession Vi,
                                ViConstString Name,
                                ViInt32 WfmHandle,
                                ViInt64 NumIndexes,
                                ViInt64 Indexes[],
                                ViConstString Destination);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
Name	The name for the new sparse marker.	ViConstString
WfmHandle (C/COM) waveformHandle (.NET)	Specifies the waveform whose indexes the sparse marker refers to.	ViInt32
NumIndexes	Specifies the number of elements provided in the Indexes array.	ViInt64
Indexes	The sample numbers of the specified waveform on which markers will be output. These indexes may be coerced by the driver. Use the Get Sparse Marker Indexes function to find the coerced values.	ViInt64[]
Destination	Specifies the destination terminal for the sparse marker.	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.

27.3.2 Get Sparse Marker Name (IVI-C Only)

Description

This function returns the repeated capability identifier for the sparse marker that corresponds to the one-based index that the user specifies. If the value that the user passes for the `Index` parameter is less than one or greater than the value of the SparseMarker Count, the function returns an empty string in the `Name` parameter and returns an error.

.NET Method Prototype

N/A.

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

COM Method Prototype

N/A.

(Use the `SparseMarkers.Name` property.)

C Prototype

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

Parameters

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

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

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.

27.3.3 Get Sparse Marker Indexes

Description

Gets the coerced indexes associated with the sparse marker. These indexes are specified by either the Configure SparseMarker function or the Set Sparse Marker Indexes function.

.NET Method Prototype

```
Int64[] SparseMarkers[].GetIndexes ();
```

COM Method Prototype

```
HRESULT SparseMarkers.Item().GetIndexes (
                            [out,retval] SAFEARRAY(__int64)
*Indexes);
```

C Prototype

```
ViStatus IviFgen_GetSparseMarkerIndexes (ViSession Vi,
                                        ViConstString Name,
                                        ViInt64 IndexesArraySize,
                                        ViInt64 Indexes[],
                                        ViInt64 *IndexesActualSize);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
IndexesArraySize	Specifies the size of the Indexes array in elements	ViInt64

Outputs	Description	Base Type
Indexes (C, COM) Return Value (.NET)	Returns the sample numbers on which markers will be output.	ViInt64[]
IndexesActualSize	Returns the number of elements in Indexes array	ViInt64*

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.

27.3.4 Set Sparse Marker Indexes

Description

Sets the indexes associated with the sparse marker. These indexes may be coerced by the driver. Use the Get Sparse Marker Indexes function to find the coerced values.

.NET Method Prototype

```
void SparseMarkers[].SetIndexes (Int64[] indexes);
```

COM Method Prototype

```
HRESULT SparseMarkers.Item().SetIndexes (
                            [in] SAFEARRAY(__int64) *Indexes);
```

C Prototype

```
ViStatus IviFgen_SetSparseMarkerIndexes (ViSession Vi,
                                    ViConstString Name,
                                    ViInt64 IndexesArraySize,
                                    ViInt64 Indexes[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
IndexesArraySize	Specifies the size of the Indexes array in elements	ViInt64
Indexes	Specifies the sample numbers on which markers will be output.	ViInt64[]

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.

27.3.5 Disable All Sparse Markers

Description

Disables all of the sparse markers by setting their Sparse Marker Destination attribute to None.

.NET Method Prototype

```
void SparseMarkers.Clear();
```

COM Method Prototype

```
HRESULT SparseMarkers.DisableAll();
```

C Prototype

```
ViStatus IviFgen_DisableAllSparseMarkers (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.

27.4 IviFgenSparseMarker Behavior Model

The IviFgenSparseMarker Extension Group follows the behavior model of the IviFgenArbWfm capability group. The only modification to the behavior model from the IviFgenArbWfm capability group is the ability to configure IviFgenSparseMarker settings.

27.5 IviFgenSparseMarker Compliance Notes

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

28 IviFgenArbSeqDepth Extension Group

28.1 IviFgenArbSeqDepth Overview

The IviFgenArbSeqDepth extension group supports arbitrary waveform generators supporting IviFgenArbSeq and that are capable of producing sequences of sequences of arbitrary waveforms.

28.2 IviFgenArbSeqDepth Attributes

The IviFgenArbSeqDepth extension group defines the following attributes:

- Max Sequence Depth

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 29, *IviFgen Attribute ID Definitions*.

28.2.1 Max Sequence Depth

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	N/A

.NET Property Name

      Arbitrary.Sequence.DepthMax

COM Property Name

      Arbitrary.Sequence.DepthMax

COM Enumeration Name

      N/A

C Constant Name

      IVIFGEN_ATTR_SEQUENCE_DEPTH_MAX

Description

Returns the maximum sequence depth – that is, the number of times a sequence can include other sequences recursively. A depth of zero indicates the generator supports waveforms only. A depth of 1 indicates a generator supports sequences of waveforms, but not sequences of sequences. A depth of 2 or greater indicates that the generator supports sequences of sequences. Note that if the MaxSequenceDepth is 2 or greater, the driver must return unique handles for waveforms and sequences so that a sequence may contain both waveform and sequence handles.

.NET Exceptions

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

28.3 IviFgenArbSeqDepth Functions

The IviFgenArbSeqDepth extension group defines no functions.

28.4 IviFgenArbSeqDepth Behavior Model

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

28.5 IviFgenArbSeqDepth Compliance Notes

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

29 IviFgen Attribute ID Definitions

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

Table 29-1. IviFgen Attributes ID Values (IVI-C only)

Attribute Name	ID Definition
IVIFGEN_ATTR_CHANNEL_COUNT	IVI_INHERENT_ATTR_BASE + 203
IVIFGEN_ATTR_OUTPUT_MODE	IVI_CLASS_ATTR_BASE + 1
IVIFGEN_ATTR_REF_CLOCK_SOURCE	IVI_CLASS_ATTR_BASE + 2
IVIFGEN_ATTR_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 3
IVIFGEN_ATTR_OUTPUT_IMPEDANCE	IVI_CLASS_ATTR_BASE + 4
IVIFGEN_ATTR_OPERATION_MODE	IVI_CLASS_ATTR_BASE + 5
IVIFGEN_ATTR_SAMPLE_CLOCK_SOURCE	IVI_CLASS_ATTR_BASE + 21
IVIFGEN_ATTR_SAMPLE_CLOCK_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 22
IVIFGEN_ATTR_TERMINAL_CONFIGURATION	IVI_CLASS_ATTR_BASE + 31
IVIFGEN_ATTR_FUNC_WAVEFORM	IVI_CLASS_ATTR_BASE + 101
IVIFGEN_ATTR_FUNC_AMPLITUDE	IVI_CLASS_ATTR_BASE + 102
IVIFGEN_ATTR_FUNC_DC_OFFSET	IVI_CLASS_ATTR_BASE + 103
IVIFGEN_ATTR_FUNC_FREQUENCY	IVI_CLASS_ATTR_BASE + 104
IVIFGEN_ATTR_FUNC_START_PHASE	IVI_CLASS_ATTR_BASE + 105
IVIFGEN_ATTR_FUNC_DUTY_CYCLE_HIGH	IVI_CLASS_ATTR_BASE + 106
IVIFGEN_ATTR_ARB_WAVEFORM_HANDLE	IVI_CLASS_ATTR_BASE + 201
IVIFGEN_ATTR_ARB_GAIN	IVI_CLASS_ATTR_BASE + 202
IVIFGEN_ATTR_ARB_OFFSET	IVI_CLASS_ATTR_BASE + 203
IVIFGEN_ATTR_ARB_SAMPLE_RATE	IVI_CLASS_ATTR_BASE + 204
IVIFGEN_ATTR_MAX_NUM_WAVEFORMS	IVI_CLASS_ATTR_BASE + 205
IVIFGEN_ATTR_WAVEFORM_QUANTUM	IVI_CLASS_ATTR_BASE + 206
IVIFGEN_ATTR_MIN_WAVEFORM_SIZE	IVI_CLASS_ATTR_BASE + 207
IVIFGEN_ATTR_MAX_WAVEFORM_SIZE	IVI_CLASS_ATTR_BASE + 208
IVIFGEN_ATTR_ARB_FREQUENCY	IVI_CLASS_ATTR_BASE + 209
IVIFGEN_ATTR_ARB_SEQUENCE_HANDLE	IVI_CLASS_ATTR_BASE + 211
IVIFGEN_ATTR_MAX_NUM_SEQUENCES	IVI_CLASS_ATTR_BASE + 212
IVIFGEN_ATTR_MIN_SEQUENCE_LENGTH	IVI_CLASS_ATTR_BASE + 213
IVIFGEN_ATTR_MAX_SEQUENCE_LENGTH	IVI_CLASS_ATTR_BASE + 214
IVIFGEN_ATTR_MAX_LOOP_COUNT	IVI_CLASS_ATTR_BASE + 215
IVIFGEN_ATTR_MIN_WAVEFORM_SIZE64	IVI_CLASS_ATTR_BASE + 221
IVIFGEN_ATTR_MAX_WAVEFORM_SIZE64	IVI_CLASS_ATTR_BASE + 222
IVIFGEN_ATTR_BINARY_ALIGNMENT	IVI_CLASS_ATTR_BASE + 241

Table 29-1. IviFgen Attributes ID Values (IVI-C only)

Attribute Name	ID Definition
IVIFGEN_ATTR_SAMPLE_BIT_RESOLUTION	IVI_CLASS_ATTR_BASE + 242
IVIFGEN_ATTR_OUTPUT_DATA_MASK	IVI_CLASS_ATTR_BASE + 261
IVIFGEN_ATTR_SEQUENCE_DEPTH_MAX	IVI_CLASS_ATTR_BASE + 281
IVIFGEN_ATTR_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 302
IVIFGEN_ATTR_INTERNAL_TRIGGER_RATE	IVI_CLASS_ATTR_BASE + 310
IVIFGEN_ATTR_START_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 320
IVIFGEN_ATTR_START_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 321
IVIFGEN_ATTR_START_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 322
IVIFGEN_ATTR_START_TRIGGER_THRESHOLD	IVI_CLASS_ATTR_BASE + 323
IVIFGEN_ATTR_STOP_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 330
IVIFGEN_ATTR_STOP_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 331
IVIFGEN_ATTR_STOP_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 332
IVIFGEN_ATTR_STOP_TRIGGER_THRESHOLD	IVI_CLASS_ATTR_BASE + 333
IVIFGEN_ATTR_HOLD_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 340
IVIFGEN_ATTR_HOLD_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 341
IVIFGEN_ATTR_HOLD_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 342
IVIFGEN_ATTR_HOLD_TRIGGER_THRESHOLD	IVI_CLASS_ATTR_BASE + 343
IVIFGEN_ATTR_BURST_COUNT	IVI_CLASS_ATTR_BASE + 350
IVIFGEN_ATTR_RESUME_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 360
IVIFGEN_ATTR_RESUME_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 361
IVIFGEN_ATTR_RESUME_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 362
IVIFGEN_ATTR_RESUME_TRIGGER_THRESHOLD	IVI_CLASS_ATTR_BASE + 363
IVIFGEN_ATTR_ADVANCE_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 370
IVIFGEN_ATTR_ADVANCE_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 371
IVIFGEN_ATTR_ADVANCE_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 372
IVIFGEN_ATTR_ADVANCE_TRIGGER_THRESHOLD	IVI_CLASS_ATTR_BASE + 373
IVIFGEN_ATTR_AM_ENABLED	IVI_CLASS_ATTR_BASE + 401
IVIFGEN_ATTR_AM_SOURCE	IVI_CLASS_ATTR_BASE + 402
IVIFGEN_ATTR_AM_INTERNAL_DEPTH	IVI_CLASS_ATTR_BASE + 403
IVIFGEN_ATTR_AM_INTERNAL_WAVEFORM	IVI_CLASS_ATTR_BASE + 404
IVIFGEN_ATTR_AM_INTERNAL_FREQUENCY	IVI_CLASS_ATTR_BASE + 405
IVIFGEN_ATTR_FM_ENABLED	IVI_CLASS_ATTR_BASE + 501
IVIFGEN_ATTR_FM_SOURCE	IVI_CLASS_ATTR_BASE + 502
IVIFGEN_ATTR_FM_INTERNAL_DEVIATION	IVI_CLASS_ATTR_BASE + 503
IVIFGEN_ATTR_FM_INTERNAL_WAVEFORM	IVI_CLASS_ATTR_BASE + 504

Table 29-1. IviFgen Attributes ID Values (IVI-C only)

Attribute Name	ID Definition
IVIFGEN_ATTR_FM_INTERNAL_FREQUENCY	IVI_CLASS_ATTR_BASE + 505
IVIFGEN_ATTR_DATAMARKER_AMPLITUDE	IVI_CLASS_ATTR_BASE + 601
IVIFGEN_ATTR_DATAMARKER_BIT_POSITION	IVI_CLASS_ATTR_BASE + 602
IVIFGEN_ATTR_DATAMARKER_COUNT	IVI_CLASS_ATTR_BASE + 603
IVIFGEN_ATTR_DATAMARKER_DELAY	IVI_CLASS_ATTR_BASE + 604
IVIFGEN_ATTR_DATAMARKER_DESTINATION	IVI_CLASS_ATTR_BASE + 605
IVIFGEN_ATTR_DATAMARKER_POLARITY	IVI_CLASS_ATTR_BASE + 606
IVIFGEN_ATTR_DATAMARKER_SOURCE_CHANNEL	IVI_CLASS_ATTR_BASE + 607
IVIFGEN_ATTR_SPARSEMARKER_AMPLITUDE	IVI_CLASS_ATTR_BASE + 701
IVIFGEN_ATTR_SPARSEMARKER_COUNT	IVI_CLASS_ATTR_BASE + 702
IVIFGEN_ATTR_SPARSEMARKER_DELAY	IVI_CLASS_ATTR_BASE + 703
IVIFGEN_ATTR_SPARSEMARKER_DESTINATION	IVI_CLASS_ATTR_BASE + 704
IVIFGEN_ATTR_SPARSEMARKER_POLARITY	IVI_CLASS_ATTR_BASE + 705
IVIFGEN_ATTR_SPARSEMARKER_WFMHANDLE	IVI_CLASS_ATTR_BASE + 706

29.1 IviFgen Obsolete Attribute Names

The following attribute names are reserved by the IviFgen Specification 1.0. Future versions of this specification cannot use these names:

- IVIFGEN_ATTR_CYCLE_COUNT

29.2 IviFgen Obsolete Attribute ID Values

The following attribute ID values are reserved by the IviFgen Specification 1.0. Future versions of this specification cannot use these names:

- `IVI_CLASS_ATTR_BASE + 301`

30 IviFgen Attribute Value Definitions

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

Output Mode

Value Name	*Language*	*Identifier*	*Actual Value*
Output Function	.NET	Function	0
	C	IVIFGEN_VAL_OUTPUT_FUNC	0
	COM	IviFgenOutputModeFunction	0
Output Arbitrary	.NET	Arbitrary	1
	C	IVIFGEN_VAL_OUTPUT_ARB	1
	COM	IviFgenOutputModeArbitrary	1
Output Sequence	.NET	Sequence	2
	C	IVIFGEN_VAL_OUTPUT_SEQ	2
	COM	IviFgenOutputModeSequence	2
Output Mode Class Ext Base	C	IVIFGEN_VAL_OUT_MODE_CLASS_EXT_BASE	500
Output Mode Specific Ext Base	C	IVIFGEN_VAL_OUT_MODE_SPECIFIC_EXT_BASE	1000
	COM		1000

Reference Clock Source

Value Name	*Language*	*Identifier*	*Actual Value*
Reference Clock Internal	C	IVIFGEN_VAL_REF_CLOCK_INTERNAL	0
	COM	IviFgenReferenceClockSourceInternal	0
Reference Clock External	C	IVIFGEN_VAL_REF_CLOCK_EXTERNAL	1
	COM	IviFgenReferenceClockSourceExternal	1
Reference Clock RTSI Clock	C	IVIFGEN_VAL_REF_CLOCK_RTSI_CLOCK	101
	COM	IviFgenReferenceClockSourceRTSI	101
Reference Clock Class Ext Base	C	IVIFGEN_VAL_CLK_SRC_CLASS_EXT_BASE	500
Reference Clock Specific Ext Base	C	IVIFGEN_VAL_CLK_SRC_SPECIFIC_EXT_BASE	1000
	COM		1000

Operation Mode

Value Name	Language	Identifier	Actual Value
Operate Continuous	.NET	Continuous	0
	C	IVIFGEN_VAL_OPERATE_CONTINUOUS	0
	COM	IviFgenOperationModeContinuous	0
Operate Burst	.NET	Burst	1
	C	IVIFGEN_VAL_OPERATE_BURST	1
	COM	IviFgenOperationModeBurst	1
Operation Mode Class Ext Base	C	IVIFGEN_VAL_OP_MODE_CLASS_EXT_BASE	500
Operation Mode Specific Ext Base	C	IVIFGEN_VAL_OP_MODE_SPECIFIC_EXT_BASE	1000
	COM		1000

Waveform

(For .NET, the values used for Waveform are also used for AM Internal Waveform and FM Internal Waveform.)

Value Name	Language	Identifier	Actual Value
Waveform Sine	.NET	Sine	0
	C	IVIFGEN_VAL_WFM_SINE	1
	COM	IviFgenWaveformSine	1
Waveform Square	.NET	Square	1
	C	IVIFGEN_VAL_WFM_SQUARE	2
	COM	IviFgenWaveformSquare	2
Waveform Triangle	.NET	Triangle	2
	C	IVIFGEN_VAL_WFM_TRIANGLE	3
	COM	IviFgenWaveformTriangle	3
Waveform Ramp Up	.NET	RampUp	3
	C	IVIFGEN_VAL_WFM_RAMP_UP	4
	COM	IviFgenWaveformRampUp	4
Waveform Ramp Down	.NET	RampDown	4
	C	IVIFGEN_VAL_WFM_RAMP_DOWN	5
	COM	IviFgenWaveformRampDown	5
Waveform DC	.NET	DC	5

Value Name	Language	Identifier	Actual Value
	C	IVIFGEN_VAL_WFM_DC	6
	COM	IviFgenWaveformDC	6
Waveform Class Ext Base	C	IVIFGEN_VAL_WFM_CLASS_EXT_BASE	500
Waveform Specific Ext Base	C	IVIFGEN_VAL_WFM_SPECIFIC_EXT_BASE	1000
	COM		1000

Trigger Source

Value Name	Language	Identifier	Actual Value
External Trigger	C	IVIFGEN_VAL_EXTERNAL	1
	COM	IviFgenTriggerSourceExternal	1
Software Trigger	C	IVIFGEN_VAL_SOFTWARE_TRIG	2
	COM	IviFgenTriggerSourceSoftware	2
Internal Trigger	C	IVIFGEN_VAL_INTERNAL_TRIGGER	3
	COM	IviFgenTriggerSourceInternal	3
TTL0 Trigger	C	IVIFGEN_VAL_TTL0	111
	COM	IviFgenTriggerSourceTTL0	111
TTL1 Trigger	C	IVIFGEN_VAL_TTL1	112
	COM	IviFgenTriggerSourceTTL1	112
TTL2 Trigger	C	IVIFGEN_VAL_TTL2	113
	COM	IviFgenTriggerSourceTTL2	113
TTL3 Trigger	C	IVIFGEN_VAL_TTL3	114
	COM	IviFgenTriggerSourceTTL3	114
TTL4 Trigger	C	IVIFGEN_VAL_TTL4	115
	COM	IviFgenTriggerSourceTTL4	115
TTL5 Trigger	C	IVIFGEN_VAL_TTL5	116
	COM	IviFgenTriggerSourceTTL5	116
TTL6 Trigger	C	IVIFGEN_VAL_TTL6	117
	COM	IviFgenTriggerSourceTTL6	117
TTL7 Trigger	C	IVIFGEN_VAL_TTL7	118
	COM	IviFgenTriggerSourceTTL7	118
ECL0 Trigger	C	IVIFGEN_VAL_ECL0	119
	COM	IviFgenTriggerSourceECL0	119
ECL1 Trigger	C	IVIFGEN_VAL_ECL1	120
	COM	IviFgenTriggerSourceECL1	120
PXI Star Trigger	C	IVIFGEN_VAL_PXI_STAR	131
	COM	IviFgenTriggerSourcePXIStar	131
RTSI0 Trigger	C	IVIFGEN_VAL_RTSI_0	141
	COM	IviFgenTriggerSourceRTSI0	141
RTSI1 Trigger	C	IVIFGEN_VAL_RTSI_1	142
	COM	IviFgenTriggerSourceRTSI1	142
RTSI2 Trigger	C	IVIFGEN_VAL_RTSI_2	143
	COM	IviFgenTriggerSourceRTSI2	143

Value Name	Language	Identifier	Actual Value
RTSI3 Trigger	C	`IVIFGEN_VAL_RTSI_3`	144
	COM	`IviFgenTriggerSourceRTSI3`	144
RTSI4 Trigger	C	`IVIFGEN_VAL_RTSI_4`	145
	COM	`IviFgenTriggerSourceRTSI4`	145
RTSI5 Trigger	C	`IVIFGEN_VAL_RTSI_5`	146
	COM	`IviFgenTriggerSourceRTSI5`	146
RTSI6 Trigger	C	`IVIFGEN_VAL_RTSI_6`	147
	COM	`IviFgenTriggerSourceRTSI6`	147
Trigger Source Class Ext Base	C	`IVIFGEN_VAL_TRIG_SRC_CLASS_EXT_BASE`	500
Trigger Source Specific Ext Base	C	`IVIFGEN_VAL_TRIG_SRC_SPECIFIC_EXT_BASE`	1000
	COM		1000

The following values are reserved by the IviFgen Instrument Class Specification for the Trigger Source attribute. Future Versions of the specification cannot use these values for this attribute:

- 0
- 101

Start Trigger Source, Stop Trigger Source, Hold Trigger Source, Resume Trigger Source, Advance Trigger Source, Data Marker Destination, and Sparse Marker Destination

Name	Description		
	Language	Identifier	Actual Value
None	No trigger source		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_NONE`	""
	COM	""	""
Immediate	Trigger Immediately		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_IMMEDIATE`	"Immediate"
	COM	"Immediate"	"Immediate"
External	External trigger source		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_EXTERNAL`	"External"
	COM	"External"	"External"
Internal	Internal trigger source		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_INTERNAL`	"Internal"
	COM	"Internal"	"Internal"
Software	Software trigger		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_SOFTWARE`	"Software"
	COM	"Software"	"Software"

Name	Description			
	Language	Identifier		Actual Value
LAN0	LAN0 (LXI defined "LAN0" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN0		"LAN0"
	COM	"LAN0"		"LAN0"
LAN1	LAN1 (LXI defined "LAN1" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN1		"LAN1"
	COM	"LAN1"		"LAN1"
LAN2	LAN2(LXI defined "LAN2" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN2		"LAN2"
	COM	"LAN2"		"LAN2"
LAN3	LAN3 (LXI defined "LAN3" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN3		"LAN3"
	COM	"LAN3"		"LAN3"
LAN4	LAN4 (LXI defined "LAN4" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN4		"LAN4"
	COM	"LAN4"		"LAN4"
LAN5	LAN5 (LXI defined "LAN5" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN5		"LAN5"
	COM	"LAN5"		"LAN5"
LAN6	LAN6 (LXI defined "LAN6" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN6		"LAN6"
	COM	"LAN6"		"LAN6"
LAN7	LAN7 (LXI defined "LAN6" LAN message)			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LAN7		"LAN7"
	COM	"LAN7"		"LAN7"
LXI0	LXI Trigger Bus Line 0			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI0		"LXI0"
	COM	"LXI0"		"LXI0"
LXI1	LXI Trigger Bus Line 1			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI1		"LXI1"
	COM	"LXI1"		"LXI1"
LXI2	LXI Trigger Bus Line 2			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI2		"LXI2"
	COM	"LXI2"		"LXI2"
LXI3	LXI Trigger Bus Line 3			
	C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI3		"LXI3"
	COM	"LXI3"		"LXI3"
LXI4	LXI Trigger Bus Line 4			

Name	Description			
		Language	Identifier	Actual Value
		C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI4	"LXI4″
		COM	"LXI4"	"LXI4"
LXI5	LXI Trigger Bus Line 5			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI5	"LXI5"
		COM	"LXI5"	"LXI5"
LXI6	LXI Trigger Bus Line 6			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI6	"LXI6"
		COM	"LXI6"	"LXI6"
LXI7	LXI Trigger Bus Line 7			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_LXI7	"LXI7"
		COM	"LXI7"	"LXI7"
TTL0	TTL Interface 0			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL0	"TTL0"
		COM	"TTL0"	"TTL0"
TTL1	TTL Interface 1			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL1	"TTL1"
		COM	"TTL1"	"TTL1"
TTL2	TTL Interface 2			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL2	"TTL2"
		COM	"TTL2"	"TTL2"
TTL3	TTL Interface 3			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL3	"TTL3"
		COM	"TTL3"	"TTL3"
TTL4	TTL Interface 4			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL4	"TTL4"
		COM	"TTL4"	"TTL4"
TTL5	TTL Interface 5			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL5	"TTL5"
		COM	"TTL5"	"TTL5"
TTL6	TTL Interface 6			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL6	"TTL6"
		COM	"TTL6"	"TTL6"
TTL7	TTL Interface 7			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_TTL7	"TTL7"
		COM	"TTL7"	"TTL7"
PXI_STAR	PXI Star Interface			
		C	IVIFGEN_VAL_TRIGGER_SOURCE_PXI_STAR	"PXI_STAR"

Name	Description			
		Language	*Identifier*	*Actual Value*
		COM	`"PXI_STAR"`	`"PXI_STAR"`
PXI_TRIG0	PXI Trigger Bus Line 0			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG0`	`"PXI_TRIG0"`
		COM	`"PXI_TRIG0"`	`"PXI_TRIG0"`
PXI_TRIG1	PXI Trigger Bus Line 1			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG1`	`"PXI_TRIG1"`
		COM	`"PXI_TRIG1"`	`"PXI_TRIG1"`
PXI_TRIG2	PXI Trigger Bus Line 2			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG2`	`"PXI_TRIG2"`
		COM	`"PXI_TRIG2"`	`"PXI_TRIG2"`
PXI_TRIG3	PXI Trigger Bus Line 3			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG3`	`"PXI_TRIG3"`
		COM	`"PXI_TRIG3"`	`"PXI_TRIG3"`
PXI_TRIG4	PXI Trigger Bus Line 4			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG4`	`"PXI_TRIG4"`
		COM	`"PXI_TRIG4"`	`"PXI_TRIG4"`
PXI_TRIG5	PXI Trigger Bus Line 5			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG5`	`"PXI_TRIG5"`
		COM	`"PXI_TRIG5"`	`"PXI_TRIG5"`
PXI_TRIG6	PXI Trigger Bus Line 6			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG6`	`"PXI_TRIG6"`
		COM	`"PXI_TRIG6"`	`"PXI_TRIG6"`
PXI_TRIG7	PXI Trigger Bus Line 7			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXI_TRIG7`	`"PXI_TRIG7"`
		COM	`"PXI_TRIG7"`	`"PXI_TRIG7"`
PXIe_DSTARA	PXI Express DStar Line A			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXIE_DSTAR A`	`"PXIe_DSTARA"`
		COM	`"PXIe_DSTARA"`	`"PXIe_DSTARA"`
PXIe_DSTARB	PXI Express DStar Line B			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXIE_DSTAR B`	`"PXIe_DSTARB"`
		COM	`"PXIe_DSTARB"`	`"PXIe_DSTARB"`
PXIe_DSTARC	PXI Express DStar Line C			
		C	`IVIFGEN_VAL_TRIGGER_SOURCE_PXIE_DSTAR C`	`"PXIe_DSTARC"`
		COM	`"PXIe_DSTARC"`	`"PXIe_DSTARC"`
RTSI0	RTSI Bus Line 0			

Name	Description		
	Language	*Identifier*	*Actual Value*
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI0`	`"RTSI0"`
	COM	`"RTSI0"`	`"RTSI0"`
RTSI1	RTSI Bus Line 1		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI1`	`"RTSI1"`
	COM	`"RTSI1"`	`"RTSI1"`
RTSI2	RTSI Bus Line 3		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI2`	`"RTSI2"`
	COM	`"RTSI2"`	`"RTSI2"`
RTSI3	RTSI Bus Line 3		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI3`	`"RTSI3"`
	COM	`"RTSI3"`	`"RTSI3"`
RTSI4	RTSI Bus Line 4		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI4`	`"RTSI4"`
	COM	`"RTSI4"`	`"RTSI4"`
RTSI5	RTSI Bus Line 5		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI5`	`"RTSI5"`
	COM	`"RTSI5"`	`"RTSI5"`
RTSI6	RTSI Bus Line 6		
	C	`IVIFGEN_VAL_TRIGGER_SOURCE_RTSI6`	`"RTSI6"`
	COM	`"RTSI6"`	`"RTSI6"`

Sample Clock Source

Value Name	*Language*	*Identifier*	*Actual Value*
Internal	.NET	`Internal`	`0`
	C	`IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_INTERNAL`	`0`
	COM	`IviFgenSampleClockSourceInternal`	`0`
External	.NET	`External`	`1`
	C	`IVIFGEN_VAL_SAMPLE_CLOCK_SOURCE_EXTERNAL`	`1`
	COM	`IviFgenSampleClockSourceExternal`	`1`

Marker Polarity

Value Name	*Language*	*Identifier*	*Actual Value*
Active High	.NET	`ActiveHigh`	0

Value Name	Language	Identifier	Actual Value
	C	IVIFGEN_VAL_MARKER_POLARITY_ACTIVE_HIGH	0
	COM	IviFgenMarkerPolarityActiveHigh	0
Active Low	.NET	ActiveLow	1
	C	IVIFGEN_VAL_MARKER_POLARITY_ACTIVE_LOW	1
	COM	IviFgenMarkerPolarityActiveLow	1

AM Source

Value Name	Language	Identifier	Actual Value
AM Internal	.NET	Internal	0
	C	IVIFGEN_VAL_AM_INTERNAL	0
	COM	IviFgenAMSourceInternal	0
AM External	.NET	External	1
	C	IVIFGEN_VAL_AM_EXTERNAL	1
	COM	IviFgenAMSourceExternal	1
AM Source Class Ext Base	C	IVIFGEN_VAL_AM_SOURCE_CLASS_EXT_BASE	500
AM Source Specific Ext Base	C	IVIFGEN_VAL_AM_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

AM Internal Waveform

(For .NET, see values for Waveform.)

Value Name	Language	Identifier	Actual Value
AM Internal Sine	C	IVIFGEN_VAL_AM_INTERNAL_SINE	1
	COM	IviFgenAMInternalWaveformSine	1
AM Internal Square	C	IVIFGEN_VAL_AM_INTERNAL_SQUARE	2
	COM	IviFgenAMInternalWaveformSquare	2
AM Internal Triangle	C	IVIFGEN_VAL_AM_INTERNAL_TRIANGLE	3
	COM	IviFgenAMInternalWaveformTriangle	3
AM Internal Ramp Up	C	IVIFGEN_VAL_AM_INTERNAL_RAMP_UP	4
	COM	IviFgenAMInternalWaveformRampUp	4
AM Internal Ramp Down	C	IVIFGEN_VAL_AM_INTERNAL_RAMP_DOWN	5
	COM	IviFgenAMInternalWaveformRampDown	5

Value Name	Language	Identifier	Actual Value
AM Internal Waveform Class Ext Base	C	IVIFGEN_VAL_AM_INTERNAL_WFM_CLASS_EXT_BASE	500
AM Internal Waveform Specific Ext Base	C	IVIFGEN_VAL_AM_INTERNAL_WFM_SPECIFIC_EXT_BASE	1000
	COM		1000

The following values are reserved by the IviFgen Instrument Class Specification for the AM Internal Waveform attribute. Future versions of the specification cannot use these values for this attribute:

- 6

FM Source

Value Name	Language	Identifier	Actual Value
FM Internal	.NET	Internal	0
	C	IVIFGEN_VAL_FM_INTERNAL	0
	COM	IviFgenFMSourceInternal	0
FM External	.NET	External	1
	C	IVIFGEN_VAL_FM_EXTERNAL	1
	COM	IviFgenFMSourceExternal	1
FM Source Class Ext Base	C	IVIFGEN_VAL_FM_SOURCE_CLASS_EXT_BASE	500
FM Source Specific Ext Base	C	IVIFGEN_VAL_FM_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

FM Internal Waveform

(For .NET, see values for Waveform.)

Value Name	Language	Identifier	Actual Value
FM Internal Sine	C	`IVIFGEN_VAL_FM_INTERNAL_SINE`	1
	COM	`IviFgenFMInternalWaveformSine`	1
FM Internal Square	C	`IVIFGEN_VAL_FM_INTERNAL_SQUARE`	2
	COM	`IviFgenFMInternalWaveformSquare`	2
FM Internal Triangle	C	`IVIFGEN_VAL_FM_INTERNAL_TRIANGLE`	3
	COM	`IviFgenFMInternalWaveformTriangle`	3
FM Internal Ramp Up	C	`IVIFGEN_VAL_FM_INTERNAL_RAMP_UP`	4
	COM	`IviFgenFMInternalWaveformRampUp`	4
FM Internal Ramp Down	C	`IVIFGEN_VAL_FM_INTERNAL_RAMP_DOWN`	5
	COM	`IviFgenFMInternalWaveformRampDown`	5
FM Internal Waveform Class Ext Base	C	`IVIFGEN_VAL_FM_INTERNAL_WFM_CLASS_EXT_BASE`	500
FM Internal Waveform Specific Ext Base	C	`IVIFGEN_VAL_FM_INTERNAL_WFM_SPECIFIC_EXT_BASE`	1000
	COM		1000

The following values are reserved by the IviFgen Instrument Class Specification for the FM Internal Waveform attribute. Future versions of the specification cannot use these values for this attribute:

- 6

Binary Alignment

Value Name	Language	Identifier	Actual Value
Binary Alignment Left	.NET	Left	0
	C	IVIFGEN_VAL_BINARY_ALIGNMENT_LEFT	0
	COM	IviFgenBinaryAlignmentLeft	0
Binary Alignment Right	.NET	Right	1
	C	IVIFGEN_VAL_BINARY_ALIGNMENT_RIGHT	1
	COM	IviFgenBinaryAlignmentRight	1
Binary Alignment Class Ext Base	C	IVIFGEN_VAL_BINARY_ALIGNMENT_CLASS_EXT_BASE	500
Binary Alignment Specific Ext Base	C	IVIFGEN_VAL_BINARY_ALIGNMENT_SPECIFIC_EXT_BASE	1000
	COM		1000

Terminal Configuration

Value Name	Language	Identifier	Actual Value
Terminal Configuration Single Ended	.NET	`SingleEnded`	0
	C	`IVIFGEN_VAL_TERMINAL_CONFIGURATION_SINGLE_ENDED`	0
	COM	`IviFgenTerminalConfigurationSingleEnded`	0
Terminal Configuration Differential	.NET	`Differential`	1
	C	`IVIFGEN_VAL_TERMINAL_CONFIGURATION_DIFFERENTIAL`	1
	COM	`IviFgenTerminalConfigurationDifferential`	1
Terminal Configuration Class Ext Base	C	`IVIFGEN_VAL_TERMINAL_CONFIGURATION_CLASS_EXT_BASE`	500
Terminal Configuration Specific Ext Base	C	`IVIFGEN_VAL_TERMINAL_CONFIGURATION_SPECIFIC_EXT_BASE`	1000
	COM		1000

TriggerSlope

Value Name	Language	Identifier	Actual Value
Positive	.NET	Positive	0
	C	IVIFGEN_VAL_TRIGGER_POSITIVE	0
	COM	IviFgenTriggerSlopePositive	0
Negative	.NET	Negative	1
	C	IVIFGEN_VAL_TRIGGER_NEGATIVE	1
	COM	IviFgenTriggerSlopNegative	1
Either	.NET	Either	2
	C	IVIFGEN_VAL_TRIGGER_EITHER	2
	COM	IviFgenTriggerSlopeEither	2
Trigger Slope Class Ext Base	C	IVIFGEN_VAL_TRIGGER_CLASS_EXT_BASE	500
Trigger Slope Specific Ext Base	C	IVIFGEN_VAL_TRIGGER_SPECIFIC_EXT_BASE	1000
	COM		1000

30.1 IviFgen Obsolete Attribute Value Names

The following attribute value names are reserved by the IviFgen specification 1.0. Future versions of this specification cannot use these names:

- IVIFGEN_VAL_50_OHMS

- IVIFGEN_VAL_75_OHMS

- IVIFGEN_VAL_1_MEGA_OHM

- IVIFGEN_VAL_WFM_BASE

- IVIFGEN_VAL_FIRST_WAVEFORM_HANDLE

- IVIFGEN_VAL_LAST_WAVEFORM_HANDLE

- IVIFGEN_VAL_FIRST_SEQUENCE_HANDLE

- IVIFGEN_VAL_LAST_SEQUENCE_HANDLE

- IVIFGEN_VAL_GENERATE_CONTINUOUS

- IVIFGEN_VAL_IMMEDIATE

- IVIFGEN_VAL_SW_TRIG_FUNC

- IVIFGEN_VAL_GPIB_GET

- IVIFGEN_VAL_AM_INTERNAL_DC

- IVIFGEN_VAL_FM_INTERNAL_DC

31 IviFgen Function Parameter Value Definitions

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

Clear Arbitrary Waveform

Parameter: `WaveformHandle`

Value Name	*Language*	*Identifier*	*Actual Value*
All Waveforms	C	`IVIFGEN_VAL_ALL_WAVEFORMS`	-1
	COM		-1
	.NET		-1

Clear Arbitrary Sequence

Parameter: `SequenceHandle`

Value Name	*Language*	*Identifier*	*Actual Value*
All Sequences	C	`IVIFGEN_VAL_ALL_SEQUENCES`	-1
	COM		-1
	.NET		-1

32 Error, Completion Code, and Exception Class Definitions

Table **32-1** specifies the actual value for each status code that the IviFgen class specification defines.

<div align="center">Table 32-1. IviFgen Completion Code</div>

Error Name	Description		
	Language	*Identifier*	*Value(hex)*
Trigger Not Software	The trigger source is not set to software trigger.		
	.NET	`Ivi.Driver.` `TriggerNotSoftwareException`	IVI Defined Exception (See IVI-3.2)
	C	`IVIFGEN_ERROR_TRIGGER_NOT_SOFTWARE`	0xBFFA1001
	COM	`E_IVIFGEN_TRIGGER_NOT_SOFTWARE`	0x80041001
No Waveforms Available	No Waveforms Available		
	.NET	`NoWaveformsAvailableException`	N/A
	C	`IVIFGEN_ERROR_NO_WFMS_AVAILABLE`	0xBFFA2004
	COM	`E_IVIFGEN_NO_WFMS_AVAILABLE`	0x80042004
Waveform In Use	Waveform In Use		
	.NET	`WaveformInUseException`	N/A
	C	`IVIFGEN_ERROR_WFM_IN_USE`	0xBFFA2008
	COM	`E_IVIFGEN_WFM_IN_USE`	0x80042008
No Sequences Available	No Sequences Available		
	.NET	`NoSequencesAvailableException`	N/A
	C	`IVIFGEN_ERROR_NO_SEQS_AVAILABLE`	0xBFFA2009
	COM	`E_IVIFGEN_NO_SEQS_AVAILABLE`	0x80042009
Sequence In Use	Sequence In Use		
	.NET	`SequenceInUseException`	N/A
	C	`IVIFGEN_ERROR_SEQ_IN_USE`	0xBFFA200D
	COM	`E_IVIFGEN_SEQ_IN_USE`	0x8004200D
Invalid Waveform Channel	Invalid Waveform Channel		
	.NET	`InvalidWaveformChannelException`	N/A
	C	`IVIFGEN_ERROR_INVALID_WFM_CHANNEL`	0xBFFA200E
	COM	`E_IVIFGEN_INVALID_WFM_CHANNEL`	0x8004200E

Table **32-2** defines the recommended format of the message string associated with the errors. In C, these strings are returned by the Get Error function. In COM, these strings are the description contained in the ErrorInfo object.   For .NET, exception default message strings are defined with the exception.

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

<div align="center">

Table 32-2. Error Message Strings

</div>

Name	Message String
Trigger Not Software	"%s: Trigger source is not set to software trigger"
No Waveforms Available	"%s: No waveforms available"
Waveform In Use	"%s: Waveform in use"
No Sequences Available	"%s: No sequences available"
Sequence In Use	"%s: Sequence in use"
Invalid Waveform Channel	"%s: Invalid Waveform Channel. Waveform was created on channel %s1 and is being configured on channel %s2." %s1 = Channel Name waveform was created on %s2 = Channel Name waveform is being configured on

32.1 IVI.NET IviFgen Exceptions and Warnings

This section defines the list of IVI.NET exceptions and warnings that are specific to the IviFgen class. For general information on IVI.NET exceptions and warnings, refer to *IVI-3.1: Driver Architecture Specification* and section 12, *Common IVI.NET Exceptions and Warnings*, of *IVI-3.2: Inherent Capabilities Specification*.

The IVI.NET exceptions defined in this specification are declared in the Ivi.Fgen namespace.
- InvalidWaveformChannelException
- NoSequencesAvailableException
- NoWaveformsAvailableException
- SequenceInUseException
- WaveformInUseException

32.1.1 InvalidWaveformChannelException

Description

This exception is used when the waveform was created on a different channel than the one for which it is being configured.

Constructors

```
Ivi.Fgen.InvalidWaveformChannelException();

Ivi.Fgen.InvalidWaveformChannelException(String message);

Ivi.Fgen.InvalidWaveformChannelException(String message,
                                         System.Exception innerException);
```

Message String

```
The waveform was created on one channel and is being configured on
another.
Created on Channel: <creationChannelName>
Configured on Channel: <configureChannelName>
```

Usage

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

32.1.2 NoSequencesAvailableException

Description

This exception is used when the driver finds that no sequences are defined.

Constructors

```
Ivi.Fgen.NoSequencesAvailableException();

Ivi.Fgen.NoSequencesAvailableException(String message);

Ivi.Fgen.NoSequencesAvailableException(String message,
                                       System.Exception innerException);
```

Message String

```
No sequences are available.
```

Usage

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

32.1.3 NoWaveformsAvailableException

Description

This exception is used when the driver finds that no waveforms are defined.

Constructors

```
Ivi.Fgen.NoWaveformsAvailableException();

Ivi.Fgen.NoWaveformsAvailableException(String message);

Ivi.Fgen.NoWaveformsAvailableException(String message,
                                       System.Exception innerException);
```

Message String

```
No waveforms are available.
```

Usage

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

32.1.4 SequenceInUseException

Description

This exception is used when the driver finds that a specified sequence is in use.

Constructors

```
Ivi.Fgen.SequenceInUseException(String message,
                                String sequenceName);

Ivi.Fgen.SequenceInUseException();

Ivi.Fgen.SequenceInUseException(String message);

Ivi.Fgen.SequenceInUseException(String message,
                                System.Exception innerException);
```

Message String

```
The specified sequence is in use.
Sequence name: <sequenceName>
```

Parameters

Inputs	Description	Base Type
sequenceName	The name of the sequence in use.	String

Usage

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

32.1.5 WaveformInUseException

Description

This exception is used when the driver finds that a specified waveform is in use.

Constructors

```
Ivi.Fgen.WaveformInUseException(String message,
                                String waveformName);

Ivi.Fgen.WaveformInUseException();

Ivi.Fgen.WaveformInUseException(String message);

Ivi.Fgen.WaveformInUseException(String message,
                                System.Exception innerException);
```

Message String

```
The specified waveform is in use.
Waveform name: <waveformName>
```

Parameters

Inputs	Description	Base Type
waveformName	The name of the waveform in use.	String

Usage

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

32.2 Obsolete Error and Completion Code Names

The following error and completion code names are reserved by the IviFgen specification 1.0. Future versions of this specification cannot use these names:

- IVIFGEN_ERROR_NOT_CONFIGURABLE

- IVIFGEN_ERROR_NOT_GENERATING

- IVIFGEN_ERROR_INVALID_MODE

- IVIFGEN_ERROR_INVALID_WFM_LENGTH

- IVIFGEN_ERROR_INVALID_WFM_ELEMENT

- IVIFGEN_ERROR_INVALID_WAVEFORM

- IVIFGEN_ERROR_INVALID_SEQ_LENGTH

- IVIFGEN_ERROR_INVALID_LOOP_COUNT

- IVIFGEN_ERROR_INVALID_SEQUENCE

32.3 Obsolete Error and Completion Code Values

The following error and completion code names are reserved by the IviFgen specification 1.0.
Future versions of this specification cannot use these names:

- `IVI_CLASS_ERROR_BASE + 1`

- `IVI_CLASS_ERROR_BASE + 2`

- `IVI_CLASS_ERROR_BASE + 3`

- `IVI_CLASS_ERROR_BASE + 5`

- `IVI_CLASS_ERROR_BASE + 6`

- `IVI_CLASS_ERROR_BASE + 7`

- `IVI_CLASS_ERROR_BASE + 10`

- `IVI_CLASS_ERROR_BASE + 11`

- `IVI_CLASS_ERROR_BASE + 12`

33 IviFgen Hierarchies

33.1 IviFgen .NET Hierarchy

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

Note that some Fgen properties are channel dependent. The properties are implemented as a pair of methods named Set<property name> and Get<property name>, where <property name> is the name of the corresponding COM property. This is due to the fact that properties in CLR-compliant .NET may not take parameters.

Table 33-1. IviFgen .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
InitiateGeneration	Initiate Generation	M
AbortGeneration	Abort Generation	M
OutputMode	Output Mode	P
ReferenceClockSource	Reference Clock Source	P
Output		
SetEnabled & GetEnabled	Output Enabled	P
SetImpedance & GetImpedance	Output Impedance	P
SetOperationMode & GetOperationMode	Operation Mode	P
Count	Count	P
Name	Name	P
TerminalConfiguration	Terminal Configuration	P
Trigger		
InternalRate	Internal Trigger Rate	P
Start		
Configure	Configure Start Trigger	M
SendSoftwareTrigger	Send Software Trigger	M
SetBurstCount & GetBurstCount	Burst Count	P
SetDelay & GetDelay	Start Trigger Delay	P
SetSlope & GetSlope	Start Trigger Slope	P
SetSource & GetSource	Trigger Source	P
SetThreshold & GetThreshold	Start Trigger Threshold	P
Stop		
Configure	Configure Stop Trigger	M
SendSoftwareTrigger	Send Software Stop Trigger	M
SetDelay & GetDelay	Stop Trigger Delay	P
SetSlope & GetSlope	Stop Trigger Slope	P

Table 33-1. IviFgen .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
`SetSource & GetSource`	Stop Trigger Source	P
`SetThreshold & GetThreshold`	Stop Trigger Threshold	P
`Hold`		
`Configure`	Configure Hold Trigger	M
`SendSoftwareTrigger`	Send Software Hold Trigger	M
`SetDelay & GetDelay`	Hold Trigger Delay	P
`SetSlope & GetSlope`	Hold Trigger Slope	P
`SetSource & GetSource`	Hold Trigger Source	P
`SetThreshold & GetThreshold`	Hold Trigger Threshold	P
`Resume`		
`Configure`	Configure Resume Trigger	M
`SendSoftwareTrigger`	Send Software Resume Trigger	M
`SetDelay & GetDelay`	Resume Trigger Delay	P
`SetSlope & GetSlope`	Resume Trigger Slope	P
`SetSource & GetSource`	Resume Trigger Source	P
`SetThreshold & GetThreshold`	Resume Trigger Threshold	P
`Advance`		
`Configure`	Configure Advance Trigger	M
`SendSoftwareTrigger`	Send Software Advance Trigger	M
`SetDelay & GetDelay`	Advance Trigger Delay	P
`SetSlope & GetSlope`	Advance Trigger Slope	P
`SetSource & GetSource`	Advance Trigger Source	P
`SetThreshold & GetThreshold`	Advance Trigger Threshold	P
`DataMarkers`		
`Clear`	Disable All Data Markers	M
`Count`	Data Marker Count	P
`Item`		P
`Configure`	Configure Data Marker	M
`Amplitude`	Data Marker Amplitude	P
`BitPosition`	Data Marker Bit Position	P
`Delay`	Data Marker Delay	P
`Destination`	Data Marker Destination	P
`Name`	Data Marker Name	P
`Polarity`	Data Marker Polarity	P
`SourceChannel`	Data Marker Source Channel	P
`SparseMarkers`		
`Clear`	Disable All Sparse Markers	M

Table 33-1. IviFgen .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Count	Sparse Marker Count	P
Item		P
Configure	Configure Sparse Marker	M
Get Indexes	Get Sparse Marker Indexes	M
Set Indexes	Set Sparse Marker Indexes	M
Amplitude	Sparse Marker Amplitude	P
Delay	Sparse Marker Delay	P
Destination	Sparse Marker Destination	P
Name	Sparse Marker Name	P
Polarity	Sparse Marker Polarity	P
WaveformHandle	Sparse Marker Waveform Handle	P
StandardWaveform		
Configure	Configure Standard Waveform	M
SetDutyCycleHigh & GetDutyCycleHigh	Duty Cycle High	P
SetAmplitude & GetAmplitude	Amplitude	P
SetDCOffset & GetDCOffset	DC Offset	P
SetFrequency & GetFrequency	Frequency	P
SetStartPhase & GetStartPhase	Start Phase	P
SetWaveformFunction & GetWaveformFunction	Waveform	P
Arbitrary		
ClearMemory	Clear Arbitrary Memory	M
Binary Alignment	Binary Alignment	P
Data Mask	Data Mask	P
SampleBitResolution	Sample Bit Resolution	P
SetGain & GetGain	Arbitrary Gain	P
SetOffset & GetOffset	Arbitrary Offset	P
SampleRate	Arbitrary Sample Rate	P
Waveform		
Configure	Configure Arbitrary Waveform	M
Clear	Clear Arbitrary Waveform	M
CreateChannelWaveform	Create Channel Arbitrary Waveform	M
SetFrequency & GetFrequency	Arbitrary Frequency	P
SetHandle & GetHandle	Waveform Handle	P
NumberWaveformsMax	Number Waveforms Max	P
Quantum	Waveform Quantum	P
SizeMax	Waveform Size Max	P

Table 33-1. IviFgen .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
SizeMin	Waveform Size Min	P
Sequence		
Configure	Configure Arbitrary Sequence	M
Clear	Clear Arbitrary Sequence	M
Create	Create Arbitrary Sequence	M
DepthMax	Sequence Depth Max	P
SetHandle & GetHandle	Sequence Handle	P
LengthMax	Sequence Length Max	P
LengthMin	Sequence Length Min	P
LoopCountMax	Loop Count Max	P
NumberSequencesMax	Number Sequences Max	P
AM		
ConfigureInternal	Configure AM Internal	M
SetEnabled & GetEnabled	AM Enabled	P
SetSource & GetSource	AM Source	P
InternalDepth	AM Internal Depth	P
InternalFrequency	AM Internal Frequency	P
InternalWaveformFunction	AM Internal Waveform	P
FM		
ConfigureInternal	Configure FM Internal	M
SetEnabled & GetEnabled	FM Enabled	P
SetSource & GetSource	FM Source	P
InternalDeviation	FM Internal Deviation	P
InternalFrequency	FM Internal Frequency	P
InternalWaveformFunction	FM Internal Waveform	P
SampleClock		
Source	Sample Clock Source	P
OutputEnabled	Sample Clock Output Enabled	P

33.1.1 IviFgen .NET Interfaces

Interface reference properties are used to navigate the IviFgen .NET hierarchy. This section describes the interface reference properties that the IIviFgen and IIviFgenArbitrary interfaces define.

Table 33-2. IviFgen .NET Interface Reference Properties

Data Type	.NET Property Name
IIviFgen	
IIviFgenOutput	Output
IIviFgenStandardWaveform	StandardWaveform
IIviFgenArbitrary	Arbitrary
IIviFgenArbitraryWaveform	Arbitrary.Waveform
IIviFgenArbitrarySequence	Arbitrary.Sequence
IIviFgenTrigger	Trigger
IIviFgenTriggerStart	Trigger.Start
IIviFgenTriggerStop	Trigger.Stop
IIviFgenTriggerHold	Trigger.Hold
IIviFgenTriggerResume	Trigger.Resume
IIviFgenTriggerAdvance	Trigger.Advance
IIviFgenAM	AM
IIviFgenFM	FM
IIviFgenDataMarkerCollection	DataMarkers
IIviFgenDataMarker	DataMarkers[]
IIviFgenSparseMarkerCollection	SparseMarkers
IIviFgenSparseMarker	SparseMarkers[]
IIviFgenSampleClock	SampleClock

33.2 IviFgen COM Hierarchy

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

Table 33-3. IviFgen COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
InitiateGeneration	Initiate Generation	M
AbortGeneration	Abort Generation	M
Output		
Enabled	Output Enabled	P
Impedance	Output Impedance	P
OutputMode	Output Mode	P
OperationMode	Operation Mode	P
ReferenceClockSource	Reference Clock Source	P
Count	Count	P
Name	Name	P
TerminalConfiguration	Terminal Configuration	P
Trigger		
SendSoftwareTrigger	Send Software Trigger	M
Source	Trigger Source	P
InternalRate	Internal Trigger Rate	P
BurstCount	Burst Count	P
Start		
Configure	Configure Start Trigger	M
Delay	Start Trigger Delay	P
Slope	Start Trigger Slope	P
Source	Trigger Source	P
Threshold	Start Trigger Threshold	P
Stop		
Configure	Configure Stop Trigger	M
SendSoftwareTrigger	Send Software Stop Trigger	M
Delay	Stop Trigger Delay	P
Slope	Stop Trigger Slope	P
Source	Stop Trigger Source	P
Threshold	Stop Trigger Threshold	P
Hold		
Configure	Configure Hold Trigger	M
SendSoftwareTrigger	Send Software Hold Trigger	M

Table 33-3. IviFgen COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Delay	Hold Trigger Delay	P
Slope	Hold Trigger Slope	P
Source	Hold Trigger Source	P
Threshold	Hold Trigger Threshold	P
Resume		
Configure	Configure Resume Trigger	M
SendSoftwareTrigger	Send Software Resume Trigger	M
Delay	Resume Trigger Delay	P
Slope	Resume Trigger Slope	P
Source	Resume Trigger Source	P
Threshold	Resume Trigger Threshold	P
Advance		
Configure	Configure Advance Trigger	M
SendSoftwareTrigger	Send Software Advance Trigger	M
Delay	Advance Trigger Delay	P
Slope	Advance Trigger Slope	P
Source	Advance Trigger Source	P
Threshold	Advance Trigger Threshold	P
DataMarkers		
Configure	Configure Data Marker	M
DisableAll	Disable All Data Markers	M
Count	Data Marker Count	P
Name	Data Marker Name	P
Item		P
Amplitude	Data Marker Amplitude	P
BitPosition	Data Marker Bit Position	P
Delay	Data Marker Delay	P
Destination	Data Marker Destination	P
Polarity	Data Marker Polarity	P
SourceChannel	Data Marker Source Channel	P
SparseMarkers		
Configure	Configure Sparse Marker	M
DisableAll	Disable All Sparse Markers	M
Count	Sparse Marker Count	P
Name	Sparse Marker Name	P
Item		P
Get Indexes	Get Sparse Marker Indexes	M

Table 33-3. IviFgen COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Set Indexes	Set Sparse Marker Indexes	M
Amplitude	Sparse Marker Amplitude	P
Delay	Sparse Marker Delay	P
Destination	Sparse Marker Destination	P
Polarity	Sparse Marker Polarity	P
WfmHandle	Sparse Marker Waveform Handle	P
StandardWaveform		
Configure	Configure Standard Waveform	M
Amplitude	Amplitude	P
DCOffset	DC Offset	P
DutyCycleHigh	Duty Cycle High	P
Frequency	Frequency	P
StartPhase	Start Phase	P
Waveform	Waveform	P
Arbitrary		
ClearMemory	Clear Arbitrary Memory	M
BinaryAlignment	Binary Alignment	P
DataMask	Data Mask	P
Gain	Arbitrary Gain	P
Offset	Arbitrary Offset	P
SampleBitResolution	Sample Bit Resolution	P
SampleRate	Arbitrary Sample Rate	P
Waveform		
Clear	Clear Arbitrary Waveform	M
Configure	Configure Arbitrary Waveform	M
Create	Create Arbitrary Waveform	M
CreateChannelWaveform	Create Channel Arbitrary Waveform	M
CreateChannelWaveformInt16	Create Channel Arbitrary Waveform Int16	M
CreateChannelWaveformInt32	Create Channel Arbitrary Waveform Int32	M
Frequency	Arbitrary Frequency	P
Handle	Waveform Handle	P
NumberWaveformsMax	Number Waveforms Max	P
Quantum	Waveform Quantum	P
SizeMax	Waveform Size Max	P
SizeMax64	Waveform Size Max 64	P
SizeMin	Waveform Size Min	P
SizeMin64	Waveform Size Min 64	P

Table 33-3. IviFgen COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Sequence		
Clear	Clear Arbitrary Sequence	M
Configure	Configure Arbitrary Sequence	M
Create	Create Arbitrary Sequence	M
DepthMax	Sequence Depth Max	P
Handle	Sequence Handle	P
LengthMax	Sequence Length Max	P
LengthMin	Sequence Length Min	P
LoopCountMax	Loop Count Max	P
NumberSequencesMax	Number Sequences Max	P
AM		
ConfigureInternal	Configure AM Internal	M
Enabled	AM Enabled	P
InternalDepth	AM Internal Depth	P
InternalFrequency	AM Internal Frequency	P
InternalWaveform	AM Internal Waveform	P
Source	AM Source	P
FM		
ConfigureInternal	Configure FM Internal	M
Enabled	FM Enabled	P
InternalDeviation	FM Internal Deviation	P
InternalFrequency	FM Internal Frequency	P
InternalWaveform	FM Internal Waveform	P
Source	FM Source	P
SampleClock		
Source	Sample Clock Source	P
OutputEnabled	Sample Clock Output Enabled	P

33.2.1 IviFgen COM Interfaces

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

IIviFgenOutput
IIviFgenOutput2
IIviFgenStandardWaveform
IIviFgenArbitrary
IIviFgenArbitrary2
IIviFgenArbitraryWaveform
IIviFgenArbitraryWaveform2
IIviFgenArbitrarySequence
IIviFgenArbitrarySequence2
IIviFgenTrigger
IIviFgenTrigger2
IIviFgenAM
IIviFgenFM
IIviFgenTriggerStart
IIviFgenTriggerStop
IIviFgenTriggerHold
IIviFgenTriggerResume
IIviFgenTriggerAdvance
IIviFgenDataMarkers
IIviFgenSparseMarkers
IIviFgenSampleClock

Table **33-4. IviFgen COM Interface** GUIDs lists the COM interfaces that this specification defines
and their GUIDs.

Table 33-4. IviFgen COM Interface GUIDs

Interface	GUID
IIviFgen	{47ed521a-a398-11d4-ba58-000064657374}
IIviFgen2	
IIviFgenOutput	{47ed521b-a398-11d4-ba58-000064657374}
IIviFgenOutput2	
IIviFgenStandardWaveform	{47ed521c-a398-11d4-ba58-000064657374}
IIviFgenArbitrary	{47ed521d-a398-11d4-ba58-000064657374}
IIviFgenArbitrary2	
IIviFgenArbitraryWaveform	{47ed521e-a398-11d4-ba58-000064657374}
IIviFgenArbitraryWaveform2	
IIviFgenArbitrarySequence	{47ed521f-a398-11d4-ba58-000064657374}
IIviFgenArbitrarySequence2	
IIviFgenTrigger	{47ed5220-a398-11d4-ba58-000064657374}
IIviFgenTrigger2	
IIviFgenAM	{47ed5221-a398-11d4-ba58-000064657374}
IIviFgenFM	{47ed5222-a398-11d4-ba58-000064657374}
IIviFgenTriggerStart	

Table 33-4. IviFgen COM Interface GUIDs

Interface	GUID
IIviFgenTriggerStop	
IIviFgenTriggerHold	
IIviFgenTriggerResume	
IIviFgenTriggerAdvance	
IIviFgenDataMarkers	
IIviFgenSparseMarkers	
IIviFgenSampleClock	

33.2.2 IviFgen COM Category

The IviFgen class COM Category shall be "IviFgen", and the Category ID (CATID) shall be {47ed5156-a398-11d4-ba58-000064657374}.

33.3 IviFgen C Function Hierarchy

The IviFgen class function hierarchy is shown in the following table. The full IviFgen C Function Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *C Inherent Capabilities* of *IVI-3.2: Ineherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 33-5. IviFgen C Function Hierarchy

Name or Class	Function Name
Configuration	
General	
Configure Output Mode	IviFgen_ConfigureOutputMode
Configure Operation Mode	IviFgen_ConfigureOperationMode
Configure Output Enabled	IviFgen_ConfigureOutputEnabled
Configure Output Impedance	IviFgen_ConfigureOutputImpedance
Configure Output Mode	IviFgen_ConfigureOutputMode
Configure Reference Clock Source	IviFgen_ConfigureRefClockSource
Configure Sample Clock	IviFgen_ConfigureSampleClock
Configure Sample Clock Output Enabled	IviFgen_ConfigureSampleClockOutputEnabled
Standard Waveform	
Configure Standard Waveform	IviFgen_ConfigureStandardWaveform
Arbitrary Waveform	
Clear Arbitrary Waveform	IviFgen_ClearArbWaveform
Configure Arbitrary Waveform	IviFgen_ConfigureArbWaveform
Configure Sample Rate	IviFgen_ConfigureSampleRate
Create Arbitrary Waveform	IviFgen_CreateArbWaveform
Create Channel Arbitrary Waveform	IviFgen_CreateChannelArbWaveform
Create Channel Arbitrary Waveform Int16	IviFgen_CreateChannelArbWaveform16
Create Channel Arbitrary Waveform Int32	IviFgen_CreateChannelArbWaveform32
Query Arbitrary Waveform Capabilities	IviFgen_QueryArbWfmCapabilities
Query Arbitrary Waveform Capabilities 64	IviFgen_QueryArbWfmCapabilities64
Arbitrary Waveform Frequency	
Configure Arbitrary Frequency	IviFgen_ConfigureArbFrequency
Arbitrary Sequence	
Clear Arbitrary Memory	IviFgen_ClearArbMemory
Clear Arbitrary Sequence	IviFgen_ClearArbSequence
Configure Arbitrary Sequence	IviFgen_ConfigureArbSequence

Create Arbitrary Sequence	`IviFgen_CreateArbSequence`
Query Arbitrary Sequence Capabilities	`IviFgen_QueryArbSeqCapabilities`
Trigger	
Configure Trigger Source	`IviFgen_ConfigureTriggerSource`
Configure Start Trigger	`IviFgen_ConfigureStartTrigger`
Configure Stop Trigger	`IviFgen_ConfigureStopTrigger`
Configure Hold Trigger	`IviFgen_ConfigureHoldTrigger`
Configure Resume Trigger	`IviFgen_ConfigureResumeTrigger`
Configure Advance Trigger	`IviFgen_ConfigureAdvanceTrigger`
Internal Trigger	
Configure Internal Trigger Rate	`IviFgen_ConfigureInternalTriggerRate`
Burst	
Configure Burst Count	`IviFgen_ConfigureBurstCount`
Data Markers	
Configure Data Marker	`IviFgen_ConfigureDataMarker`
Get Data Marker Name	`IviFgen_GetDataMarkerName`
Disable All Data Markers	`IviFgen_DisableAllDataMarkers`
Sparse Markers	
Configure Sparse Marker	`IviFgen_ConfigureSparseMarker`
Get Sparse Marker Name	`IviFgen_GetSparseMarkerName`
Get Sparse Marker Indexes	`IviFgen_GetSparseMarkerIndexes`
Set Sparse Marker Indexes	`IviFgen_SetSparseMarkerIndexes`
Disable All Sparse Markers	`IviFgen_DisableAllSparseMarkers`
Amplitude Modulation	
Configure AM Enabled	`IviFgen_ConfigureAMEnabled`
Configure AM Internal	`IviFgen_ConfigureAMInternal`
Configure AM Source	`IviFgen_ConfigureAMSource`
Frequency Modulation	
Configure FM Enabled	`IviFgen_ConfigureFMEnabled`
Configure FM Internal	`IviFgen_ConfigureFMInternal`
Configure FM Source	`IviFgen_ConfigureFMSource`
Action	
Abort Generation	`IviFgen_AbortGeneration`
Initiate Generation	`IviFgen_InitiateGeneration`
Send Software Stop Trigger	`IviFgen_SendSoftwareStopTrigger`
Send Software Hold Trigger	`IviFgen_SendSoftwareHoldTrigger`
Send Software Resume Trigger	`IviFgen_SendSoftwareResumeTrigger`
Send Software Advance Trigger	`IviFgen_SendSoftwareAdvanceTrigger`
Send Software Trigger	`IviFgen_SendSoftwareTrigger`
Utility	

Get Channel Name	IviFgen_GetChannelName

33.4 IviFgen Obsolete Function Names

The following function names are reserved by the IviFgen specification 1.0. Future versions of this specification cannot use these names:

- `IviFgen_EnableOutput`

- `IviFgen_DisableOutput`

- `IviFgen_ConfigureTrigger`

- `IviFgen_EnableAM`

- `IviFgen_DisableAM`

- `IviFgen_EnableFM`

- `IviFgen_DisableFM`

- `IviFgen_SendSWTrigger`

33.5 IviFgen C Attribute Hierarchy

The IviFgen class attribute hierarchy is shown in the following table. The full IviFgen C Attribute Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *C Inherent Capabilities* of *IVI-3.2: Ineherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 33-6. IviFgen C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Output	
Output Count	IVIFGEN_ATTR_CHANNEL_COUNT
Operation Mode	IVIFGEN_ATTR_OPERATION_MODE
Output Enabled	IVIFGEN_ATTR_OUTPUT_ENABLED
Output Impedance	IVIFGEN_ATTR_OUTPUT_IMPEDANCE
Output Mode	IVIFGEN_ATTR_OUTPUT_MODE
Reference Clock Source	IVIFGEN_ATTR_REF_CLOCK_SOURCE
Sample Clock Source	IVIFGEN_ATTR_SAMPLE_CLOCK_SOURCE
Sample Clock Output Enabled	IVIFGEN_ATTR_SAMPLE_CLOCK_OUTPUT_ENABLED
Terminal Configuration	IVIFGEN_ATTR_TERMINAL_CONFIGURATION
Standard Waveform	
Amplitude	IVIFGEN_ATTR_FUNC_AMPLITUDE
DC Offset	IVIFGEN_ATTR_FUNC_DC_OFFSET
Duty Cycle High	IVIFGEN_ATTR_FUNC_DUTY_CYCLE_HIGH
Frequency	IVIFGEN_ATTR_FUNC_FREQUENCY
Start Phase	IVIFGEN_ATTR_FUNC_START_PHASE
Waveform	IVIFGEN_ATTR_FUNC_WAVEFORM
Arbitrary	
Arbitrary Gain	IVIFGEN_ATTR_ARB_GAIN
Arbitrary Offset	IVIFGEN_ATTR_ARB_OFFSET
Arbitrary Sample Rate	IVIFGEN_ATTR_ARB_SAMPLE_RATE
Waveform	
Arbitrary Waveform Handle	IVIFGEN_ATTR_ARB_WAVEFORM_HANDLE
Number Waveforms Max	IVIFGEN_ATTR_MAX_NUM_WAVEFORMS
Waveform Size Max	IVIFGEN_ATTR_MAX_WAVEFORM_SIZE
Waveform Size Max 64	IVIFGEN_ATTR_MAX_WAVEFORM_SIZE64
Waveform Size Min	IVIFGEN_ATTR_MIN_WAVEFORM_SIZE
Waveform Size Min 64	IVIFGEN_ATTR_MIN_WAVEFORM_SIZE64
Waveform Quantum	IVIFGEN_ATTR_WAVEFORM_QUANTUM
Arbitrary Frequency	IVIFGEN_ATTR_ARB_FREQUENCY

Table 33-6. IviFgen C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Binary Alignment	IVIFGEN_ATTR_BINARY_ALIGNMENT
Sample Bit Resolution	IVIFGEN_ATTR_SAMPLE_BIT_RESOLUTION
Data Mask	IVIFGEN_ATTR_OUTPUT_DATA_MASK
Sequence	
Arbitrary Sequence Handle	IVIFGEN_ATTR_ARB_SEQUENCE_HANDLE
Number Sequences Max	IVIFGEN_ATTR_MAX_NUM_SEQUENCES
Loop Count Max	IVIFGEN_ATTR_MAX_LOOP_COUNT
Sequence Length Max	IVIFGEN_ATTR_MAX_SEQUENCE_LENGTH
Sequence Length Min	IVIFGEN_ATTR_MIN_SEQUENCE_LENGTH
Sequence Depth Max	IVIFGEN_ATTR_SEQUENCE_DEPTH_MAX
Trigger	
Trigger Source	IVIFGEN_ATTR_TRIGGER_SOURCE
Internal Trigger Rate	IVIFGEN_ATTR_INTERNAL_TRIGGER_RATE
Burst Count	IVIFGEN_ATTR_BURST_COUNT
Start Trigger	
Start Trigger Delay	IVIFGEN_ATTR_START_TRIGGER_DELAY
Start Trigger Slope	IVIFGEN_ATTR_START_TRIGGER_SLOPE
Start Trigger Source	IVIFGEN_ATTR_START_TRIGGER_SOURCE
Start Trigger Threshold	IVIFGEN_ATTR_START_TRIGGER_THRESHOLD
Stop Trigger	
Stop Trigger Delay	IVIFGEN_ATTR_STOP_TRIGGER_DELAY
Stop Trigger Slope	IVIFGEN_ATTR_STOP_TRIGGER_SLOPE
Stop Trigger Source	IVIFGEN_ATTR_STOP_TRIGGER_SOURCE
Stop Trigger Threshold	IVIFGEN_ATTR_STOP_TRIGGER_THRESHOLD
Hold Trigger	
Hold Trigger Delay	IVIFGEN_ATTR_HOLD_TRIGGER_DELAY
Hold Trigger Slope	IVIFGEN_ATTR_HOLD_TRIGGER_SLOPE
Hold Trigger Source	IVIFGEN_ATTR_HOLD_TRIGGER_SOURCE
Hold Trigger Threshold	IVIFGEN_ATTR_HOLD_TRIGGER_THRESHOLD
Resume Trigger	
Resume Trigger Delay	IVIFGEN_ATTR_RESUME_TRIGGER_DELAY
Resume Trigger Slope	IVIFGEN_ATTR_RESUME_TRIGGER_SLOPE
Resume Trigger Source	IVIFGEN_ATTR_RESUME_TRIGGER_SOURCE
Resume Trigger Threshold	IVIFGEN_ATTR_RESUME_TRIGGER_THRESHOLD
Advance Trigger	

Table 33-6. IviFgen C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Advance Trigger Delay	IVIFGEN_ATTR_ADVANCE_TRIGGER_DELAY
Advance Trigger Slope	IVIFGEN_ATTR_ADVANCE_TRIGGER_SLOPE
Advance Trigger Source	IVIFGEN_ATTR_ADVANCE_TRIGGER_SOURCE
Advance Trigger Threshold	IVIFGEN_ATTR_ADVANCE_TRIGGER_THRESHOLD
Data Markers	
Data Marker Amplitude	IVIFGEN_ATTR_DATAMARKER_AMPLITUDE
Data Marker Bit Position	IVIFGEN_ATTR_DATAMARKER_BIT_POSITION
Data Marker Count	IVIFGEN_ATTR_DATAMARKER_COUNT
Data Marker Delay	IVIFGEN_ATTR_DATAMARKER_DELAY
Data Marker Destination	IVIFGEN_ATTR_DATAMARKER_DESTINATION
Data Marker Polarity	IVIFGEN_ATTR_DATAMARKER_POLARITY
Data Marker Source Channel	IVIFGEN_ATTR_DATAMARKER_SOURCE_CHANNEL
Sparse Markers	
Sparse Marker Amplitude	IVIFGEN_ATTR_SPARSEMARKER_AMPLITUDE
Sparse Marker Count	IVIFGEN_ATTR_SPARSEMARKER_COUNT
Sparse Marker Delay	IVIFGEN_ATTR_SPARSEMARKER_DELAY
Sparse Marker Destination	IVIFGEN_ATTR_SPARSEMARKER_DESTINATION
Sparse Marker Polarity	IVIFGEN_ATTR_SPARSEMARKER_POLARITY
Sparse Marker Waveform Handle	IVIFGEN_ATTR_SPARSEMARKER_WFMHANDLE
AM	
AM Enabled	IVIFGEN_ATTR_AM_ENABLED
AM Internal Depth	IVIFGEN_ATTR_AM_INTERNAL_DEPTH
AM Internal Frequency	IVIFGEN_ATTR_AM_INTERNAL_FREQUENCY
AM Internal Waveform	IVIFGEN_ATTR_AM_INTERNAL_WAVEFORM
AM Source	IVIFGEN_ATTR_AM_SOURCE
FM	
FM Enabled	IVIFGEN_ATTR_FM_ENABLED
FM Internal Deviation	IVIFGEN_ATTR_FM_INTERNAL_DEVIATION
FM Internal Frequency	IVIFGEN_ATTR_FM_INTERNAL_FREQUENCY
FM Internal Waveform	IVIFGEN_ATTR_FM_INTERNAL_WAVEFORM
FM Source	IVIFGEN_ATTR_FM_SOURCE

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 IviFgen 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 Initialize, or Reset functions execute. This assumes that the extension capability groups remain disabled until the application program explicitly uses them. For the large majority of instruments, this assumption is true.

Under certain conditions, a specific driver might have to implement a more complex approach. For some instruments, configuring a capability group might affect instrument settings that correspond to an unused extension capability group. If these instrument settings affect the behavior of the instrument, then this might result in an interchangeability problem. If this can occur, the specific driver must take appropriate action so that the instrument settings that correspond to the unused extension capability group do not affect the behavior of the instrument when the application program performs an operation that might be affected by those settings.

The remainder of this section recommends attribute values that effectively disable each extension capability group.

Disabling the IviFgenStdFunc Extension Group

The IviFgenStdFunc extension group affects the instrument behavior only when the Output Mode attribute is set to Output Function. Therefore, this specification does not recommend attribute values that disable the IviFgenStdFunc extension group.

Disabling the IviFgenArbWaveform Extension Group

The IviFgenArbWaveform extension group affects the instrument behavior only when the Output Mode attribute is set to Output Arbitrary. Therefore, this specification does not recommend attribute values that disable the IviFgenArbWaveform extension group.

Disabling the IviFgenArbFrequency Extension Group

The IviFgenArbFrequency extension group affects the instrument behavior only when the Output Mode attribute is set to Output Arbitrary, and the user has accessed the Arbitrary Frequency attribute. Therefore, this specification does not recommend attribute values that disable the IviFgenArbFrequency extension group.

Disabling the IviFgenArbSequence Extension Group

The IviFgenArbSequence extension group affects the instrument behavior only when the Output Mode attribute is set to Output Sequence. Therefore, this specification does not recommend attribute values that disable the IviFgenArbSequence extension group.

Disabling the IviFgenTrigger Extension Group

The IviFgenTrigger extension group affects the instrument behavior only when the Operation Mode attribute is set to Operate Burst. Therefore, this specification does not recommend attribute values that disable the IviFgenTrigger extension group.

Disabling the IviFgenStartTrigger Extension Group

The IviFgenStartTrigger extension group affects the instrument behavior only when the start trigger source property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenStartTrigger extension group.

Disabling the IviFgenStopTrigger Extension Group

The IviFgenStopTrigger extension group affects the instrument behavior only when the stop trigger source property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenStopTrigger extension group.

Disabling the IviFgenHoldTrigger Extension Group

The IviFgenHoldTrigger extension group affects the instrument behavior only when the hold trigger source property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenHoldTrigger extension group.

Disabling the IviFgenResumeTrigger Extension Group

The IviFgenResumeTrigger extension group affects the instrument behavior only when the resume trigger source property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenResumeTrigger extension group.

Disabling the IviFgenAdvanceTrigger Extension Group

The IviFgenAdvanceTrigger extension group affects the instrument behavior only when the advance trigger source property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenAdvanceTrigger extension group.

Disabling the IviFgenInternalTrigger Extension Group

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

Disabling the IviFgenSoftwareTrigger Extension Group

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

Disabling the IviFgenBurst Extension Group

The IviFgenBurst extension group affects the instrument behavior only when the Operation Mode attribute is set to Operation Burst. Therefore, this specification does not recommend attribute values that disable the IviFgenBurst extension group.

Disabling the IviFgenModulateAM Extension Group

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

Table A.1. Values for Disabling the IviFgenModulateAM Extension Group

Attribute	Value
AM Enabled	False

Disabling the IviFgenModulateFM Extension Group

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

Table A.2. Values for Disabling the IviFgenModulateFM Extension Group

Attribute	Value
FM Enabled	False

Disabling the IviFgenSampleClock Extension Group

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

Table. Values for Disabling the IviFgenSampleClock Extension Group

Attribute	Value
Sample Clock Source	Internal

Disabling the IviFgenTerminalConfiguration Extension Group

The IviFgenTerminalConfiguration extension group ???. Therefore, this specification does not recommend attribute values that disable the IviFgenTerminalConfiguration extension group.

Disabling the IviFgenArbWfmSize64 Extension Group

The IviFgenArbWfmSize64 extension group only has read only attributes and therefore does not impact instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviFgenArbWfmSize64 extension group.

Disabling the IviFgenArbChannelWaveform Extension Group

The IviFgenArbChannelWaveform extension group affects the instrument behavior only when the Output Mode attribute is set to Output Arbitrary. Therefore, this specification does not recommend attribute values that disable the IviFgenArbChannelWaveform extension group.

Disabling the IviFgenArbWfmBinary Extension Group

The IviFgenArbWfmBinary extension group affects the instrument behavior only when the Output Mode attribute is set to Output Arbitrary. Therefore, this specification does not recommend attribute values that disable the IviFgenWfmBinary extension group.

Disabling the IviFgenDataMarker Extension Group

The IviFgenDataMarker extension group affects the instrument behavior only when the data marker destination property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenDataMarker extension group.

Disabling the IviFgenDataMask Extension Group

The IviFgenArbDataMask extension group affects the instrument behavior only when the DataMask attribute has bit-values of 0. Therefore, this specification does not recommend attribute values that disable the IviFgenArbDataMask extension group.

Disabling the IviFgenSparseMarker Extension Group

The IviFgenSparseMarker extension group affects the instrument behavior only when the sparse marker destination property is set to something other than None. Therefore, this specification does not recommend attribute values that disable the IviFgenSparseMarker extension group.

Disabling the IviFgenArbSeqDepth Extension Group

The IviFgenArbSeqDepth extension group only has read only attributes and therefore does not impact instrument behavior. Therefore, this specification does not recommend attribute values that disable the IviFgenArbSeqDepth extension group.

A.3 Query Instrument Status

Based on the value of Query Instrument Status, the 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 Generation and Abort Generation. 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.

A.4 Special Considerations for Initiate Generation and Abort Generation

The Abort Generation and Initiate Generation functions cause the function generator to move between a Configuration State and a Generation State. If the current state of the instrument affects whether or not attributes can be configured, the driver shall implement a mechanism to track the instrument's state.

One solution is to create a hidden Boolean attribute, Generating. Any attribute or function whose behavior depends on the instrument's state can reference this attribute's value to determine it. The Initiate Generation function would set this attribute to True and the Abort Generation function would set it to False. If the user attempts to change the configuration of the function generator while the Generating attribute is set to True (in other words, while the function generator is in the process of generating a waveform) the driver should halt signal generation, changes the instrument's configuration, and re-start signal generation. This allows function generators that have to be explicitly armed to behave like function generators that are constantly generating, thus helping to achieve interchangeability.

A.5 Special Considerations for Output Mode

On function generators that support multiple output modes, some attributes may not be configurable when the user changes from one output mode to another. For example, it might not be possible to set the standard waveform amplitude on a channel while in the arbitrary sequence output mode. If an attribute or function's behavior relies on the function generator being in a specific mode, the driver should return the error an invalid configuration error. This indicates to the user that the current output mode does not allow a certain action to be taken. This error is typically used for the attributes and functions of the IviFgenStdFunc, IviFgenArbWfm, and IviFgenArbSeq extensions, as these extensions provide different interfaces for configuring the function generator's output.

Some multi-channel function generators support the simultaneous use of different output modes on multiple channels. However, the Output Made attribute is not channel based. If a multi-channel function generator supports channel based output modes, the driver developer can implement an instrument-specific, multi-channel output mode attribute. Setting the class-defined attribute, Output Mode, would change the output mode of all channels. Setting the instrument-specific attribute would change the output mode only on the specified channel. If the user attempts to read the class-defined attribute, and the output modes of all the channels are not set to the same value, the instrument driver should return an error code. This indicates to the user that use of the instrument-specific attribute has caused the instrument to be in a state that is not compliant with the expected behavior of the class-defined attribute.

A.6 Special Considerations for Output Impedance

The Output Impedance attribute allows the driver to specify the function generator's output impedance. Not all function generators have variable output impedance.
However, some function generators that have fixed output impedance can be configured to operate differently based on the system (i.e., load) impedance. A change in system impedance affects how these instruments determine the voltage at the instrument's output terminals. These instruments, when properly configured, can set the voltage at the output terminal to be the programmed values even though the system impedance does not match the output impedance.

A driver for a function generator that can be configured based on the system impedance can accept different values for this attribute even if the instrument has fixed impedance. The value of output impedance specified should be considered to be equal to the system impedance. The exception to this is the case of very high system impedance. For the case of high system impedance, an output impedance of 0.0 should be used.

Drivers for function generators in which neither the output impedance nor the system impedance can be specified should accept only the value of the function generator's fixed output impedance.

A.7 Special Considerations for Create Arbitrary Waveform and Create Arbitrary Sequence

The process of creating an arbitrary waveform or sequence and downloading it to a function generator can be very complicated. The following overall procedure is used to create a new waveform or sequence:

- Verify that the instrument is in a configurable state.

- Verify that the instrument's output mode allows a new arbitrary waveform or sequence to be downloaded.

- Verify that the waveform or sequence is of valid size, that its data is valid, and that the instrument has memory available for it.

- Obtain a new arbitrary waveform or sequence handle.

- If the instrument driver is not simulating, download the arbitrary data.

- Update any internal parts of the driver to reflect the creation of a new arbitrary waveform or sequence.

- Return the new handle to the user

A.8 Special Considerations for Clear Arbitrary Waveform, Clear Arbitrary Sequence, and Clear Arbitrary Memory

The process of clearing arbitrary waveform and sequence memory from a function generator can be very complicated. The following overall procedure is used to clear arbitrary data:

- Verify that the instrument is in a configurable state.

- Verify that the instrument's output mode allows arbitrary waveform or sequence data to be cleared.

- Verify that the waveform or sequence to be cleared actually exists.

- Verify that the waveform or sequence is not currently configured for generation.

- Verify that a waveform to be cleared is not currently part of an existing sequence.

- If the instrument driver is not simulating, remove the arbitrary data from the instrument.

- Update any internal parts of the driver to reflect the removal of an arbitrary waveform or sequence.

Function generators typically do not allow the modification of arbitrary waveform or sequence memory while they generate an arbitrary waveform. When this restriction exists, the specific instrument driver shall not allow a waveform or sequence to be cleared while it is being generated. In addition, arbitrary waveforms shall not be cleared while a sequence that uses that waveform exists. The result of these restrictions is that in order to clear all arbitrary waveforms and sequences, the instrument driver shall be configured to generate something other than an arbitrary waveform or sequence.

Appendix B   Interchangeability Checking Rules

B.1 Introduction

IVI drivers have a feature called interchangeability checking. Interchangeability checking returns a warning when it encounters a situation were 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 Initiate Generation

B.3 Interchangeability Checking Rules

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the base capability group. In addition, interchangeability checking is performed on extension capability groups for which the user has ever set any of the attributes of the group. If the user has never set any attributes of an extension capability group, interchangeability checking is not performed on that group.

In general interchangeability warnings are generated if the following conditions are encountered:

- An attribute that affects the behavior of the instrument is not in a state that the user specifies.

- The user sets a class driver defined attribute to an instrument-specific value.

- The user configures the value of an attribute that the class defines as read-only. In a few cases the class drivers define read-only attributes that specific drivers might implement as read/write.

The remainder of this section defines additional rules and exceptions for each capability group.

IviFgenBase Capability Group

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

IviFgenStdFunc Capability Group

1. If the Output Mode attribute is not set to Output Function, no attributes in the IviFgenStdFunc extension group are required to be in a user specified state.

2. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenStdFunc extension group are required to be in a user-specified state on that channel

3. If the Waveform attribute is set to Waveform DC on a channel, the following attributes are not required to be in a user specified state on that channel:

   - Amplitude

   - Frequency

   - Start Phase

4. If the Waveform attribute is not set to Waveform Square on a channel, the Duty Cycle High attribute is not required to be in a user specified state on that channel.

IviFgenArbWaveform Capability Group

1. If the Output Mode attribute is not set to Output Arbitrary, no attributes in the IviFgenArbWaveform extension group are required to be in a user specified state.

2. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenArbWaveform extension group are required to be in a user-specified state on that channel.

IviFgenArbSequence Capability Group

1. If the Output Mode attribute is not set to output Sequence, no attributes in the IviFgenArbSequence extension group are required to be in a user specified state.

2. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenArbSequence extension group are required to be in a user-specified state on that channel.

IviFgenTrigger Capability Group

1. If the Operation Mode attribute is not set to Operate Burst, no attributes in the IviFgenTrigger extension group are required to be in a user specified state.

IviFgenStartTrigger Capability Group

1. If the Start Trigger Source attribute is set to None, no attributes in the IviFgenStartTrigger extension group are required to be in a user specified state.

IviFgenStopTrigger Capability Group

1. If the Stop Trigger Source attribute is set to None, no attributes in the IviFgenStopTrigger extension group are required to be in a user specified state.

IviFgenHoldTrigger Capability Group

1. If the Hold Trigger Source attribute is set to None, no attributes in the IviFgenHoldTrigger extension group are required to be in a user specified state.

IviFgenResumeTrigger Capability Group

1. If the Resume Trigger Source attribute is set to None, no attributes in the IviFgenResumeTrigger extension group are required to be in a user specified state.

IviFgenAdvanceTrigger Capability Group

1. If the Advance Trigger Source attribute is set to None, no attributes in the IviFgenAdvanceTrigger extension group are required to be in a user specified state.

IviFgenInternalTrigger Capability Group

1. If the Trigger Source attribute is not set to Internal Trigger, no attributes in the IviFgenInternalTrigger extension group are required to be in a user specified state.

IviFgenSoftwareTrigger Capability Group

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

IviFgenBurst Capability Group

1. If the Operation Mode attribute is not set to Operate Burst, no attributes in the IviFgenBurst extension group are required to be in a user specified state.

2. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenBurst extension group are required to be in a user-specified state on that channel.

IviFgenModulateAM Capability Group

1. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenModulateAM extension group are required to be in a user-specified state on that channel.

2. If the AM Enabled attribute is not set to True on a channel, the following attributes are not required to be ins a user specified state on that channel.

   - AM Source

IviFgenModulateFM Capability Group

1. If the Output Enabled attribute is set to False on a channel, no attributes in the IviFgenModulateFM extension group are required to be in a user-specified state on that channel.

2. If the FM Enabled attribute is not set to True on a channel, the following attributes are not required to be in a user specified state on that channel.

   - FM Source

IviFgenSampleClock Capability Group

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

IviFgenTerminalConfiguration Capability Group

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

IviFgenArbWfmSize64 Capability Group

No additional interchangeability rules or exceptions are defined for the IviFgenArbWfmSize64 capability group.

IviFgenArbChannelWfm Capability Group

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

IviFgenArbWfmBinary Capability Group

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

IviFgenDataMarker Capability Group

1. If the Data Marker Destination attribute is set to None, no attributes in the IviFgenDataMarker extension group are required to be in a user specified state.

IviFgenArbDataMask Capability Group

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

IviFgenSparseMarker Capability Group

1. If the Sparse Marker Destination attribute is set to None, no attributes in the IviFgenSparseMarker extension group are required to be in a user specified state.

IviFgenArbSeqDepth Capability Group

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