IVI-3.15: IviLxiSync Specification

August 23, 2018
Revision 2.0

Important Information

The IviLxiSync Specification (IVI-3.15) 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`.

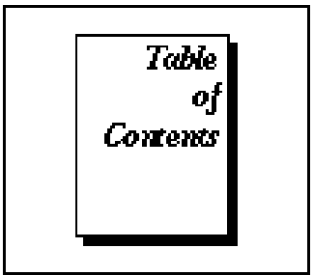
Warranty

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.



IviLxiSync Interface Specification

Revision History

Table 1. Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 1.0	April 4, 2006	First approved version. Initial Revision
	September 20, 2006	Editorial revision to 1.0 3.1 Clarify use of ARM is optional and recommend that no ARM config needed if not used 2.1.3, 2.1.4, 4.2.10 Clarify namespace overlap between TriggerAlarm, TriggerSource 2.1.1, 2.1.2 Clarify namespace between ArmAlarm and ArmSource 2.1.6 Pointed out that LXI reserves identifiers beginning with "LXI" 2.1.9 Pointed out that Custom capability groups are optional 4.1 Pointed out that trigger defaults should allow Initiate to start measurement 3.2.15, 4.2.17 Corrected prose regarding filter related to implied monitoring of send port. Eliminated syntactically incorrect example. 5.2.2 Pointed out that LXI reserves identifiers beginning with "LXI" 5.2.7 Corrected table and pointed out applicability beyond LXI 3.2.6, 3.2.7, 4.2.6, 4.2.7 Clarification of the behavior of Arm/Trigger count and interaction with Arm/Trigger period.
	June 19, 2008	Editorial Revision: 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 2.0	June 9, 2010	Incorporated .NET
Revision 2.0	October 7, 2014	Editorial Change: Change the COM defined values listed in sections 3.2.13 and 4.2.14 to match the names in section 9 and the type library.
Revision 2.0	August 23, 2018	Editorial Change: Errant comments about case preserving and case insensitive were removed.

API Versions

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

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

1. Overview of the IviLxiSync Interface Specification

1.1 Introduction

This section introduces the *IviLxiSync Interface Specification*. This section summarizes the *IviLxiSync Interface 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:

- IviLxiSync API Overview
- References
- The definitions of terms and acronyms

1.2 IviLxiSync Interface Overview

This specification defines the API for controlling the arming, triggering, and event functionality of LXI devices. The functionality defined in the *IviLxiSync Interface Specification* is pertinent to LXI Class A and Class B devices, but is independent of the IVI instrument class supported by the device, if any. The *IviLxiSync Interface Specification* conceptualizes an LXI device as an instrument that can "listen" for LXI trigger bus events or LXI LAN-based events and can fire such events. Inbound events control the device's arming and triggering subsystem for performing measurements and other operations. A device may include conventional inbound events in addition to the LXI events. Outbound events can be used to notify other LXI devices of specific conditions.

The IviLxiSync API is divided into the following five subsystems:

- IviLxiSyncArm
- IviLxiSyncTrigger
- IviLxiSyncEvent
- IviLxiSyncEventLog
- IviLxiSyncTime

The IviLxiSyncArm subsystem controls when triggers are accepted.

This IviLxiSyncTrigger subsystem controls when the LXI device triggers a measurement or other operation.

The IviLxiSyncEvent subsystem controls when the LXI device signals specific conditions to other LXI devices.

The IviLxiSyncEventLog subsystem provides access to the event logging features of an LXI device.

The IviLxiSyncTime subsystem exposes functionality for access to the 1588 timebase of the LXI bus.

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-3.4: API Style Guide
- IVI-3.18: IVI.NET Utility Classes and Interfaces Specification
- IVI-5.0: Glossary

- LXI Standard

1.4 Definitions of Terms and Acronyms

Refer to *IVI-5: Glossary* and to *LXI Standards Definition: Appendix A – Glossary of Terms* for a description of the terms and acronyms used in this specification. This specification does not define any additional terms.

1.5 IviLxiSync Device Model

This section describes the logical model that is controlled by the IviLxiSync API.

1.5.1 High-Level Device Model

The diagram below shows a high-level view of the major components of an LXI device from a trigger and synchronization perspective. The three main IviLxiSync subsystems controlled by the interface in this specification (IviLxiSyncArm, IviLxiSyncTrigger, and IviLxiSyncEvent) are shown in the diagram as the "Arm Logic", "Trigger Logic", and "Event Logic" blocks, respectively.

The Arm Logic and Trigger Logic blocks receive input from both the LXI trigger bus and the LAN input registers. LXI-defined LAN events (LAN0..7), as well as custom LAN events can drive the arm-trigger logic. The output of the arm-trigger logic drives the Arm-Trigger State Machine, which is detailed further in the following section.

The Event Logic controls signals that are emitted from the LXI device. It specifies which LXI devices are notified and what type of event notification is used. The Event Logic can drive any of the LXI trigger bus lines, any LXI-defined LAN event (LAN0..7) or a custom LAN event.

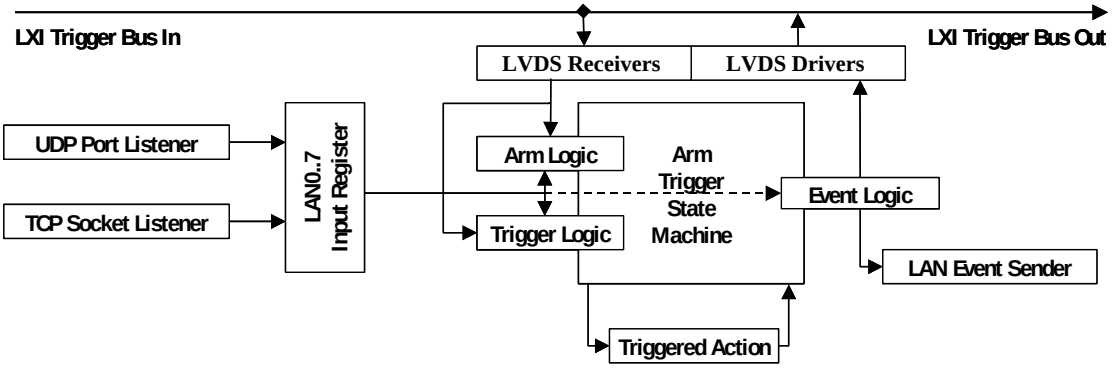


Figure 1. High-Level LXI Device Model

1.6 Integrating IviLxiSync With Existing Classes

The IviLxiSync API is used with some other instrument API that supplies the measurement or stimulus capabilities. This existing API could be an IVI Class or it could be a purely custom API. Usually, that other API will already define a trigger source attribute. That attribute could be either an enumerated value

or a string. Since the IviLxiSync API also includes a trigger source, the two trigger source attributes and corresponding queries need to be integrated.

To integrate the IviLxiSync API with existing API's, the following approach should be taken:

- The instrument should configure the trigger source to the value most recently configured by the customer, with the corresponding behavior model. That is, if the customer configures the IviLxiSync trigger source via the IviLxiSync API, the IviLxiSync API will set the source and govern the behavior model. If a class driver or specific driver API is used to set the trigger source (for instance, the DMM trigger source attribute), then the source specified via that API, and the corresponding behaviors, become active.

- When querying the trigger source on a given API, if the trigger source was set via that API, then the corresponding value is returned. If the value was set via the alternate API and no corresponding value is available, then the driver should generate an error.

- The IviLxiSync API sets the trigger source using a string. Instruments are therefore, free to support additional trigger sources through that API.

1.7 Implementing IviLxiSync for an IVI-C Instrument Driver

The IviLxiSync API is used in conjunction with existing IVI Foundation-defined API's for instrument drivers. The IviLxiSync API defines some C interface elements which, if standard IVI naming conventions were followed, would conflict with existing IVI Foundation Class and Inherent interfaces for IVI-C instrument drivers. When implementing IviLxiSync for an IVI-C driver, different naming guidelines must therefore be followed for the IviLxiSync interface elements:

If you are implementing the IviLxiSync API as part of an IVI-C specific driver's API:

- Replace "`IviLxiSync_" in method names with "<Prefix>_IviLxiSync"`, where <Prefix> is the actual driver prefix used for all specific driver function names.

Replace "`IVILXISYNC_ATTR`" in attribute identifiers with "<PREFIX>_ATTR_IVILXISYNC", where <PREFIX> is the actual driver prefix using uppercase characters.

Replace "IVILXISYNC_ERROR" in error code identifiers with "<PREFIX>_ERROR_IVILXISYNC", where <PREFIX> is the actual driver prefix using uppercase characters.

Replace "IVILXISYNC_VAL" in value identifiers with "<PREFIX>_VAL_IVILXISYNC", where <PREFIX> is the actual driver prefix using uppercase characters.

If you are implementing the IviLxiSync API as a common library to be shared between multiple specific driver API's:

Use the attribute, value, function, and error code identifiers with the "IviLxiSync" or "IVILXISYNC" prefixes as described in this specification.

1.8 Implementing IviLxiSync for an IVI-COM Instrument Driver

Guidelines for supporting the IviLxiSync API for an IVI-COM Instrument Driver are described in section *12.1.4. COM Interface Accessibility.*

2. IviLxiSync Repeated Capabilities

This specification makes extensive use of IVI repeated capabilities to model collections of arm and trigger alarms and sources, as well as outbound events. Each of the five repeated capabilities defined in Section 2.1, *Repeated Capability Names*, shall be implemented according to the standard IVI repeated capability standards laid out in *IVI 3.3-Standard Cross-Class Capabilities*. That specification, along with *IVI 3.4-API Style Guide*, describes three techniques for implementing IVI repeated capabilities – the *selector* technique, the *parameter* technique, and the *IVI-COM collection* technique. All of the repeated capabilities in the IviLxiSync API shall use the IVI-COM collection technique for IVI-COM interfaces and the parameter technique for IVI-C interfaces.

2.1 Repeated Capability Names

The IviLxiSync Specification defines five 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*, Section 4.3.9, *Repeated Capabilities*, and Section 5.9, *Repeated Capability Identifiers and Selectors*.

- IviLxiSyncArmAlarm
- IviLxiSyncArmSource
- IviLxiSyncTriggerAlarm
- IviLxiSyncTriggerSource
- IviLxiSyncEvent

2.1.1 IviLxiSyncArmSource

In the configuration store, the name for the LXI arm source repeated capability shall be "IviLxiSyncArmSource".

Although there is no intrinsic tie between Arm sources and Arm alarms, there is some potential for customer confusion since Trigger sources and Trigger alarms share the same namespace. Therefore, drivers should place Arm sources and Arm alarms in the same namespace for consistency.

2.1.2 IviLxiSyncArmAlarm

In the configuration store, the name for the LXI arm alarm repeated capability shall be "IviLxiSyncArmAlarm".

Although there is no intrinsic tie between Arm sources and Arm alarms, there is some potential for customer confusion since Trigger sources and Trigger alarms share the same namespace. Therefore, drivers should place Arm sources and Arm alarms in the same namespace for consistency.

2.1.3 IviLxiSyncTriggerAlarm

In the configuration store, the name for the LXI trigger alarm capability shall be "IviLxiSyncTriggerAlarm".

Note that the IviLxiSyncTriggerAlarm repeated capability group shares its namespace with IviLxiSyncTriggerSource. See section 2.1.4.

2.1.4 IviLxiSyncTriggerSource

In the configuration store, the name for the LXI trigger source capability shall be "IviLxiSyncTriggerSource".

Note that the IviLxiSyncTriggerSource repeated capability shares its namespace with IviLxiSyncTriggerAlarm. Therefore, once an Alarm has been defined, it may be used as a trigger source using the Trigger.TriggerSource attribute.

Other than this specific semantic connection, the repeated capability groups of TriggerSource and TriggerAlarm are unreleated.

2.1.5 IviLxiSyncEvent

In the configuration store, the name for the LXI event repeated capability shall be "IviLxiSyncEvent".

2.1.6 Reserved Repeated Capability Identifiers

The following repeated capability identifiers are reserved and must be supported by the specific driver according to the LXI Functional Class (Class A, Class B, or Class C) of the instrument. The specific driver must register each of the reserved repeated capability identifiers as physical repeated capability names in the IVI Configuration Store as per *IVI-3.1: Driver Architecture Specification*.

There are two types of reserved repeated capability identifiers. The first is used for the pre-defined LXI trigger identifiers. The second provides a predefined alarm to simplify programming alarms in the common case when only one is used.

The reserved trigger repeated capability identifiers apply only to the following repeated capabilities:

- IviLxiSyncArmSource
- IviLxiSyncTriggerSource
- IviLxiSyncEvent

The reserved alarm repeated capability identifier is "ALARM0". It applies to the following repeated capabilities:

- IviLxiSyncTriggerAlarm
- IviLxiSyncArmAlarm

Note that the LXI specification reserves event identifiers that begin with the characters "LXI" for LXI use. The strings "LXI0", "LXI1", … ,"LXI7" refer to the 8 LXI wired trigger bus triggers. See LXI 1.1 rule 6.4.5.

Table 2-1. IviLxiSync Reserved Trigger Repeated Capability Identifiers

Repeated Capability Identifier	Required for LXI Functional Class
LXI0	Class A
LXI1	Class A
LXI2	Class A

Table 2-1. IviLxiSync Reserved Trigger Repeated Capability Identifiers

LXI3	Class A
LXI4	Class A
LXI5	Class A
LXI6	Class A
LXI7	Class A
LAN0	Class A, Class B
LAN1	Class A, Class B
LAN2	Class A, Class B
LAN3	Class A, Class B
LAN4	Class A, Class B
LAN5	Class A, Class B
LAN6	Class A, Class B
LAN7	Class A, Class B

2.1.7 Custom Repeated Capability Identifiers

In addition to the reserved repeated capability names defined in Section 2.1, many LXI devices allow custom repeated capability names for arm sources, trigger sources, and events. These are referred to as *custom repeated capability identifiers*.

Custom repeated capability identifiers are used to dynamically add custom arm sources, arm alarms, trigger sources, trigger alarms, and events. Client programs use standard Add and Remove methods (defined later in this specification) to manage the custom sources and events.

In addition to the requirements for valid IVI repeated capability identifiers, a custom repeated capability identifier must meet the following requirement:

The names must contain no more than 16 characters.

Since custom repeated capability identifiers are dynamic, they are not registered in the IVI Configuration Store when the specific driver is installed.

2.1.8 Repeated Capability Identifier Case Sensitivity

Per *IVI-3.1: Driver Architecture Specification,* repeated capability identifiers are case sensitive.

2.1.9 Repeated Capability Implementation Requirements

Drivers are not required to implement extensible repeated capabilities. If a driver does not implement one of the following custom capability groups, it may define a fixed set of repeated capabilities and identifiers:

IviLxiSyncCustomTriggerSource

IviLxiSyncCustomArmSource

IviLxiSyncCustomTriggerAlarm

IviLxiSyncCustomerTriggerArm

IviLxiSyncCustomEvent

Note that for LXI instrument drivers the fixed set of repeated capabilities for IviLxiSyncTriggerSource and IviLxiSyncTriggerArm will generally include the reserved repeated capability identifiers listed in section 2.1.6 as well as others that may be relevant for that instrument.

2.2 IviLxiSync Group Names

The IviLxiSync specification capabilities are divided up into a base capabilities group and multiple extension capabilities groups. This section defines names for each capability group and the functions and attributes that must be implemented by drivers that support the various capability groups.

2.2.1 IviLxiSync Group Names

The capability groups for IviLxiSync are defined in the following table. The Group Name is used to represent a particular capability group. It is returned as one of the possible group names from the drivers group capabilities attribute. See *IVI-3.3: Standard Cross Class Capabilities Specification.*

Table 2-1. IviLxiSync Group Names

Group Name	Attributes	Functions
IviLxiSyncBase	Arm Count Arm Delay Arm Source Count Arm Source Detection Arm Source Enabled Arm Source EventId Arm Source Filter Arm Source Item (IVI-COM and IVI.NET Only) Arm Source Name (IVI-COM and IVI.NET Only) Arm Source Or Enabled Trigger Count Trigger Source Trigger Source Count Trigger Source Delay Trigger Source Detection Trigger Source EventId Trigger Source Item (IVI-COM and IVI.NET Only) Trigger Source Name (IVI-COM and IVI.NET Only) Trigger Source Filter	Configure Arm Source Disable All Arm Sources Get Arm Source Name (IVI-C Only) Configure Trigger Alarm Configure Trigger Source Disable All Trigger Alarms Get Trigger Source Name (IVI-C Only)
IviLxiSyncCustomArmSource		Add Arm Source Remove Arm Source Remove All Custom Arm Sources
IviLxiSyncCustomTriggerSource		Add Trigger Source Remove Trigger Source Remove All Custom Trigger Sources
IviLxiSyncTriggerAlarm	Trigger Alarm Count Trigger Alarm Enabled Trigger Alarm Item (IVI-COM Only)	Configure Trigger Alarm Disable All Trigger Alarms Get Trigger Alarm Name (IVI-C Only)

Table 2-1. IviLxiSync Group Names

	Trigger Alarm Name (IVI-COM Only) Trigger Alarm Period Trigger Alarm Repeat Count Trigger Alarm Time Seconds (IVI-C and IVI-COM Only) Trigger Alarm Time (IVI.NET Only) Trigger Alarm Time Fraction (IVI-C and IVI-COM Only)	
IviLxiSyncArmAlarm	Arm Alarm Count Arm Alarm Enabled Arm Alarm Item (IVI-COM and IVI.NET Only) Arm Alarm Name (IVI-COM and IVI.NET Only) Arm Alarm Period Arm Alarm Repeat Count Arm Alarm Time Seconds (IVI-C and IVI-COM Only) Arm Alarm Time (IVI.NET Only) Arm Alarm Time Fraction (IVI-C and IVI-COM Only)	Configure Arm Alarm Disable All Arm Alarms Get Arm Alarm Name (IVI-C Only)
IviLxiSyncCustomTriggerAlarm		Add Trigger Alarm Remove Trigger Alarm Remove All Trigger Alarms
IviLxiSyncCustomArmAlarm		Add Arm Alarm Remove Arm Alarm Remove All Custom Arm Alarms
IviLxiSyncEvent	Event Count Event Destination Path Event Drive Mode Event Item (IVI-COM and IVI.NET Only) Event Name (IVI-COM and IVI.NET Only) Event Slope Event Source Event Wired OR Bias Mode	Configure Event Disable All Events Get Event Name (IVI-C Only)
IviLxiSyncCustomEvent		Add Event Remove Event Remove All Custom Events
IviLxiSyncEventLog	Event Log Entry Count Event Log Enabled	Clear Event Log Entries Get Next Event Log Entry
IviLxiSyncSyncTime	Is Time Master Is Time Synchronized SystemTime (IVI.NET Only)	Get System Time (IVI-C and IVI-COM Only)

2.3 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`

2.4 .NET Namespace

The .NET namespace for the IviLxiSync class is `Ivi.LxiSync`.

3. IviLxiSyncArm Subsystem

The IviLxiSyncArm subsystem provides a mechanism for using inbound events to control when triggers are accepted by an LXI device. Arming can be controlled via the LXI trigger bus, LXI-defined LAN events, or custom user-defined events. In addition, arming can be controlled via a clocking mechanism. Specifically, the LXI device can arm at a specific 1588 time and repeat at user-specified intervals.

The Arm subsystem also provides a way to logically OR or AND events together. In cases where this is needed, the user can set the trigger to immediate and thereby initiate operations based on an AND or OR of events.

3.1 Behavior Model

The figures below shows the behavior model for the IviLxiSyncArm subsystem.

Figure 2 shows how the IviLxiSync capabilities fit with the conventional IVI Initiate/Abort pattern. Notice that the Arm Clear message in Figure 3 occurs when the Initiate occurs. Similarly, the arm is the number of arms that the instrument accepts before requiring another initiate.

Figure 3 shows the logic to selectively enable any of the inputs as well as selections for edge or level sensitivity and positive or negative slope. The Arm Source Enabled attribute controls whether specific inputs are enabled, while Arm Source Edge and Arm Source Slope control edge or level detection and positive or negative slope, respectively.

Additional logic allows the input signals to be either OR-summed or AND-summed, as shown below. The disposition of input signal summing is controlled by the Arm Source Or Enabled attribute.

The delay block shown in the diagram is used to compensate for electrical delay differences between the arm signal path and the signal path through the device under test. Some devices, such as narrow-band crystal filters, have significant delay that must be taken into account in order to make valid measurements.

Arm capabilities should be implemented if the underlying instrument supports arming. If the driver and instrument do not implement Arm capabilities, the existence of the functions in the class should not impact an application that only uses trigger.

Drivers that implement Arm should be designed so that users that do not use Arm capabilities do not need to issue any specific invocations of the driver to make the Arm system not intrude on a triggered measurement. That is, if the application does not have a need for Arm, the default behaviors of the Arm system should configure the instrument so that it works with no required Arm preconditioning or additional programming.

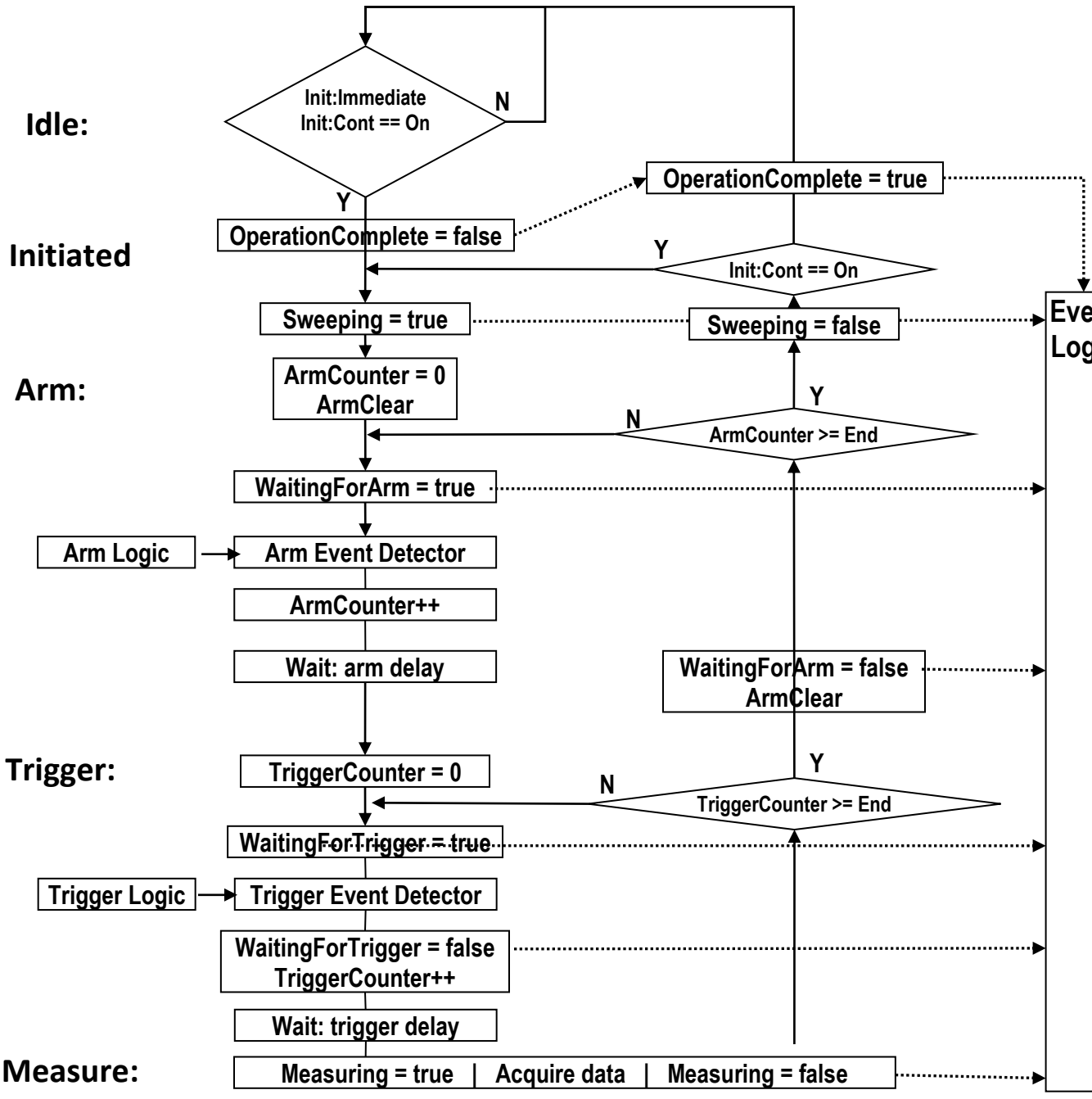


Figure 2. Overall Trigger Behavior Model

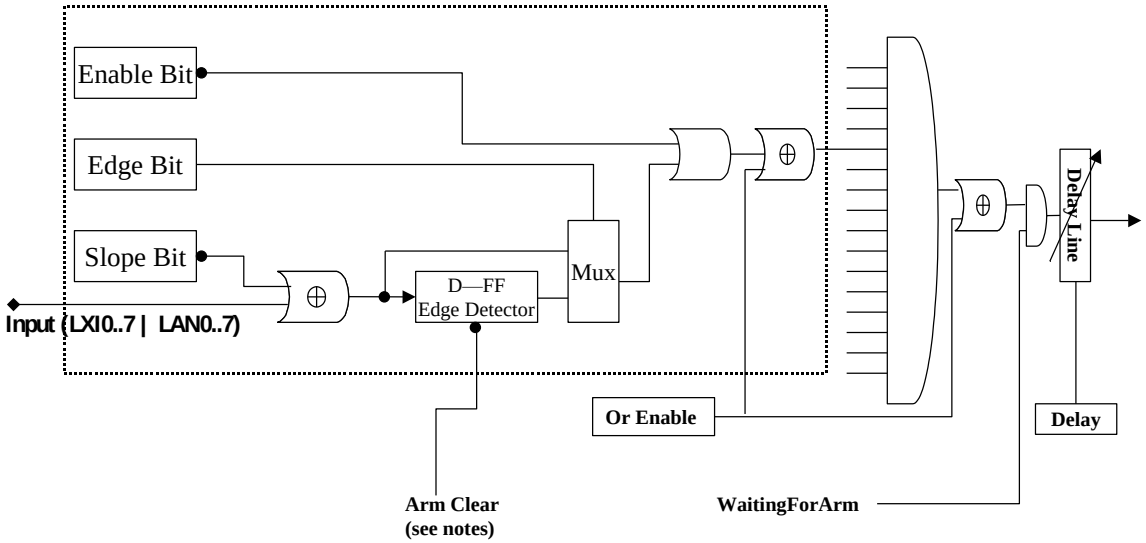


Figure 3. IviLxiSyncArm Behavior Model

3.2 IviLxiSyncArm Attributes

The IviLxiSyncArm subsystem defines the following attributes:

- Arm Count
- Arm Alarm Count
- Arm Alarm Enabled
- Arm Alarm Item (IVI-COM and IVI.NET Only)
- Arm Alarm Name (IVI-COM and IVI.NET Only)
- Arm Alarm Period
- Arm Alarm Repeat Count
- Arm Alarm Time (IVI.NET Only)
- Arm Alarm Time Fraction (IVI-C and IVI-COM Only)
- Arm Alarm Time Seconds (IVI-C and IVI-COM Only)
- Arm Delay
- Arm Source Count
- Arm Source Detection
- Arm Source Enabled
- Arm Source Event ID
- Arm Source Filter
- Arm Source Item (IVI-COM and IVI.NET Only)
- Arm Source Name (IVI-COM and IVI.NET Only)
- Arm Source Or Enabled

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

3.2.1 Arm Count

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

.NET Property Name

Arm.ArmCount

COM Property Name

Arm.ArmCount

C Constant Name

IVILXISYNC_ATTR_ARM_COUNT

Description

Specifies the number of times the arm has to occur to complete the arm loop; that is, the number of arms that are accepted before the measurement must be initiated again.

.NET 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.

3.2.2 Arm Alarm Count

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

.NET Property Name

```
Arm.Alarms.Count
```

This property is inherited from `IIviRepeatedCapabilityCollection.`

COM Property Name

```
Arm.Alarms.Count
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_ALARM_COUNT
```

Description

Returns the number of currently available arm alarms. The count returned includes the reserved repeated capability named “ALARM0” as defined Section 2.1.6, *Reserved Repeated Capability Identifiers* as well as any custom 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.

3.2.3 Arm Alarm Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

```
Arm.Alarms[].Enabled
```

COM Property Name

```
Arm.Alarms.Item().Enabled
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_ALARM_ENABLED
```

Description

If set to True, the LXI device enables the arm alarm. If set to False, the LXI device disables the arm alarm.

Compliance Notes

The specific driver shall implement both the True and False values.

Return Values

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

Completion Codes	Description
Alarm Time Invalid	Error: Alarm time is inconsistent with the instrument hardware. For instance, the specified time (as indicated by the Arm Alarm Time Seconds and Arm Alarm Time Fractional attributes) has already passed.

.NET 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.

3.2.4 Arm Alarm Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviLxiSyncArmAlarm*	RO	IviLxiSyncArmAlarm	None	N/A

.NET Property Name

```
Arm.Alarms[String name]
```

This indexer is inherited from the base interface `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular arm alarm in the arm alarms collection.

COM Property Name

```
Arm.Alarms.Item ([in] BSTR SourceName)
```

C Constant Name

```
N/A
```

Description

Arm Alarm Item uniquely identifies an arm alarm in the arm alarms collection. It returns an interface pointer which can be used to control the attributes and other functionality of that arm alarm.

The Item property takes an arm alarm name. If the user passes an invalid value for the source name parameter, the property returns an error.

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

Return Values

If the IVI-COM driver cannot recognize the `Name` parameter, it returns an Unknown Name in Selector error as described in *IVI-3.2: Inherent Capabilities Specification*, Section 9.3.

.NET 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.

3.2.5 Arm Alarm Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

```
Arm.Alarms[].Name
```

This property is inherited from `IIviRepeatedCapabilityIdentification`.

COM Property Name

```
Arm.Alarms.Name ([in] LONG SourceIndex)
```

C Constant Name

N/A.

(Use the `GetArmAlarmName` function.)

Description

Returns the physical repeated capability identifier defined by the specific driver for the arm alarm that corresponds to the one-based index that the user specifies. For custom arm sources added with the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

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

.NET Exceptions

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

3.2.6 Arm Alarm Period

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm
Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

```
Arm.Alarms[].Period
```

COM Property Name

```
Arm.Alarms.Item().Period
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_ALARM_PERIOD
```

Description

Specifies the period of the arm alarm; that is, the amount of time that transpires before the alarm repeats.

For C and COM, time is in seconds, and a period of zero means there is no repeat and a single arm is generated.

For .NET, the units are implicit in the definition of PrecisionTimeSpan. A period of PrecisionTimeSpan.Zero means there is no repeat and a single arm 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 this property.

3.2.7 Arm Alarm Repeat Count

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

```
Arm.Alarms[].RepeatCount
```

COM Property Name

```
Arm.Alarms.Item().RepeatCount
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_ALARM_REPEAT_COUNT
```

Description

Specifies the number of times the trigger will occur at the period specified by the Arm Alarm Period attribute. If Arm Alarm Repeat Count is zero, then the alarm shall repeat forever at the Arm Alarm Period.

For IVI-C, the defined value of IVILXISYNC_VAL_REPEAT_CONTINUOUS is provided to set the repeat count to forever.

.NET 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.

3.2.8 Arm Alarm Time (IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
`Ivi.Driver.PrecisionDateTime`	`R/W`	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

`Arm.Alarms[].Time`

COM Property Name

`N/A`

(Use the Arm Alarm Time Seconds and Arm Alarm Time Fraction attributes.)

C Constant Name

`N/A`

(Use the Arm Alarm Time Seconds and Arm Alarm Time Fraction attributes.)

Description

Specifies the time at which the alarm will go off. Note that PrecisionDateTime includes both seconds and fractional seconds, so that Arm Alarm Time in the .NET API replaces both Arm Alarm Time Seconds and Arm Alarm Time Fraction in the C/COM API.

Once the alarm goes off, it will repeat at the period set by Arm Alarm Period the number of times set by Arm Alarm Count.

.NET 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.

3.2.9 Arm Alarm Time Seconds (IVI-C and IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

N/A

(Use the Arm Alarm Time attribute.)

COM Property Name

Arm.Alarms.Item().TimeSeconds

C Constant Name

IVILXISYNC_ATTR_ARM_ALARM_TIME_SECONDS

Description

Specifies the seconds portion of time at which the alarm will go off. Note that the actual time of the alarm is the sum of Arm Alarm Time Seconds and Arm Alarm Time Fraction. The time is specified as the sum of two values because a single double-precision floating-point number does not have sufficient range and resolution to specify the time.

Once the alarm goes off, it will repeat at the period set by Arm Alarm Period the number of times set by Arm Alarm Count.

3.2.10 Arm Alarm Time Fraction (IVI-C and IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	IviLxiSyncArmAlarm	None	Configure Arm Alarm

.NET Property Name

N/A

(Use the Arm Alarm Time attribute.)

COM Property Name

Arm.Alarms.Item().TimeFraction

C Constant Name

IVILXISYNC_ATTR_ARM_ALARM_TIME_FRACTION

Description

Specifies the fractional portion of the time at which the alarm will go off. Note that the actual time of the alarm is the sum of Arm Alarm Time Second and Arm Alarm Time Fraction. The time is specified as the sum of two values because a single double-precision floating-point value does not have sufficient range and resolution to specify the time.

Once the alarm goes off, it will repeat at the period set by Arm Alarm Period the number of times set by Arm Alarm Count.

3.2.11 Arm Delay

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	N/A
Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	N/A	None	N/A

.NET Property Name

Arm.Delay

COM Property Name

Arm.Delay

C Constant Name

IVILXISYNC_ATTR_ARM_DELAY

Description

Specifies the delay from when the arm logic satisfied until the waiting for trigger state is entered. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

.NET Exceptions

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

3.2.12 Arm Source Count

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

.NET Property Name

```
Arm.Sources.Count
```

This property is inherited from `IIviRepeatedCapabilityCollection.`

COM Property Name

```
Arm.Sources.Count
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_COUNT
```

Description

Returns the number of currently available arm sources. The count returned includes any of the supported reserved repeated capability names defined in Section 2.1.6, *Reserved Repeated Capability Identifiers* as well as any custom 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.

3.2.13 Arm Source Detection

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	IviLxiSyncArmSource	None	Configure Arm Source

.NET Property Name

```
Arm.Sources[].Detection
```

.NET Enumeration Name

```
Ivi.LxiSynchronization.ArmSourceDetection
```

COM Property Name

```
Arm.Sources.Item().Detection
```

COM Enumeration Name

```
IviLxiSyncSourceDetectionEnum
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_DETECTION
```

Description

Specifies the style of arm source detection.

If the source is a LAN event and the source detection is set to rise, this Arm repeated capability will be satisfied when the designated LAN packet arrives with a True indication. If the source detection is set to fall, this Arm repeated capability will be satisfied when a LAN packet arrives with a False indication. If the detection is set to high, the source will be satisfied when the designated LAN packet arrives with a True indication and remain satisfied until the designated LAN packet arrives with a False indication. If the detection is to low, the source will be satisfied when the designated LAN packet arrives with a False indication and remain satisfied until the designated LAN packet arrives with a True indication.

.NET Exceptions

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

Defined Values

Name	*Description*		
		Language	*Identifier*
Rise	Configures the LXI device to arm on the rising edge of the arm source.		
		.NET	ArmSourceDetection.Rise
		C	IVILXISYNC_VAL_DETECTION_RISE
		COM	IviLxiSyncArmSourceDetectionRise
Fall	Configures the LXI device to arm on the falling edge of the arm source.		
		.NET	ArmSourceDetection.Fall
		C	IVILXISYNC_VAL_DETECTION_FALL
		COM	IviLxiSyncArmSourceDetectionFall
High	Configures the LXI device to arm while the arm source is high, that is, while it remains true		
		.NET	ArmSourceDetection.High
		C	IVILXISYNC_VAL_DETECTION_HIGH
		COM	IviLxiSyncArmSourceDetectionHigh
Low	Configures the LXI device to arm while the arm source is low, that is, while it remains false		
		.NET	ArmSourceDetection.Low
		C	IVILXISYNC_VAL_DETECTION_LOW
		COM	IviLxiSyncArmSourceDetectionLow

Compliance Notes

The specific driver shall implement all values.

3.2.14 Arm Source Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	IviLxiSyncArmSource	None	Configure Arm Source

.NET Property Name

```
Arm.Sources[].Enabled
```

COM Property Name

```
Arm.Sources.Item().Enabled
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_ENABLED
```

Description

If set to True, the LXI device enables the arm source.. If set to False, the LXI device disables the arm source. If a source is disabled, it has no affect on the summary arm signal.

.NET Exceptions

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

Compliance Notes

The specific driver shall implement both the True and False values.

3.2.15 Arm Source EventId

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

.NET Property Name

```
Arm.Sources[].EventId
```

COM Property Name

```
Arm.Sources.Item().EventId
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_EVENTID
```

Description

This specifies the LAN event identifier that is associated with this arm source. LAN Events with this identifier are accepted from the source described in the filter.

The default value for EventId is the repeated capability specifier for this arm source.

.NET 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.

3.2.16 Arm Source Filter

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

.NET Property Name

```
Arm.Sources[].Filter
```

COM Property Name

```
Arm.Sources.Item().Filter
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_FILTER
```

Description

Specifies a filter for restricting arm sources. The filter specified by this attribute denotes the *accepted* sources. The syntax for specifying a filter is as follows:

<Filter> == [(<tcp> | <udp> | <any>) [, <Filter>]]

<tcp> == <host> [:<port>]

<udp> == ALL [: <port>]

<any> == : <port>

<host> is either a hostname or host number. Note that the hostname can not be "ALL" since that would indicate the <udp> construct.

<port> is a series of decimal digits indicating the port number.

Specifying an empty string or VI_NULL means that LXI arm packets are accepted via either TCP or UDP multicast from any host. Note that ":5044" is equivalent to the empty string since 5044 is the IANA registered port for LXI events (lxi-evntsvc).

In the <tcp>, <udp> and <any> constructs, <port> refers to the port the device receives the LAN message on. If <port> is omitted from <tcp> or <udp>, packets are only accepted on the IANA registered port for LXI events (lxi-evntsvc).

Specifying the <host> (<tcp> construct) indicates that packets via TCP on the port indicated are accepted.

Specifying ALL (<udp> construct) indicates that UDP multicast packets are accepted if they are directed to the IANA registered port for LXI events (lxi-evntsvc) on the IANA registered multicast address (LXI-EVENT). No TCP packets are accepted unless a <tcp> syntax is also included in the filter. The multicast address can not be altered with this syntax.

Specifying any protocol (<any> construct) indicates that both packets via TCP and UDP multicast packets are accepted if they are directed to the specified port. UDP multicast packets must be received at the IANA registered multicast addres (LXI-EVENT).

The send port is not monitored. This allows the transmitter to use any available port.

Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible filters syntaxes.

White space shall be ignored. The <Filter> string is case insensitive.

Conventional devices should consider restricting the <port> to only the IANA registered port for LXI events (lxi-evntsvc) and not accepting the generalized syntax.

Example Filter	Description
"192.168.0.1:23"	Accepts arm source packets via TCP on port 23 from the host at IP address 192.168.0.1.
"A_SIGGEN1:23,A_SPECAN2:23"	Accepts arm source packets via TCP on port 23 from the host with DNS name A_SIGGEN1 as well as packets via TCP on port 23 on the host with DNS name A_SPECAN2.
"192.168.0.1"	Accepts arm source packets from via TCP on the IANA registered port for LXI events (lxi-eventsvc) from the host at IP address 192.168.0.1.
"All:23,A_SPECAN2"	Accepts arm source packets via UDP multicast on the IANA assigned multicast address to port 23 from any host as well as packets via TCP on the IANA registered port for LXI events (lxi-eventsvc) from the host with DNS name A_SPECAN2.
""	Accepts arm source packets via TCP or UDP multicast on the IANA assigned multicast address on the IANA registered port for LXI events (lxi-eventsvc) from any host. This is equivalent to <any> with the IANA registered port for LXI events (lxi-evntsvc).
"All"	Accepts arm source packets via UDP multicast on the IANA assigned multicast address on the IANA registered port for LXI events (lxi-eventsvc) from any host. This is equivalent to "".
"All:8543"	Accepts arm source packets arriving via UDP multicast on the IANA UDP multicast address on port 8543 from any host.
"All, 192.168.1.1"	Accepts arm source packets arriving via UDP multicast on the IANA assigned multicast address at the IANA registered port for LXI events (lxi-eventsvc), or TCP packets arriving on the IANA registered port for LXI events (lxi-eventsvc) from IP address 192.168.1.1
":23"	Accepts arm source packets arriving via either UDP multicast or TCP on port 23 from any host.

.NET 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.

3.2.17 Arm Source Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviLxiSyncArmSource*	RO	IviLxiSyncArmSource	None	N/A

.NET Property Name

```
Arm.Sources[String name]
```

This indexer is inherited from `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular arm source in the arm sources collection.

COM Property Name

```
Arm.Sources.Item ([in] BSTR SourceName)
```

C Constant Name

```
N/A
```

Description

Arm Source Item uniquely identifies an arm source in the arm sources collection. It returns an interface pointer which can be used to control the attributes and other functionality of that arm source.

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

The .NET indexer may take an arm source name or an index into the arm source collection. For .NET, valid values for the indexer's `index` parameter are between zero and the value of the Arm Source Count attribute, minus 1, inclusive. If the user passes an invalid value for the `name` or `index` parameter, the indexer 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.

3.2.18 Arm Source Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

```
Arm.Sources[].Name
```

This property is inherited from `IIviRepeatedCapabilityIdentification`.

COM Property Name

```
Arm.Sources.Name ([in] LONG SourceIndex)
```

C Constant Name

N/A.

(Use the Get Arm Source Name function.)

Description

Returns the physical repeated capability identifier defined by the specific driver for the arm source that corresponds to the index that the user specifies. For custom arm sources added with the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

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

.NET Exceptions

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

3.2.19 Arm Source Or Enabled

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

.NET Property Name

Arm.Sources.OrEnabled

COM Property Name

Arm.Sources.OrEnabled

C Constant Name

IVILXISYNC_ATTR_ARM_SOURCE_OR_ENABLED

Description

If set to True, the arm sources use OR-summing of the arm sources. If set to False, the arm sources use AND-summing of the arm sources.

Compliance Notes

The specific driver shall implement both the True and False values.

.NET 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.

3.3 IviLxiSyncArm Functions

The IviLxiSyncArm subsystem defines the following functions:

- Add Arm Alarm
- Add Arm Source
- Configure Arm Alarm
- Configure Arm Source
- Disable All Arm Alarms
- Disable All Arm Sources
- Get Arm Alarm Name (IVI-C Only)
- Get Arm Source Name (IVI-C Only)
- Remove Arm Alarm
- Remove Arm Source
- Remove All Custom Arm Alarms
- Remove All Custom Arm Sources

This section describes the behavior and requirements of each function.

3.3.1 Add Arm Alarm

Description

This function creates a new arm alarm.

When a new alarm is added, the default values for the associated attributes are as follows:

Attribute	Default Value
Arm Alarm Enabled	False
Arm Alarm Period	0
Arm Alarm Repeat Count	1
Arm Alarm Time (.NET)	00:00:00 Jan 1, 1970
Arm Alarm Time Seconds (C/COM)	0
Arm Alarm Time Fraction (C/COM)	0

.NET Method Prototype

```
IIviLxiSyncArmAlarm Arm.Alarms.Add(String alarmName);
```

COM Method Prototype

```
HRESULT Arm.Alarms.Add([in] BSTR AlarmName);
```

C Prototype

```
ViStatus IviLxiSync_AddArmAlarm (ViSession Vi,
                                 ViConstString AlarmName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
AlarmName	Specifies the name of the arm alarm to create.	ViConstString

Outputs	Description	Base Type
Return value (.NET)	A reference to the arm alarm object that was added.	IIviLxiSyncArmAlarm

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 status codes for this function.

Completion Codes	Description
Alarm Exists	Error: Alarm already exists.

.NET 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
AlarmExistsException	Alarm already exists.

3.3.2 Add Arm Source

Description

This function creates a new arm source.

When a new source is added, the default values for the associated attributes are as follows:

Attribute	Default Value
`Arm Source Edge`	True (edge-detection)
`Arm Source Enabled`	True
`Arm Source EventId`	The repeated capability source name specified in this call
`Arm Source Slope`	Positive
`Arm Source Filter`	“” (Empty string)
`Arm Source Or Enabled`	False

.NET Method Prototype

```
IIviLxiSyncArmSource Arm.Sources.Add(String sourceName);
```

COM Method Prototype

```
HRESULT Arm.Sources.Add([in] BSTR SourceName);
```

C Prototype

```
ViStatus IviLxiSync_AddArmSource (ViSession Vi,
                                  ViConstString SourceName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`SourceName`	Specifies the name of the arm source to create.	`ViConstString`

Outputs	Description	Base Type
`Return value (.NET)`	A reference to the arm source object that was added .	`IIviLxiSyncArmAlarm`

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 status codes for this function.

Completion Codes	Description
Event Source Exists	Error: Event source already exists.
Out of Event Resources	Error: Out of event resources.

.NET 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
EventSourceExistsException	Event source already exists.
OutOfEventResourcesException	Out of event resources.

3.3.3 Configure Arm Alarm

Description

This function configures the most commonly used attributes of the arm alarm subsystem.

.NET Method Prototype

```
void Arm.Alarms[].Configure(Boolean enabled,
                            PrecisionDateTime time,
                            PrecisionTimeSpan period,
                            Int32 repeatCount);
```

COM Method Prototype

```
HRESULT Arm.Alarms.Item().Configure([in] VARIANT_BOOL Enabled,
                                    [in] DOUBLE TimeSeconds,
                                    [in] DOUBLE TimeFraction,
                                    [in] DOUBLE Period,
                                    [in] LONG RepeatCount);
```

C Prototype

```
ViStatus IviLxiSync_ConfigureArmAlarm (ViSession Vi,
                                       ViConstString AlarmName,
                                       ViBoolean Enabled,
                                       ViReal64 TimeSeconds,
                                       ViReal64 TimeFraction,
                                       ViReal64 Period,
                                       ViInt32 RepeatCount);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
AlarmName	The name of the alarm.	ViConstString
Enabled	Enables or disables the arm alarm. The driver uses this value to set the Arm Alarm Enabled attribute. See the attribute description for more information.	ViBoolean
time (.NET)	PrecisionDateTime. The driver uses this value to set the Arm Alarm Time attribute. See the attribute description for more information.	Ivi.Driver. PrecisionDateTime
TimeSeconds (C/COM)	Specifies the seconds part of the time. The driver uses this value to set the Arm Alarm Time Seconds attribute. See the attribute description for more information.	ViReal64
TimeFraction (C/COM)	Specifies the fractional part of the time. The driver uses this value to set the Arm Alarm Time Fractional attribute. See the attribute description for more information.	ViReal64
Period	Specifies the period of the arm alarm. The driver uses this value to set the Arm Alarm Period attribute. See the attribute description for more information.	ViReal64 (C/COM) Ivi.Driver. PrecisionTimeSpan (.NET)

RepeatCount	Specifies the number of times to repeat the trigger at the period specified by the Arm Alarm Repeat Period attribute. The driver uses this value to set the Arm Alarm Repeat Count attribute. See the attribute description for more information.	ViInt32

Return Values (C/COM)

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

.NET Exceptions

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

3.3.4 Configure Arm Source

Description

This function configures the most commonly used attributes of the arm source subsystem.

.NET Method Prototype

```
void Arm.Sources[].Configure(Boolean enabled,
                             ArmSourceDetection detection);
```

COM Method Prototype

```
HRESULT Arm.Sources.Item().Configure([in] VARIANT_BOOL Enabled,
                                     [in] IviLxiSyncArmSourceDetectionEnum
Detection);
```

C Prototype

```
ViStatus IviLxiSync_ConfigureArmSource (ViSession Vi,
                                        ViConstString SourceName,
                                        ViBoolean Enabled,
                                        ViInt32 Detection);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
SourceName	Name of the arm source to configure.	ViConstString
Enabled	Enables or disables the arm source. The driver uses this value to set the Arm Source Enabled attribute. See the attribute description for more information.	ViBoolean
Detection	Specifies the detection mode of the arm source. The driver uses this value to set the Arm Source Detection attribute. See the attribute description for more information.	ViInt32

Return Values (C/COM)

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

.NET Exceptions

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

3.3.5 Disable All Arm Alarms

Description

This function disables all arm alarms. The specific driver uses this function to set the Arm Alarm Enabled property to False for all arm alarms.

.NET Method Prototype

```
void Arm.Alarms.DisableAll();
```

COM Method Prototype

```
HRESULT Arm.Alarms.DisableAll();
```

C Prototype

```
ViStatus IviLxiSync_DisableAllArmAlarms (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.

3.3.6 Disable All Arm Sources

Description

This function disables all arm sources. The specific driver uses this function to set the Arm Source Enabled property to False for all arm sources.

.NET Method Prototype

```
void Arm.Sources.DisableAll();
```

COM Method Prototype

```
HRESULT Arm.Sources.DisableAll();
```

C Prototype

```
ViStatus IviLxiSync_DisableAllArmSources (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.

3.3.7 Get Arm Alarm Name (IVI-C Only)

Description

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the `AlarmIndex` parameter is less than one or greater than the value of the Arm Alarm Count attribute, the function returns an empty string in the `AlarmName` parameter and returns an error. For custom arm sources added vith the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

.NET Method Prototype

`N/A`

(Use the `Arm.Alarms[].Name` property)

COM Method Prototype

`N/A`

(Use the `Arm.Alarms.Item().Name` property.)

C Prototype

```
ViStatus IviLxiSync_GetArmAlarmName (ViSession Vi,
                                     ViInt32 AlarmIndex,
                                     ViInt32 AlarmNameBufferSize,
                                     ViChar AlarmName[]);
```

Parameters

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

Outputs	Description	Data Type
`AlarmName`	The buffer into which the function returns the alarm name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the AlarmNameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

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

3.3.8 Get Arm Source Name (IVI-C Only)

Description

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the `SourceIndex` parameter is less than one or greater than the value of the Arm Source Count attribute, the function returns an empty string in the `SourceName` parameter and returns an error. For custom arm sources added vith the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

.NET Method Prototype

`N/A`

(Use the `Arm.Sources[].Name` property)

COM Method Prototype

`N/A`

(Use the `Arm.Sources.Item().Name` property.)

C Prototype

```
ViStatus IviLxiSync_GetArmSourceName (ViSession Vi,
                                      ViInt32 SourceIndex,
                                      ViInt32 SourceNameBufferSize,
                                      ViChar SourceName[]);
```

Parameters

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

Outputs	Description	Data Type
`SourceName`	The buffer into which the function returns the source name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the SourceNameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

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

3.3.9 Remove Arm Alarm

Description

This function removes an arm alarm. Any resources associated with this alarm should be freed.

The AlarmName parameter is case-insensitive.

.NET Method Prototype

```
void Arm.Alarms.Remove(String alarmName);
```

COM Method Prototype

```
HRESULT Arm.Alarms.Remove([in] BSTR AlarmName);
```

C Prototype

```
ViStatus IviLxiSync_RemoveArmAlarm (ViSession Vi,
                                    ViConstString AlarmName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`AlarmName`	Specifies the name of the arm alarm to remove.	`ViConstString`

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 status codes for this function.

Completion Codes	Description
Alarm Does Not Exist	Error: Alarm does not exist.
Attempt To Remove Reserved Repeated Capability	Error: The repeated capability name is reserved and cannot be removed.

.NET 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
AlarmDoesNotExistException	Alarm does not exist.
CannotRemoveReservedRepeatedCapabilityException	The repeated capability name is reserved and cannot be removed.

3.3.10 Remove Arm Source

Description

This function removes an arm source. Any resources associated with this alarm should be freed.

The SourceName parameter is case-insensitive.

.NET Method Prototype

```
void Arm.Sources.Remove(String sourceName);
```

COM Method Prototype

```
HRESULT Arm.Sources.Remove([in] BSTR SourceName);
```

C Prototype

```
ViStatus IviLxiSync_RemoveArmSource (ViSession Vi,
                                     ViConstString SourceName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
SourceName	Specifies the name of the arm source to remove.	ViConstString

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 status codes for this function.

Completion Codes	Description
Event Source Does Not Exist	Error: Event source does not exist.
Attempt To Remove Reserved Repeated Capability	Error: The repeated capability name is reserved and cannot be removed.

.NET 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
EventSourceDoesNotExistException	Event source does not exist.
CannotRemoveReservedRepeatedCapabilityException	The repeated capability name is reserved and cannot be removed.

3.3.11 Remove All Custom Arm Alarms

Description

This function removes all of the arm alarms that were added using the Add Arm Alarm function.

.NET Method Prototype

```
void Arm.Alarms.RemoveAllCustomArmAlarms();
```

COM Method Prototype

```
HRESULT Arm.Alarms.RemoveAllCustomArmAlarms();
```

C Prototype

```
ViStatus IviLxiSync_RemoveAllCustomArmAlarms (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.

3.3.12 Remove All Custom Arm Sources

Description

This function removes all of the custom arm sources that were added using the Add Arm Source function. The arm sources associated with the reserved repeated capability identifiers, as defined in Section 2.1.6, *Reserved Repeated Capability Identifiers*, are not affected by this function.

.NET Method Prototype

```
void Arm.Sources.RemoveAllCustomArmSources();
```

COM Method Prototype

```
HRESULT Arm.Sources.RemoveAllCustomArmSources();
```

C Prototype

```
ViStatus IviLxiSync_RemoveAllCustomArmSources (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. IviLxiSyncTrigger Subsystem

The IIviLxiSyncTrigger interface provides a mechanism for using inbound events to control when to trigger an LXI device. This subsystem provides control for triggers from the LXI trigger bus, LXI-defined LAN events, or custom user-defined events. Methods and properties for controlling this mode of triggering are exposed via the IIviLxiSyncTriggerSources sub-interface. In addition, triggering can be controlled via a clocking mechanism. Specifically, the LXI device can trigger at a specific time or at user-specified intervals.

4.1 Behavior Model

The IviLxiSyncTrigger subsystem behavior model is somewhat simpler than that of the IviLxiSyncArm subsystem. This is due to the fact that only a single trigger source is selected at a time and that triggers are always initiated on an edge (that is, when the trigger occurs, the operation begins immediately, so a "level trigger" is not a meaningful concept).

If an application needs to OR-sum or AND-sum events and use the result to trigger a measurement (or other operation). This can be done by using the summing feature of the IviLxiSyncArm subsystem and selecting an Immediate trigger in the trigger logic. A similar technique can be used for trigger gating (controlling the acquisition process with the level of a signal). To do this set the Arm Source Edge attribute to False and select the Immediate trigger in the trigger logic.

The Trigger subsystem should be designed so that users that do not use these trigger capabilities do not need to issue any specific invocations of the driver to prevent the trigger system from intruding on a measurement that is made with Initiate or Read. That is, if the application does not have a need for Trigger, the default behaviors of the Trigger system should configure the instrument so that it works with no required Trigger conditions beyond the Initiate.

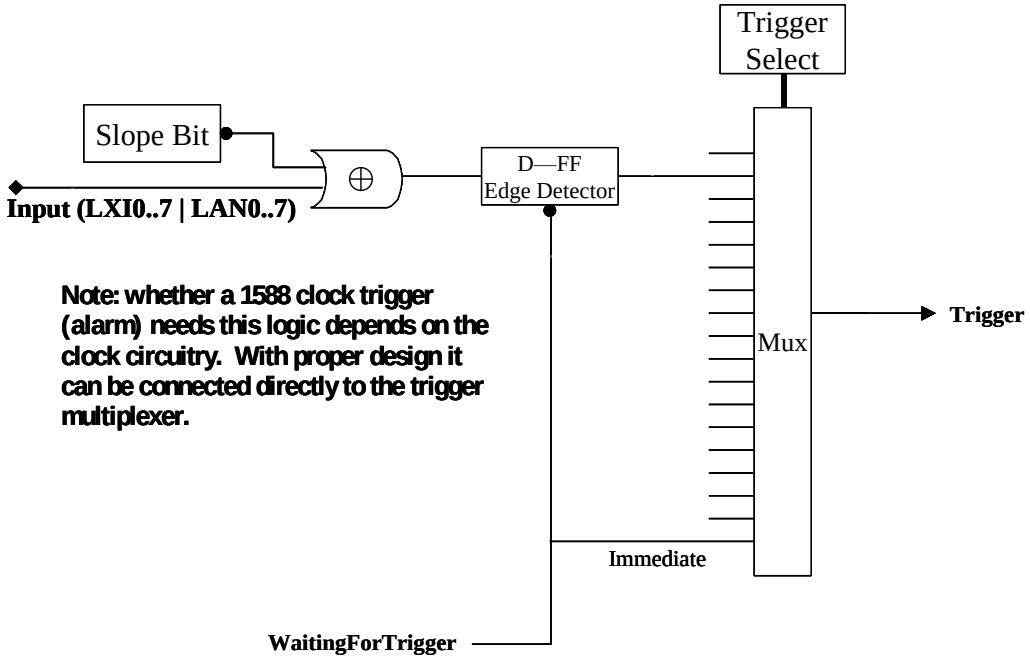


Figure 4. IviLxiSyncTrigger Behavior Model

4.2 IviLxiSyncTrigger Attributes

The IviLxiSyncTrigger subsystem defines the following attributes:

- Trigger Alarm Count
- Trigger Alarm Enabled
- Trigger Alarm Item (IVI-COM and IVI.NET Only)
- Trigger Alarm Name (IVI-COM and IVI.NET Only)
- Trigger Alarm Period
- Trigger Alarm Repeat Count
- Trigger Alarm Time (IVI.NET Only)
- Trigger Alarm Time Fraction (IVI-C and IVI-COM Only)
- Trigger Alarm Time Seconds (IVI-C and IVI-COM Only)
- Trigger Count
- Trigger Source
- Trigger Source Count
- Trigger Source Delay
- Trigger Source Detection
- Trigger Source Event ID
- Trigger Source Item (IVI-COM and IVI.NET Only)
- Trigger Source Name (IVI-COM and IVI.NET Only)
- Trigger Source Filter

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

4.2.1 Trigger Alarm Count

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

.NET Property Name

```
Trigger.Alarms.Count
```

COM Property Name

```
Trigger.Alarms.Count
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_ALARM_COUNT
```

Description

Returns the number of currently available trigger alarms. The count returned includes the reserved repeated capability named "ALARM0" as defined Section 2.1.6, *Reserved Repeated Capability Identifiers* as well as any custom repeated capability identifiers.

.NET Exceptions

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

4.2.2 Trigger Alarm Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	IviLxiSyncTriggerAlarm	None	Configure Trigger Alarm

.NET Property Name

```
Trigger.Alarms[].Enabled
```

COM Property Name

```
Trigger.Alarms.Item().Enabled
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_ALARM_ENABLED
```

Description

If set to True, the LXI device enables the trigger alarm. If set to False, the LXI device disables the trigger alarm.

Compliance Notes

The specific driver shall implement both the True and False values.

Return Values

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

Completion Codes	Description
Alarm Time Invalid	Error: Alarm time is inconsistent with the instrument hardware. For instance, the specified time (as indicated by the Trigger Alarm Time Seconds and Trigger Alarm Time Fractional attributes) has already passed.

.NET Exceptions

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

4.2.3 Trigger Alarm Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviLxiSyncTriggerAlarm*	RO	IviLxiSyncArmAlarm	None	N/A

.NET Property Name

```
Trigger.Alarms[String name]
```

This indexer is inherited from `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular trigger alarm in the Trigger Alarms collection.

COM Property Name

```
Trigger.Alarms.Item ([in] BSTR SourceName)
```

C Constant Name

```
N/A
```

Description

Trigger Alarm Item uniquely identifies a trigger alarm in the trigger alarms collection. It returns an interface pointer which can be used to control the attributes and other functionality of that trigger alarm.

The Item property takes a trigger alarm name. If the user passes an invalid value for the `SourceName` parameter, the property returns an error.

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

Return Values

If the IVI-COM driver cannot recognize the `Name` parameter, it returns an Unknown Name in Selector completion code as described in *IVI-3.2: Inherent Capabilities Specification*, Section 9.3.

.NET Exceptions

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

4.2.4 Trigger Alarm Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

```
Trigger.Alarms[].Name
```

This property is inherited from `IIviRepeatedCapabilityIdentification`.

COM Property Name

```
Trigger.Alarms.Name ([in] LONG SourceIndex)
```

C Constant Name

```
N/A
```

(Use the `GetTriggerAlarmName` function.)

Description

Returns the physical repeated capability identifier defined by the specific driver for the trigger alarm that corresponds to the index that the user specifies. For custom arm sources added with the Add Trigger Source function, this function returns the arm source name in the original casing used when Add Trigger Source was called.

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

.NET Exceptions

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

4.2.5 Trigger Alarm Period

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Trigger Alarm
Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Trigger Alarm

.NET Property Name

```
Trigger.Alarms[].Period
```

COM Property Name

```
Trigger.Alarms.Item().Period
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_ALARM_PERIOD
```

Description

Specifies the period of the trigger alarm; that is, the amount of time that transpires before the alarm repeats.

For C and COM, time is in seconds, and a period of zero means there is no repeat and a single trigger is generated.

For .NET, the units are implicit in the definition of PrecisionTimeSpan. A period of PrecisionTimeSpan.Zero means there is no repeat and a single arm 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 this property.

4.2.6 Trigger Alarm Repeat Count

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

.NET Property Name

```
Trigger.Alarms[].RepeatCount
```

COM Property Name

```
Trigger.Alarms.Item().RepeatCount
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_ALARM_REPEAT_COUNT
```

Description

Specifies the number of times the trigger will occur at the period specified by the Trigger Alarm Period attribute. If Trigger Alarm Repeat Period is non-zero and Trigger Alarm Repeat Count is zero, then the alarm shall repeat forever at the Trigger Alarm Period.

For IVI-C, the defined value of IVILXISYNC_VAL_REPEAT_CONTINUOUS is provided to set the repeat count to forever.

4.2.7 Trigger Alarm Time (IVI.NET Only)

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

.NET Property Name

Trigger.Alarms[].Time

COM Property Name

N/A

(Use the Trigger Alarm Time Seconds and Trigger Alarm Time Fraction attributes.)

C Constant Name

N/A

(Use the Trigger Alarm Time Seconds and Trigger Alarm Time Fraction attributes.)

Description

Specifies the time at which the alarm will go off. Note that PrecisionDateTime includes both seconds and fractional seconds, so that Arm Alarm Time in the .NET API replaces both Trigger Alarm Time Seconds and Trigger Alarm Time Fraction in the C/COM API.

Once the alarm goes off, it will repeat at the period set by Trigger Alarm Period the number of times set by Trigger Alarm Count.

.NET Exceptions

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

4.2.8 Trigger Alarm Time Seconds (IVI-C and IVI-COM Only)

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

.NET Property Name

N/A

(Use the Trigger Alarm Time attribute.)

COM Property Name

Trigger.Alarms.Item().TimeSeconds

C Constant Name

IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_SECONDS

Description

Specifies the seconds portion of the time at which the alarm will go off. Note that the actual time of the alarm is the sum of Trigger Alarm Time Seconds and Trigger Alarm Time Fraction. The time is specified as the sum of two values because a single double-precision floating-point does not have sufficient range and resolution to specify the time.

Once the alarm goes off, it will repeat at the period set by Trigger Alarm Period the number of times set by Trigger Alarm Count.

4.2.9 Trigger Alarm Time Fraction (IVI-C and IVI-COM Only)

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

.NET Property Name

N/A

(Use the Trigger Alarm Time attribute.)

COM Property Name

Trigger.Alarms.Item().TimeFraction

COM Enumeration Name

N/A

C Constant Name

IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_FRACTION

Description

Specifies the fractional seconds portion of the time at which the alarm will go off. Note that the actual time of the alarm is the sum of Trigger Alarm Time Seconds and Trigger Alarm Time Fraction. The time is specified as the sum of two values because a single double-precision floating-point value does not have sufficient range and resolution to specify the time.

Once the alarm goes off, it will repeat at the period set by Trigger Alarm Period the number of times set by Trigger Alarm Count.

4.2.10 Trigger Count

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

.NET Property Name

Trigger.TriggerCount

COM Property Name

Trigger.TriggerCount

C Constant Name

IVILXISYNC_ATTR_TRIGGER_COUNT

Description

Specifies the number of times a trigger has to occur to complete a measurement; that is, the number of triggers that are accepted before the measurement must be armed again.

.NET Exceptions

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

4.2.11 Trigger Source

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

.NET Property Name

```
Trigger.TriggerSource
```

COM Property Name

```
Trigger.TriggerSource
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_SOURCE
```

Description

Specifies which of the available trigger sources to use as the signal for triggering the device-specific operation (for example, a measurement).

The value specified for this attribute may be one of the names in the IviLxiSyncTriggerSource repeated capability collection as returned from either the Trigger Source Name attribute (for IVI-COM and IVI.NET) or the GetTriggerSourceName function (for IVI-C).

The value specified for this attribute may also be one of the names in the IviLxiSyncTriggerAlarm repeated capability collection as returned from either the Trigger Alarm Name attribute (for IVI-COM and IVI.NET) or the GetTriggerAlarmName function (for IVI-C).

The name specified here may also correspond to a non-LXI trigger event. For instance, the caller can use this attribute to program the trigger source to external or immediate triggering, by specifying values such as “EXT” or “INT”. Such trigger source names are device-dependent.

If the device trigger source has been programmed to a non-LXI event using an attribute or function other than the Trigger Source attribute, then this attribute shall return that value when read. For instance, if the specific driver implements an IVI instrument class and the class-compliant API is used to set the trigger source to external, then this property shall return a string that reflects the value set through the class-compliant API.

.NET Exceptions

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

4.2.12 Trigger Source Count

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

.NET Property Name

`Trigger.Sources.Count`

This property is inherited from `IIviRepeatedCapabilityCollection.`

COM Property Name

`Trigger.Sources.Count`

C Constant Name

`IVILXISYNC_ATTR_TRIGGER_SOURCE_COUNT`

Description

Returns the number of currently available trigger sources. The count returned includes any of the supported reserved repeated capability names defined in Section 2.1.6, *Reserved Repeated Capability Identifiers* as well as any custom repeated capability identifiers.

.NET Exceptions

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

4.2.13 Trigger Source Delay

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	IviLxiSyncTriggerSource	None	Configure Trigger Source
Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	IviLxiSyncTriggerSource	None	Configure Trigger Source

.NET Property Name

```
Trigger.Sources[].Delay
```

COM Property Name

```
Trigger.Sources.Item().Delay
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_SOURCE_DELAY
```

Description

Specifies the trigger source delay from when the trigger logic is satisfied until the device specific action (for instance a measurement) is triggered. A negative value implies pre-trigger acquisition. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

.NET Exceptions

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

4.2.14 Trigger Source Detection

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

.NET Property Name

`Trigger.Sources[].Detection`

.NET Enumeration Name

`Slope`

COM Property Name

`Trigger.Sources.Item().Detection`

COM Enumeration Name

`IviLxiSyncSourceDetectionEnum`

C Constant Name

`IVILXISYNC_ATTR_TRIGGER_SOURCE_DETECTION`

Description

Specifies the slope of the trigger source.

If the source is a LAN event and the source slope is set to positive, this Trigger repeated capability will be satisfied when the designated LAN packet arrives with a true indication. If the source slope is set to negative, this Trigger repeated capability will be satisfied when a LAN packet arrives with a false indication.

Defined Values

Name	***Description***		
		Language	***Identifier***
Rise	Configures the LXI device to trigger on the rising edge of the trigger source.		
		.NET	`Slope.Positive`
		C	`IVILXISYNC_VAL_DETECTION_RISE`
		COM	`IviLxiSyncTriggerSourceDetectionRise`
Fall	Configures the LXI device to trigger on the falling edge of the trigger source.		
		.NET	`Slope.Negative`
		C	`IVILXISYNC_VAL_DETECTION_FALL`
		COM	`IviLxiSyncTriggerSourceDetectionFall`

.NET Exceptions

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

4.2.15 Trigger Source EventId

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

.NET Property Name

```
Trigger.Sources[].EventId
```

COM Property Name

```
Trigger.Sources.Item().EventId
```

C Constant Name

```
IVILXISYNC_ATTR_ARM_SOURCE_EVENTID
```

Description

This specifies the LAN event identifier that is associated with this trigger source. LAN Events with this identifier are accepted from the source described in the filter.

The default value for EventId is the repeated capability specifier for this trigger source.

4.2.16 Trigger Source Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviLxiSyncTriggerSource*	RO	IviLxiSyncArmSource	None	N/A

.NET Property Name

```
Trigger.Sources[String name]
```

This indexer is inherited from `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular trigger source in the trigger sources collection.

COM Property Name

```
Trigger.Sources.Item ([in] BSTR SourceName)
```

C Constant Name

```
N/A
```

Description

Trigger Source Item uniquely identifies a trigger source in the trigger sources collection. It returns an interface pointer which can be used to control the attributes and other functionality of that trigger source.

The Item property takes a trigger source name. If the user passes an invalid value for the `SourceName` parameter, the property returns an error.

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

Return Values

If the IVI-COM driver cannot recognize the `Name` parameter, it returns an Unknown Name in Selector completion code as described in *IVI-3.2: Inherent Capabilities Specification*, Section 9.3.

.NET Exceptions

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

4.2.17 Trigger Source Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

`Trigger.Sources[].Name`

This property is inherited from `IIviRepeatedCapabilityIdentification.`

COM Property Name

`Trigger.Sources.Name ([in] LONG SourceIndex)`

C Constant Name

`N/A`

(Use the `GetTriggerSourceName` function.)

Description

Returns the physical repeated capability identifier defined by the specific driver for the trigger source that corresponds to the one-based index that the user specifies. For custom trigger sources added vith the Add Trigger Source function, this function returns the trigger source name in the original casing used when Add Trigger Source was called.

For C and COM, valid values for the `SourceIndex` parameter are between one and the value of the Trigger Source Count attribute. If the user passes an invalid value for the `SourceIndex` parameter, the value of this attribute is an empty string.

.NET Exceptions

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

4.2.18 Trigger Source Filter

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

.NET Property Name

```
Trigger.Sources[].Filter
```

COM Property Name

```
Trigger.Sources.Item().Filter
```

C Constant Name

```
IVILXISYNC_ATTR_TRIGGER_SOURCE_FILTER
```

Description

Specifies a filter for restricting trigger sources. The filter specified by this attribute denotes the *accepted* sources. The syntax for specifying a filter is as follows:

<Filter> == [(<tcp> | <udp> | <any>) [, <Filter>]]

<tcp> == <host> [:<port>]

<udp> == ALL [: <port>]

<any> == : <port>

<host> is either a hostname or host number. Note that the hostname can not be "ALL" since that would indicate the <udp> construct.

<port> is a series of decimal digits indicating the port number.

Specifying an empty string or VI_NULL means that LXI trigger packets are accepted via either TCP or UDP multicast from any host. ":5044" is equivalent to the empty string since 5044 is the IANA registered port for LXI events (lxi-evntsvc).

In the <tcp>, <udp> and <any> constructs, <port> refers to the port the device receives the LAN message on. If <port> is omitted from <tcp> or <udp>, packets are only accepted on the IANA registered port for LXI events (lxi-evntsvc).

Specifying the <host> (<tcp> construct) indicates that packets via TCP on the port indicated are accepted.

Specifying ALL (<udp> construct) indicates that UDP multicast packets are accepted if they are directed to the IANA registered port for LXI events (lxi-evntsvc) on the IANA registered multicast address (LXI-EVENT). No TCP packets are accepted unless a <tcp> syntax is also included in the filter. The multicast address can not be altered with this syntax.

Specifying any protocol (<any> construct) indicates that both TCP and UDP multicast packets are accepted if they are directed to the specified port. UDP multicast packets must be received at the IANA registered multicast addres (LXI-EVENT).

The send port is not monitored. This allows the transmitter to use any available port.

Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible filters syntaxes.

White space shall be ignored. The <Filter> string is case insensitive.

Conventional devices should consider restricting the <port> to only the IANA registered port for LXI events (lxi-eventsvc) and not accepting the generalized syntax.

Example Filter	Description
"192.168.0.1:23"	Accepts trigger source packets via TCP on port 23 from the host at IP address 192.168.0.1.
"A_SIGGEN1:23,A_SPECAN2:23"	Accepts trigger source packets via TCP on port 23 from the host with DNS name A_SIGGEN1 as well as packets via TCP on port 23 on the host with DNS name A_SPECAN2.
"192.168.0.1"	Accepts trigger source packets from via TCP on the IANA registered port for LXI events (lxi-eventsvc) from the host at IP address 192.168.0.1.
"All:23,A_SPECAN2"	Accepts trigger source packets via UDP multicast on the IANA assigned multicast address to port 23 from any host as well as packets via TCP on the IANA registered port for LXI events (lxi-eventsvc) from the host with DNS name A_SPECAN2.
""	Accepts trigger source packets via TCP or UDP multicast on the IANA assigned multicast address on the IANA registered port for LXI events (lxi-eventsvc) from any host. This is equivalent to <any> with the IANA registered port for LXI events (lxi-evntsvc).
"All"	Accepts trigger source packets via UDP multicast on the IANA assigned multicast address on the IANA registered port for LXI events (lxi-eventsvc) from any host. This is equivalent to "".
"All:8543"	Accepts trigger source packets arriving via UDP multicast on the IANA UDP multicast address on port 8543 from any host.
"All, 192.168.1.1"	Accepts trigger source packets arriving via UDP multicast on the IANA assigned multicast address at the IANA registered port for LXI events (lxi-eventsvc), or TCP packets arriving on the IANA registered port for LXI events (lxi-eventsvc) from IP address 192.168.1.1
":23"	Accepts trigger source packets arriving via either UDP multicast or TCP on port 23 from any host.

.NET 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.

4.3 IviLxiSyncTrigger Functions

The IviLxiSyncTrigger subsystem defines the following functions:

- Add Trigger Alarm
- Add Trigger Source
- Configure Trigger Alarm
- Configure Trigger Source
- Disable All Trigger Alarms
- Get Trigger Alarm Name (IVI-C Only)
- Get Trigger Source Name (IVI-C Only)
- Remove Trigger Alarm
- Remove Trigger Source
- Remove All Custom Trigger Sources
- Remove All Trigger Alarms

This section describes the behavior and requirements of each function.

4.3.1 Add Trigger Alarm

Description

This function creates a new trigger alarm.

When a new alarm is added, the default values for the associated attributes are as follows:

Attribute	Default Value
Trigger Alarm Enabled	False
Trigger Alarm Period	0
Trigger Alarm Repeat Count	1
Trigger Alarm Time (.NET only)	00:00:00 Jan 1, 1970
Trigger Alarm Time Fraction (C/COM only)	0
Trigger Alarm Time Seconds (C/COM only)	0

.NET Method Prototype

```
IIviLxiSyncTriggerAlarm Trigger.Alarms.Add(String alarmName);
```

COM Method Prototype

```
HRESULT Trigger.Alarms.Add([in] BSTR AlarmName);
```

C Prototype

```
ViStatus IviLxiSyncTriger_AddTriggerAlarm (ViSession Vi,
                                           ViConstString AlarmName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
AlarmName	Specifies the name of the trigger alarm to create.	ViConstString

Outputs	Description	Base Type
Return value (.NET)	A reference to the trigger alarm object that was added .	IIviLxiSyncTriggerAlarm

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 status codes for this function.

Completion Codes	Description
Alarm Exists	Error: Alarm already exists.

.NET 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
AlarmExistsException	Alarm already exists.

4.3.2 Add Trigger Source

Description

This function creates a new trigger source.

When a new alarm is added, the default values for the associated attributes are as follows:

Attribute	Default Value
`Trigger Source Delay`	0
`Trigger Source Detection`	Positive
`Trigger EventId`	The repeated capability source name specified in this call
`Trigger Source Filter`	"" (Empty string)

.NET Method Prototype

```
IIviLxiSyncTriggerSource Trigger.Sources.Add(String sourceName);
```

COM Method Prototype

```
HRESULT Trigger.Sources.Add([in] BSTR SourceName);
```

C Prototype

```
ViStatus IviLxiSyncTriger_AddTriggerSource (ViSession Vi,
                                            ViConstString SourceName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`SourceName`	Specifies the name of the trigger source to create.	`ViConstString`

Outputs	Description	Base Type
`Return value (.NET)`	A reference to the trigger source object that was added .	`IIviLxiSyncTriggerSource`

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 status codes for this function.

Completion Codes	Description
Event Source Exists	Error: Event source already exists.
Out of Event Resources	Error: Out of event resources.

.NET 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
EventSourceExistsException	Event source already exists.
OutOfEventResourcesException	Out of event resources.

4.3.3 Configure Trigger Alarm

Description

This function configures the most commonly used attributes of the trigger alarm subsystem.

.NET Method Prototype

```
void Trigger.Alarms[].Configure(PrecisionDateTime time,
                                PrecisionTimeSpan period,
                                Int32 repeatCount);
```

COM Method Prototype

```
HRESULT Trigger.Alarms.Item().Configure([in] DOUBLE TimeSeconds,
                                        [in] DOUBLE TimeFraction,
                                        [in] DOUBLE Period,
                                        [in] LONG RepeatCount);
```

C Prototype

```
ViStatus IviLxiSync_ConfigureTriggerAlarm (ViSession Vi,
                                           ViConstString AlarmName,
                                           ViReal64 TimeSeconds,
                                           ViReal64 TimeFraction,
                                           ViReal64 Period,
                                           ViInt32 RepeatCount);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
AlarmName	The name of the alarm.	ViConstString
time (.NET)	Specifies the date and time of the trigger alarm. The driver uses this value to set the Trigger Alarm Time attribute. See the attribute description for more information.	Ivi.Driver. PrecisionDateTime
TimeSeconds (C/COM)	Specifies the seconds part of time. The driver uses this value to set the Trigger Alarm Time Seconds attribute. See the attribute description for more information.	ViReal64
TimeFraction (C/COM)	Specifies the fractional part of time. The driver uses this value to set the Trigger Alarm Time Fractional attribute. See the attribute description for more information.	ViReal64
Period	Specifies the period of the trigger alarm. The driver uses this value to set the Trigger Alarm Period attribute. See the attribute description for more information.	ViReal64 (C/COM) Ivi.Driver. PrecisionTimeSpan (.NET)
RepeatCount	Specifies the number of times to repeat the trigger at the period specified by the Trigger Alarm Repeat Period attribute. The driver uses this value to set the Trigger Alarm Repeat Count attribute. See the attribute description for more information.	ViInt32

Return Values (C/COM)

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

.NET Exceptions

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

4.3.4 Configure Trigger Source

Description

This function configures the most commonly used attributes of the trigger source subsystem.

.NET Method Prototype

```
void Trigger.Sources[].Configure(PrecisionTimeSpan delay,
                                 Slope detection);
```

COM Method Prototype

```
HRESULT Trigger.Sources.Item().Configure([in] DOUBLE Delay,
                                         [in] IviLxiSyncSourceDetectionEnum
Slope);
```

C Prototype

```
ViStatus IviLxiSync_ConfigureTriggerSource (ViSession Vi,
                                            ViConstString SourceName,
                                            ViReal64 Delay,
                                            ViInt32 Detection);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`SourceName`	Name of the trigger source to configure.	`ViConstString`
`Delay`	Specifies the trigger source delay. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan. A negative value implies pre-trigger acquisition. The driver uses this value to set the Trigger Source Delay attribute. See the attribute description for more information.	`ViReal64 (C/COM)` `Ivi.Driver. PrecisionTimeSpan (.NET)`
`Detection`	Specifies the slope of the trigger source. The driver uses this value to set the Trigger Source Detection attribute. See the attribute description for more information.	`ViInt32`

Return Values (C/COM)

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

.NET Exceptions

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

4.3.5 Disable All Trigger Alarms

Description

This function disables all trigger alarms. The specific driver uses this function to set the Trigger Alarm Enabled property to False for all trigger alarms.

.NET Method Prototype

```
void Trigger.Alarms.DisableAll();
```

COM Method Prototype

```
HRESULT Trigger.Alarms.DisableAll();
```

C Prototype

```
ViStatus IviLxiSync_DisableAllTriggerAlarms (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.

4.3.6 Get Trigger Alarm Name (IVI-C Only)

Description

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the `AlarmIndex` parameter is less than one or greater than the value of the Trigger Alarm Count attribute, the function returns an empty string in the `AlarmName` parameter and returns an error. For custom trigger sources added with the Add Arm Source function, this function returns the arm source name in the original casing used when Add Arm Source was called.

.NET Method Prototype

`N/A`

(use the `Trigger.Alarms[].Name` property)

COM Method Prototype

`N/A`

(Use the `Trigger.Alarms.Item().Name` property.)

C Prototype

```
ViStatus IviLxiSync_GetTriggerAlarmName (ViSession Vi,
                                         ViInt32 AlarmIndex,
                                         ViInt32 AlarmNameBufferSize,
                                         ViChar AlarmName[]);
```

Parameters

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

Outputs	Description	Data Type
`AlarmName`	The buffer into which the function returns the alarm name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the AlarmNameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

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

4.3.7 Get Trigger Source Name (IVI-C Only)

Description

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the `SourceIndex` parameter is less than one or greater than the value of the Trigger Source Count attribute, the function returns an empty string in the `SourceName` parameter and returns an error. For custom trigger sources added vith the Add Trigger Source function, this function returns the trigger source name in the original casing used when Add Trigger Source was called.

.NET Method Prototype

`N/A`

(use the `Trigger.Sources[].Name` property)

COM Method Prototype

`N/A`

(Use the `Trigger.Sources.Item().Name` property.)

C Prototype

```
ViStatus IviLxiSync_GetTriggerSourceName (ViSession Vi,
                                          ViInt32 SourceIndex,
                                          ViInt32 SourceNameBufferSize,
                                          ViChar SourceName[]);
```

Parameters

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

Outputs	Description	Data Type
`SourceName`	The buffer into which the function returns the source name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the SourceNameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

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

4.3.8 Remove Trigger Alarm

Description

This function removes a trigger alarm.

The AlarmName parameter is case-insensitive.

.NET Method Prototype

```
void Trigger.Alarms.Remove(String alarmName);
```

COM Method Prototype

```
HRESULT Trigger.Alarms.Remove([in] BSTR AlarmName);
```

C Prototype

```
ViStatus IviLxiSync_RemoveTriggerAlarm (ViSession Vi,
                                        ViConstString AlarmName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`AlarmName`	Specifies the name of the trigger alarm to remove.	`ViConstString`

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 status codes for this function.

Completion Codes	Description
Alarm Does Not Exist	Error: Alarm does not exist.
Attempt To Remove Reserved Repeated Capability	Error: The repeated capability name is reserved and cannot be removed.

.NET 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
AlarmDoesNotExistException	Alarm does not exist.
CannotRemoveReservedRepeatedCapabilityException	The repeated capability name is reserved and cannot be removed.

4.3.9 Remove Trigger Source

Description

This function removes a trigger source.

The SourceName parameter is case-insensitive.

.NET Method Prototype

```
void Trigger.Sources.Remove(String sourceName);
```

COM Method Prototype

```
HRESULT Trigger.Sources.Remove([in] BSTR SourceName);
```

C Prototype

```
ViStatus IviLxiSync_RemoveTriggerSource (ViSession Vi,
                                         ViConstString SourceName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
SourceName	Specifies the name of the trigger source to remove.	ViConstString

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 status codes for this function.

Completion Codes	Description
Event Source Does Not Exist	Error: Event source does not exist.
Attempt To Remove Reserved Repeated Capability	Error: The repeated capability name is reserved and cannot be removed.

.NET 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
EventSourceDoesNotExistException	Event source does not exist.
CannotRemoveReservedRepeatedCapabilityException	The repeated capability name is reserved and cannot be removed.

4.3.10 Remove All Custom Trigger Sources

Description

This function removes all of the custom trigger sources that were added using the Add Trigger Source function. The trigger sources associated with the reserved repeated capability identifiers, as defined in Section 2.1.6, *Reserved Repeated Capability Identifiers*, are not affected by this function.

.NET Method Prototype

```
void Trigger.Sources.RemoveAllCustomTriggerSources();
```

COM Method Prototype

```
HRESULT Trigger.Sources.RemoveAllCustomTriggerSources();
```

C Prototype

```
ViStatus IviLxiSync_RemoveAllCustomTriggerSources (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.

4.3.11 Remove All Trigger Alarms

Description

This function removes all of the trigger alarms that were added using the Add Trigger Alarm function.

.NET Method Prototype

```
void Trigger.Alarms.RemoveAllTriggerAlarms();
```

COM Method Prototype

```
HRESULT Trigger.Alarms.RemoveAllTriggerAlarms();
```

C Prototype

```
ViStatus IviLxiSync_RemoveAllTriggerAlarms (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.

5. IviLxiSyncEvent Subsystem

The IIviLxiSyncEvents interface provides a mechanism for configuration the generation of outbound events from an LXI device. Events can be transmitted over the LXI trigger bus, as LXI-defined LAN events, or as custom user-defined events. In addition, events can be transmitted to specific ports on specific hosts or to multiple hosts.

5.1 Behavior Model

The diagram below shows the behavior model for the IviLxiSyncEvent subsystem. This subsystem is responsible for routing signals to the appropriate event transmitter (either an LXI trigger bus line or a LAN event packet). All signals (not just signals from the Arm-Trigger state machine) which are intended to be utilized for sending events or routed to the LXI trigger bus, need to be connected to the input multiplexers in the event logic shown below.

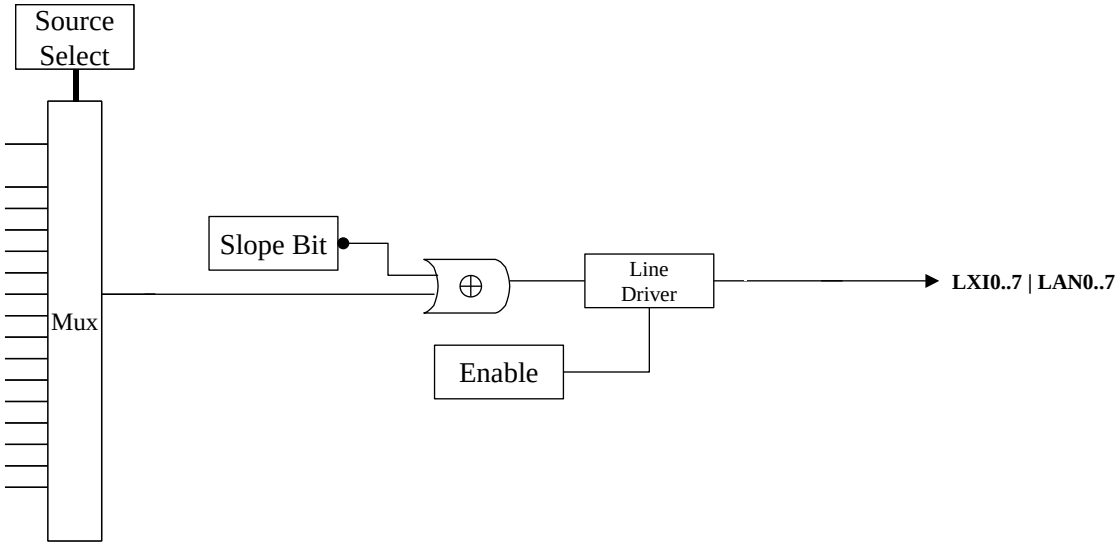


Figure 5. IviLxiSyncEvent Behavior Model

The LXI trigger bus may be used in either wired-OR or driven mode. To use the bus in wired-OR mode, one LXI device must be configured to provide the wired-OR bias. The device providing the bias can also either source events or respond to them on that trigger line.

For an LXI device to source events on the trigger bus, it must be configured to either drive the line in wired-OR or driven mode. Therefore, when an event is configured to drive a trigger line, the mode of that line must be specified.

When an LXI device is configured to source events used the LXI LAN protocol, it must be configured to source LAN packets on the leading edge of the event, the falling edge of the event or both. To accomplish this, if the event is configured to be wired-OR, then only logical true transitions are emitted. If the event is not configured to be wired-OR, then both rising and falling transitions result in LAN packets being sourced. Finally, if the system design requires only packets when the event becomes false, the device is configured in wired-OR mode, however the slope is set to false.

5.2 IviLxiSyncEvent Attributes

The IviLxiSyncEvent subsystem defines the following attributes:

- Event Count
- Event Destination Path
- Event Drive Mode
- Event Item (IVI-COM and IVI.NET Only)
- Event Name (IVI-COM and IVI.NET Only)
- Event Slope
- Event Source
- Event Wired OR Bias Mode

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

5.2.1 Event Count

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

.NET Property Name

`Events.Count`

This property is inherited from `IIviRepeatedCapabilityCollection.`

COM Property Name

`Events.Count`

C Constant Name

`IVILXISYNC_ATTR_EVENT_COUNT`

Description

Returns the number of defined events. The count returned includes any of the supported reserved repeated capability names defined in Section 2.1.6, *Reserved Repeated Capability Identifiers* as well as any custom 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.

5.2.2 Event Destination Path

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	IviLxiSyncEvent	None	Configure Event

.NET Property Name

```
Events[].DestinationPath
```

COM Property Name

```
Events.Item().DestinationPath
```

C Constant Name

```
IVILXISYNC_ATTR_EVENT_DESTINATION_PATH
```

Description

Specifies a list of places to send the event.

The default value for this attribute is the repeated capability name.

The grammar for the parameter is:

```
<DestinationPath>== [(<tcp>|<udp>|<TriggerBus>)[, <DestinationPath>]]
<tcp> == <host> [: <port>] [/<LANIdent>]
<udp> == [ALL] [: <port> ] [/<LANIdent>]
<TriggerBus> == LXI0|LXI1|LXI2|LXI3|LXI4|LXI5|LXI6|LXI7
```

host is either a hostname or host number, and port is a series of decimal digits indicating the port number. Note that the hostname can not be “ALL” or one of the <TriggerBus> designations since that would indicate the <udp> or <TriggerBus> construct.

<LANIdent> is a string indicating the LAN identifier that will be sent in the LAN message. The <LANIdent> is not case sensitive. <LANIdent> is from one to 16 ASCII characters inclusive. The characters may be numeric or underscore or hyphen or upper or lower-case alphabetic characters.

Defaults:

The default <DestinationPath> is the repeated capability name. This may either be a <TriggerBus> identifier or a <LANIdent> identifier.

<LANIdent> defaults to the repeated capability name.

<port> defaults to the IANA registered port for LXI events (lxi-evntsvc).

If the repeated capability name is not a trigger bus specifier then the default <DestinationPath> is ‘ALL’ with the <LANIdent> as the repeated capability name.

If the repeated capability name corresponds to a <TriggerBus>, the default <DestinationPath> is the repeated capability name.

If multiple <DestinationPath>s are specified, the event is transmitted to each.

The <tcp> construct specifies that a TCP message will be sent to the destination when the bound event occurs.

The <udp> construct specifies that a UDP multicast message will be sent to the IANA registered multi-cast address (LXI-EVENT) on the designated port. UDP unicast and UDP broadcasts are not supported by this syntax.

The <TriggerBus> construct specifies that a physical LXI wired trigger bus is used to signal the event.

Note that the LXI specification reserves event identifiers that begin with the characters "LXI" for LXI use. The strings "LXI0", "LXI1", … ,"LXI7" refer to the 8 LXI wired trigger bus triggers. See LXI 1.1 rule 6.4.5.

White space shall be ignored. The <Destination> string is case insensitive.

Drivers may accept additional vendor-defined syntaxes

Drivers (and the corresponding instruments) that support this syntax are permitted to not support all possible destination syntaxes.

Conventional devices should consider restricting the port to only the IANA registered port for LXI events (lxi-evntsvc) and not accepting the generalized syntax.

Example Destination Path	Description
"192.168.0.1:23/LAN2"	When the event designated by the event source property occurs, a TCP packet with event ID LAN2 is sent to 192.168.0.1, port 23. .
"LXI5"	When the event designated by the event source property occurs, the LXI5 wired-trigger-bus line generates a transition.
"ALL:23"	When the event designated by the event source property occurs, a UDP multicast packet is sent to the IANA assigned UDP multicast address port 23 with the event ID corresponding to the repeated capability name.
"A_SIGGEN1:23,A_SPECAN2:23"	When the event designated by the event source property occurs, TCP packets with the event ID corresponding to the repeated capability name are sent to hosts A_SIGGEN1 and A_SPECAN2 on port 23.
"192.168.0.1"	When the event designated by the event source property occurs, a TCP packet corresponding to the repeated capability name is sent to 192.168.0.1 on the IANA registered port for LXI events (lxi-eventsvc).
":23"	When the event designated by the event source property occurs, a UDP multicast packet with the event id corresponding to the repeated capability name is sent to the IANA designated multicast address, port 23
":23,A_SPECAN2"	When the event designated by the event source property occurs, a UDP multicast packet is sent to the IANA designated multicast address, port 23, subsequently, a TCP packet is sent to host A_SPECAN2 on the IANA registered port for LXI events (lxi-eventsvc)

.NET Exceptions

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

5.2.3 Event Drive Mode

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

.NET Property Name

`Events[].DriveMode`

COM Property Name

`Events.Item().DriveMode`

COM Enumeration Name

`IviLxiSyncEventDriveModeEnum`

C Constant Name

`IVILXISYNC_ATTR_EVENT_DRIVE_MODE`

Description

Specifies how this event is transmitted

It is an error to turn on the Wired OR Bias Mode for this device for a particular LXI trigger line and then set the Event Enabled attribute to On instead of Wired OR for an event whose destination path includes that LXI trigger line.

Defined Values

Name	***Description***		
		Language	***Identifier***
On	Enables the event in driven mode.		
		.NET	`EventDriveMode.On`
		C	`IVILXISYNC_VAL_EVENT_ON`
		COM	`IviLxiSyncEventDriveModeOn`
Off	Disables the event.		
		.NET	`EventDriveMode.Off`
		C	`IVILXISYNC_VAL_EVENT_OFF`
		COM	`IviLxiSyncEventDriveModeOff`
Wired OR	Enables the event in wired-OR mode.		
		.NET	`EventDriveMode.WiredOr`
		C	`IVILXISYNC_VAL_EVENT_WIREDOR`
		COM	`IviLxiSyncEventDriveModeWiredOr`

Return Values

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

Completion Codes	Description
Event Source Not Set	Error: Event source not set.
Wired OR Mode Invalid	Error: Event source cannot operate in driven mode while serving as the wired-OR bias.

.NET Exceptions

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

Completion Codes	Description
Event Source Not Set Exception	Event source not set.
Wired OR Mode Invalid Exception	Event source cannot operate in driven mode while serving as the wired-OR bias.

5.2.4 Event Item (IVI-COM and IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviLxiSyncEvent*	RO	IviLxiSyncEvent	None	N/A

.NET Property Name

```
Events[String name]
```

This indexer is inherited from the base interface `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular event in the events collection.

COM Property Name

```
Events.Item ([in] EventName)
```

C Constant Name

```
N/A
```

Description

Event Item uniquely identifies an event in the events collection. It returns an interface pointer which can be used to control the attributes and other functionality of that event.

The Item property may only take an event name. If the user passes an invalid value for the event name parameter, the property returns an error.

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

Return Values

If the IVI-COM driver cannot recognize the `Name` parameter, it returns an Unknown Name in Selector completion code as described in *IVI-3.2: Inherent Capabilities Specification*, Section 9.3.

.NET Exceptions

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

5.2.5 Event Name (IVI-COM and IVI.NET Only)

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

.NET Property Name

```
Events[].Name
```

The .NET property is inherited from `IIviRepeatedCapabilityIdentification`.

COM Property Name

```
Events.Name ([in] LONG EventIndex)
```

C Constant Name

```
N/A
```

(Use the `GetEventName` function.)

Description

Returns the physical repeated capability identifier defined by the specific driver for the event that corresponds to the one-based index that the user specifies. For custom event sources added vith the Add Event Source function, this function returns the event source name in the original casing used when Add Event Source was called.

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

.NET Exceptions

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

5.2.6 Event Slope

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

.NET Property Name

`Events[].Slope`

.NET Enumeration Name

`Slope`

COM Property Name

`Events.Item().Slope`

COM Enumeration Name

`IviLxiSyncSourceSlopeEnum`

C Constant Name

`IVILXISYNC_ATTR_EVENT_SLOPE`

Description

Specifies the slope of the event that is inbound to the event subsystem that will cause the generation of an outbound event. The outbound event shall be transmitted with the same slope as the inbound event.

Defined Values

Name	***Description***		
		Language	***Identifier***
Positive	The event will be transmitted with a rising edge.		
		.NET	`Slope.Positive`
		C	`IVILXISYNC_VAL_SLOPE_POSITIVE`
		COM	`IviLxiSyncSourceSlopePositive`
Negative	The event will be transmitted with a falling edge.		
		.NET	`Slope.Negative`
		C	`IVILXISYNC_VAL_SLOPE_NEGATIVE`
		COM	`IviLxiSyncSourceSlopeNegative`

.NET Exceptions

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

5.2.7 Event Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	IviLxiSyncEvent	None	Configure Event

.NET Property Name

Events[].Source

COM Property Name

Events.Item().Source

C Constant Name

IVILXISYNC_ATTR_EVENT_SOURCE

Description

Specifies the signal which causes an event to be transmitted.

In general, drivers will define events that are relevant for the instrument they support.

The following table lists reserved event source names that are defined by LXI and may be recognized by specific drivers according to their LXI Functional class.

Table 5-1. Reserved Event Source Names

Repeated Capability Identifier	Defined for LXI Functional Class
OperationComplete	Class A, Class B
Measuring	Class A, Class B
Settling	Class A, Class B
Sweeping	Class A, Class B
WaitingForArm	Class A, Class B
WaitingForTrigger	Class A, Class B
LAN0	Class A, Class B
LAN1	Class A, Class B
LAN2	Class A, Class B
LAN3	Class A, Class B
LAN4	Class A, Class B
LAN5	Class A, Class B
LAN6	Class A, Class B
LAN7	Class A, Class B
LXI0	Class A
LXI1	Class A
LXI2	Class A
LXI3	Class A

Table 5-1. Reserved Event Source Names

LXI3	Class A
LXI5	Class A
LXI6	Class A
LXI7	Class A

Return Values

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

Completion Codes	Description
Invalid Event Source	Error: Invalid source.
Out of Event Resources	Error: Out of event resources.

.NET Exceptions

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

Completion Codes	Description
Invalid Event Source Exception	Invalid source.
Out of Event Resources Exception	Out of event resources.

5.2.8 Event Wired OR Bias Mode

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	IviLxiSyncEvent	None	N/A
WiredOrBiasModeLines (.NET)	R/W	IviLxiSyncEvent	None	N/A

.NET Property Name

Events.WiredOrBiasMode

.NET Enumeration Name

WiredOrBiasModeLines

COM Property Name

Events.WiredOrBiasMode

C Constant Name

IVILXISYNC_ATTR_EVENT_WIRED_OR_BIAS_MODE

Description

Specifies whether this LXI device will serve as the wired-OR bias for specific LXI trigger bus lines.

For C and COM, the allowed values for this attribute are 0 to 255. This attribute is a bit field, where bit 0 represents LXI0, bit 1 represents LXI1, and so on. A value of one in a particular bit indicates that the LXI device shall serve as the bias for the corresponding trigger bus line. A value of zero in a particular bit disables the bias for the corresponding trigger bus line. To use a trigger bus line in driven mode, the bias must be disabled.

For .NET, multiple lines are specified by bit-wise OR'ing enumeration values, since the enumeration is a flags enumeration.

Enabling wired-OR bias has no impact on the device's ability to either respond to signals on trigger bus lines or to send events on trigger bus lines.

One and only one LXI device can serve as the wired-OR bias for a particular trigger bus line, although different devices can serve as the wired-OR bias for different trigger bus lines.

Defined Values

Note that the .NET enumeration is a flags enumeration.

Name	*Description*		
		Language	*Identifier*
LXI0	Wired Trigger Bus line LXI0.		
		.NET	`WiredOrBiasModeLines.Lxi0`
LXI1	Wired Trigger Bus line LXI1.		
		.NET	`WiredOrBiasModeLines.Lxi1`
LXI2	Wired Trigger Bus line LXI2.		
		.NET	`WiredOrBiasModeLines.Lxi2`
LXI3	Wired Trigger Bus line LXI3.		
		.NET	`WiredOrBiasModeLines.Lxi3`
LXI4	Wired Trigger Bus line LXI4.		
		.NET	`WiredOrBiasModeLines.Lxi4`
LXI5	Wired Trigger Bus line LXI5.		
		.NET	`WiredOrBiasModeLines.Lxi5`
LXI6	Wired Trigger Bus line LXI6.		
		.NET	`WiredOrBiasModeLines.Lxi6`
LXI7	Wired Trigger Bus line LXI7.		
		.NET	`WiredOrBiasModeLines.Lxi7`

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 property.

5.3 IviLxiSyncEvent Functions

The IviLxiSyncEvent subsystem defines the following functions:

- Add Event
- Configure Event
- Disable All Events
- GetEvent name (IVI-C Only)
- Remove Event
- Remove All Custom Events

This section describes the behavior and requirements of each function.

5.3.1 Add Event

Description

This function creates a new event.

When a new event is added, the default values for the associated attributes are as follows:

Attribute	Default Value
`Event Enabled`	False
`Event Slope`	Positive
`Event Source`	"" (Empty string)
`Event Destination Path`	Current repeated capability instance name

.NET Method Prototype

```
IIviLxiSyncEvent Events.Add(String eventName);
```

COM Method Prototype

```
HRESULT Events.Add([in] BSTR EventName);
```

C Prototype

```
ViStatus IviLxiSync_AddEvent (ViSession Vi,
                              ViConstString EventName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`EventName`	Specifies the name of the event to create.	`ViConstString`

Outputs	Description	Base Type
`Return value (.NET)`	A reference to the event object that was added .	`IIviLxiSyncEvent`

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 status codes for this function.

Completion Codes	Description
Event Source Exists	Error: Event source already exists.
Out of Event Resources	Error: Out of event resources.

.NET 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
EventSourceExistsException	Event source already exists.
OutOfEventResourcesException	Out of event resources.

5.3.2 Configure Event

Description

This function configures the most commonly used attributes of the event subsystem.

.NET Method Prototype

```
void Events[].Configure(EventDriveMode driveMode,
                        String source,
                        String destinationPath,
                        Slope slope);
```

COM Method Prototype

```
HRESULT Events.Item().Configure([in] IviLxiSyncEventDriveModeEnum DriveMode,
                                [in] BSTR Source,
                                [in] BSTR DestinationPath,
                                [in] IviLxiSyncSourceSlopeEnum Slope);
```

C Prototype

```
ViStatus IviLxiSync_ConfigureEvent (ViSession Vi,
                                    ViConstString repCapIdentifier,
                                    ViInt32 DriveMode,
                                    ViConstString Source,
                                    ViConstString DestinationPath,
                                    ViInt32 Slope);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
repCapIdentifier	Specifies the name of the event to configure.	ViConstString
DriveMode	Specifies the mode of the event. The driver uses this value to set the Event Drive Mode attribute. See the attribute description for more information.	ViInt32
Source	Specifies the signal which causes an event to be transmitted. The driver uses this value to set the Event Source attribute. See the attribute description for more information.	ViConstString
DestinationPath	Specifies a list of places to send the event. The driver uses this value to set the Event Destination Path attribute. See the attribute description for more information.	ViConstString
Slope	Specifies the slope of the event signal. The driver uses this value to set the Event Slope attribute. See the attribute description for more information.	ViInt32

Return Values (C/COM)

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

.NET Exceptions

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

5.3.3 Disable All Events

Description

This function disables all events. The specific driver uses this function to set the Event Drive Mode property to Off for all events.

.NET Method Prototype

```
void Events.DisableAll();
```

COM Method Prototype

```
HRESULT Events.DisableAll();
```

C Prototype

```
ViStatus IviLxiSync_DisableAllEvents (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.

5.3.4 Get Event Name (IVI-C Only)

Description

This function returns the physical repeated capability identifier that corresponds to the one-based index that the user specifies. If the value that the user passes for the `EventIndex` parameter is less than one or greater than the value of the Event Count attribute, the function returns an empty string in the `EventName` parameter and returns an error. For custom event sources added vith the Add Event Source function, this function returns the event source name in the original casing used when Add Event Source was called.

.NET Method Prototype

`N/A`

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

COM Method Prototype

`N/A`

(Use the `Events.Item().Name` property)

C Prototype

```
ViStatus IviLxiSync_GetEventName (ViSession Vi,
                                  ViInt32 EventIndex,
                                  ViInt32 EventNameBufferSize,
                                  ViChar EventName[]);
```

Parameters

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

Outputs	Description	Data Type
`EventName`	The buffer into which the function returns the alarm name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the EventNameBufferSize parameter is 0.	`ViChar[]`

Return Values (C)

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

.NET Exceptions

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

5.3.5 Remove Event

Description

This function removes an event.

The EventName parameter is case-insensitive.

.NET Method Prototype

```
void Events.Remove(String eventName);
```

COM Method Prototype

```
HRESULT Events.Remove([in] BSTR EventName);
```

C Prototype

```
ViStatus IviLxiSync_RemoveEvent (ViSession Vi,
                                 ViConstString EventName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
EventName	Specifies the name of the event to remove.	ViConstString

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 status codes for this function.

Completion Codes	Description
Event Source Does Not Exist	Error: Event source does not exist.
Attempt To Remove Reserved Repeated Capability	Error: The repeated capability name is reserved and cannot be removed.

.NET 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
EventSourceDoesNotExistException	Event source does not exist.
CannotRemoveReservedRepeatedCapabilityException	The repeated capability name is reserved and cannot be removed.

5.3.6 Remove All Custom Events

Description

This function removes all of the custom events that were added using the Add Event function. The events associated with the reserved repeated capability identifiers, as defined in Section 2.1.6, *Reserved Repeated Capability Identifiers*, are not affected by this function.

.NET Method Prototype

```
void Events.RemoveAllCustomEvents();
```

COM Method Prototype

```
HRESULT Events.RemoveAllCustomEvents();
```

C Prototype

```
ViStatus IviLxiSync_RemoveAllCustomEvents (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.

6. IviLxiSyncEventLog Subsystem

6.1 IviLxiSyncEventLog Attributes

The IviLxiSyncEventLog subsystem defines the following attributes:

- Event Log Enabled
- Event Log Entry Count

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

6.1.1 Event Log Entry Count

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

.NET Property Name

EventLog.EntryCount

COM Property Name

EventLog.EntryCount

C Constant Name

IVILXISYNC_ATTR_EVENT_LOG_ENTRY_COUNT

Description

Returns the number of event log entries available.

.NET 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.

6.1.2 Event Log Enabled

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

.NET Property Name

EventLog.Enabled

COM Property Name

EventLog.Enabled

C Constant Name

IVILXISYNC_ATTR_EVENT_LOG_ENABLED

Description

If set to True, the LXI device enables the event logging feature. If set to False, the LXI device disables the event logging feature.

Compliance Notes

The specific driver shall implement both the True and False values.

.NET 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 IviLxiSyncEventLog Functions

The IviLxiSyncEventLog subsystem defines the following functions:

- Clear Event Log
- Get Next Event Log Entry

This section describes the behavior and requirements of each function.

6.2.1 Clear Event Log Entries

Description

This function removes all existing entries from the event log.

.NET Method Prototype

```
void EventLog.ClearEntries();
```

COM Method Prototype

```
HRESULT EventLog.ClearEntries();
```

C Prototype

```
ViStatus IviLxiSync_ClearEventLogEntries (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.

6.2.2 Get Next Event Log Entry

Description

This function retrieves and clears the oldest event log entry for the IVI session. If there are no entries in the event log, then the function returns an empty string in the LogEntry parameter.

The following rules apply to the C interface of the Get Next Event Log Entry function:

- The function complies with the rules in IVI-3.2, *Inherent Capabilities*, Section 3.1.2.1, *Additional Compliance Rules for C Functions with ViChar Array Output Parameters*.
- If the user passes 0 for the `LogEntryBufferSize` parameter, the function does not clear the oldest event log entry from the list.

.NET Method Prototype

```
String EventLog.GetNextEntry();
```

COM Method Prototype

```
HRESULT EventLog.GetNextEntry([out, retval] BSTR* LogEntry);
```

C Prototype

```
ViStatus IviLxiSync_GetNextEventLogEntry (ViSession Vi,
                                          ViInt32 LogEntryBufferSize,
                                          ViChar LogEntry[]);
```

Parameters

Inputs	Description	Data Type
`Vi`	Unique identifier for an IVI session.	`ViSession`
`LogEntryBufferSize`	The number of bytes in the `ViChar` array that the user specifies for the `LogEntry` parameter.	`ViInt32`

Outputs	Description	Data Type
`LogEntry (C/COM)`	The buffer in which the function returns the oldest event log entry. Can be `VI_NULL` if `LogEntryBufferSize` is 0.	`ViChar[]`
`Return Value (.NET)`	The oldest event log entry.	`String`

Return Values (C/COM)

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

.NET Exceptions

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

7. IviLxiSyncTime Subsystem

7.1 IviLxiSyncTime Attributes

The IviLxiSyncTime subsystem defines the following attributes:

- Is Time Master
- Is Time Synchronized
- System Time (IVI.NET Only)

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

7.1.1 Is Time Master

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

.NET Property Name

Time.IsMaster

COM Property Name

Time.IsMaster

C Constant Name

IVILXISYNC_ATTR_IS_TIME_MASTER

Description

If True, the device is the 1588 master. If False, the device is not the 1588 master.

Compliance Notes

The specific driver shall implement both the True and False values.

.NET Exceptions

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

7.1.2 Is Time Synchronized

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

.NET Property Name

```
Time.IsSynchronized
```

COM Property Name

```
Time.IsSynchronized
```

C Constant Name

```
IVILXISYNC_ATTR_IS_TIME_SYNCHRONIZED
```

Description

If True, the LXI device is synchronized with the 1588 master. If False, the LXI device is not synchronized with the 1588 master.

Compliance Notes

The specific driver shall implement both the True and False values.

.NET Exceptions

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

7.1.3 System Time (IVI.NET Only)

Data Type	Access	Applies to	Coercion	High Level Functions
`Ivi.Driver.PrecisionDateTime`	`RO`	N/A	None	N/A

.NET Property Name

`Time.SystemTime`

COM Property Name

`N/A`

(Use the `GetSystemTime` method.)

C Constant Name

`N/A`

(Use the `GetSystemTime` function.)

Description

The system time, expressed as a `PrecisionDateTime` object.

.NET Exceptions

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

7.2 IviLxiSyncTime Functions

The IviLxiSyncTime subsystem defines the following functions:

- Get System Time (IVI-C and IVI-COM Only)

This section describes the behavior and requirements of each function.

7.2.1 Get System Time (IVI-C and IVI-COM Only)

Description

This function retrieves the current 1588 time.

.NET Method Prototype

`N/A`

(Use the `SystemTime` property)

COM Method Prototype

```
HRESULT Time.GetSystemTime([in, out] double* TimeSeconds,
                           [in, out] double* TimeFractional);
```

C Prototype

```
ViStatus IviLxiSync_GetSystemTime (ViSession Vi,
                                   ViReal64* TimeSeconds,
                                   ViReal64* TimeFractional);
```

Parameters

Inputs	Description	Data Type
`Vi`	Unique identifier for an IVI session.	`ViSession`

Outputs	Description	Data Type
`TimeSeconds`	Indicates the seconds portion of the current 1588 time.	`ViReal64`
`TimeFractional`	Indicates the fractional portion of the current 1588 time.	`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. Attribute ID Definitions

The following table defines the ID value for all IviLxiSync attributes. Each value is defined in terms of the IVILXISYNC_ATTR_BASE base ID, which is itself defined in terms of IVI_ATTR_BASE as follows:

IVILXISYNC_ATTR_BASE = IVI_ATTR_BASE + 950000

Table 8-1. IviLxiSync Attribute ID Values

Attribute Name	ID Value
IVILXISYNC_ATTR_ARM_ALARM_COUNT	IVILXISYNC_ATTR_BASE + 1
IVILXISYNC_ATTR_ARM_ALARM_ENABLED	IVILXISYNC_ATTR_BASE + 2
IVILXISYNC_ATTR_ARM_ALARM_PERIOD	IVILXISYNC_ATTR_BASE + 3
IVILXISYNC_ATTR_ARM_ALARM_REPEAT_COUNT	IVILXISYNC_ATTR_BASE + 4
IVILXISYNC_ATTR_ARM_ALARM_TIME_FRACTION	IVILXISYNC_ATTR_BASE + 5
IVILXISYNC_ATTR_ARM_ALARM_TIME_SECONDS	IVILXISYNC_ATTR_BASE + 6
IVILXISYNC_ATTR_ARM_COUNT	IVILXISYNC_ATTR_BASE + 7
IVILXISYNC_ATTR_ARM_DELAY	IVILXISYNC_ATTR_BASE + 8
IVILXISYNC_ATTR_ARM_SOURCE_COUNT	IVILXISYNC_ATTR_BASE + 9
IVILXISYNC_ATTR_ARM_SOURCE_DETECTION	IVILXISYNC_ATTR_BASE + 10
IVILXISYNC_ATTR_ARM_SOURCE_ENABLED	IVILXISYNC_ATTR_BASE + 11
IVILXISYNC_ATTR_ARM_SOURCE_EVENTID	IVILXISYNC_ATTR_BASE + 12
IVILXISYNC_ATTR_ARM_SOURCE_FILTER	IVILXISYNC_ATTR_BASE + 13
IVILXISYNC_ATTR_ARM_SOURCE_OR_ENABLED	IVILXISYNC_ATTR_BASE + 14
IVILXISYNC_ATTR_TRIGGER_ALARM_COUNT	IVILXISYNC_ATTR_BASE + 15
IVILXISYNC_ATTR_TRIGGER_ALARM_ENABLED	IVILXISYNC_ATTR_BASE + 16
IVILXISYNC_ATTR_TRIGGER_ALARM_PERIOD	IVILXISYNC_ATTR_BASE + 17
IVILXISYNC_ATTR_TRIGGER_ALARM_REPEAT_COUNT	IVILXISYNC_ATTR_BASE + 18
IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_FRACTION	IVILXISYNC_ATTR_BASE + 19
IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_SECONDS	IVILXISYNC_ATTR_BASE + 20
IVILXISYNC_ATTR_TRIGGER_COUNT	IVILXISYNC_ATTR_BASE + 21
IVILXISYNC_ATTR_TRIGGER_SOURCE	IVILXISYNC_ATTR_BASE + 22
IVILXISYNC_ATTR_TRIGGER_SOURCE_COUNT	IVILXISYNC_ATTR_BASE + 23
IVILXISYNC_ATTR_TRIGGER_SOURCE_DELAY	IVILXISYNC_ATTR_BASE + 24
IVILXISYNC_ATTR_TRIGGER_SOURCE_DETECTION	IVILXISYNC_ATTR_BASE + 25
IVILXISYNC_ATTR_TRIGGER_SOURCE_EVENTID	IVILXISYNC_ATTR_BASE + 26
IVILXISYNC_ATTR_TRIGGER_SOURCE_FILTER	IVILXISYNC_ATTR_BASE + 27
IVILXISYNC_ATTR_EVENT_COUNT	IVILXISYNC_ATTR_BASE + 28
IVILXISYNC_ATTR_EVENT_DESTINATION_PATH	IVILXISYNC_ATTR_BASE + 29
IVILXISYNC_ATTR_EVENT_DRIVE_MODE	IVILXISYNC_ATTR_BASE + 30
IVILXISYNC_ATTR_EVENT_SLOPE	IVILXISYNC_ATTR_BASE + 31
IVILXISYNC_ATTR_EVENT_SOURCE	IVILXISYNC_ATTR_BASE + 32
IVILXISYNC_ATTR_EVENT_WIRED_OR_BIAS_MODE	IVILXISYNC_ATTR_BASE + 33

Table 8-1. IviLxiSync Attribute ID Values

Attribute Name	ID Value
IVILXISYNC_ATTR_EVENT_LOG_ENABLED	IVILXISYNC_ATTR_BASE + 34
IVILXISYNC_ATTR_EVENT_LOG_ENTRY_COUNT	IVILXISYNC_ATTR_BASE + 35
IVILXISYNC_ATTR_IS_TIME_MASTER	IVILXISYNC_ATTR_BASE + 36
IVILXISYNC_ATTR_IS_TIME_SYNCHRONIZED	IVILXISYNC_ATTR_BASE + 37

9. Attribute Value Definitions

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

Arm and Trigger Alarm Repeat Count

Value Name	Language	Identifier	Actual Value
Continuous	C	IVILXISYNC_VAL_REPEAT_CONTINUOUS	0
	COM	N/A	N/A

Arm Source Detection

Value Name	Language	Identifier	Actual Value
Rise	.NET	ArmSourceDetection.Rise	0
	C	IVILXISYNC_VAL_DETECTION_RISE	0
	COM	IviLxiSyncArmSourceDetectionRise	0
Fall	.NET	ArmSourceDetection.Fall	1
	C	IVILXISYNC_VAL_DETECTION_FALL	1
	COM	IviLxiSyncArmSourceDetectionFall	1
High	.NET	ArmSourceDetection.High	2
	C	IVILXISYNC_VAL_DETECTION_HIGH	2
	COM	IviLxiSyncArmSourceDetectionHigh	2
Low	.NET	ArmSourceDetection.Low	3
	C	IVILXISYNC_VAL_DETECTION_LOW	3
	COM	IviLxiSyncArmSourceDetectionLow	3

Event Drive Mode

Value Name	Language	Identifier	Actual Value
On	.NET	EventDriveMode.Driven	0
	C	IVILXISYNC_VAL_EVENT_DRIVEN	0
	COM	IviLxiSyncEventDriveModeDriven	0
Off	.NET	EventDriveMode.Off	1
	C	IVILXISYNC_VAL_EVENT_OFF	1
	COM	IviLxiSyncEventDriveModeOn	1
Wired OR	.NET	EventDriveMode.WiredOr	2
	C	IVILXISYNC_VAL_EVENT_WIREDOR	2
	COM	IviLxiSyncEventDriveModeWiredOr	2

Source Slope

Value Name	Language	Identifier	Actual Value
Rise	.NET	`Slope.Positive`	0
	C	`IVILXISYNC_VAL_SLOPE_RISE`	0
	COM	`IviLxiSyncSourceSlopeRise`	0
Fall	.NET	`Slope.Negative`	1
	C	`IVILXISYNC_VAL_SLOPE_FALL`	1
	COM	`IviLxiSyncSourceSlopeFall`	1

Trigger Source Detection

Value Name	Language	Identifier	Actual Value
Rise	.NET	`Slope.Positive`	0
	C	`IVILXISYNC_VAL_DETECTION_RISE`	0
	COM	`IviLxiSyncTriggerSourceDetectionRise`	0
Fall	.NET	`Slope.Negative`	1
	C	`IVILXISYNC_VAL_DETECTION_FALL`	1
	COM	`IviLxiSyncTriggerSourceDetectionFall`	1

10. Function Parameter Value Definitions

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

Configure Arm Source

Legal values for the Detection parameter are the same as those for Arm Source Detection in section 9, *Attribute Value Definitions*.

Configure Event

Legal values for the DriveMode parameter are the same as those for Event Drive Mode in section 9, *Attribute Value Definitions*.

Legal values for the Slope parameter are the same as those for Source Slope in section 9, *Attribute Value Definitions*.

Configure Trigger Source

Legal values for the Detection parameter are the same as those for Trigger Source Detection in section 9, *Attribute Value Definitions*.

11. Error and Completion Code Value Definitions

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

Table 11-1. IviLxiSync Error and Completion Codes

Error Name	*Description*			
		Language	*Identifier*	*Value(hex)*
Alarm Time Invalid	The alarm time is not valid. For instance, the time may have already passed.			
		.NET	AlarmTimeInvalidException	N/A
		C	IVILXISYNC_ALARM_TIME_INVALID	0xBFFA3001
		COM	E_IVILXISYNC_ALARM_TIME_INVALID	0x80043001
Event Source Exists	The event source is already a member of the collection.			
		.NET	EventSourceExistsException	N/A
		C	IVILXISYNC_EVENT_SOURCE_EXISTS	0xBFFA3002
		COM	E_IVILXISYNC_EVENT_SOURCE_EXISTS	0x80043002
Out of Event Resources	The device has no more event resources to allocate.			
		.NET	OutOfEventResourcesException	N/A
		C	IVILXISYNC_OUT_OF_EVENT_RESOURCES	0xBFFA3003
		COM	E_IVILXISYNC_OUT_OF_EVENT_RESOURCES	0x80043003
Event Source Does Not Exist	The specified event source has not been defined.			
		.NET	EventSourceDoesNotExistException	N/A
		C	IVILXISYNC_EVENT_SOURCE_DOES_NOT_EXISTS	0xBFFA3004
		COM	E_IVILXISYNC_EVENT_SORUCE_DOES_NOT_EXIST	0x80043004
Event Source Not Set	The event cannot be enabled while the event source has not been set.			
		.NET	EventSourceNotSetException	N/A
		C	IVILXISYNC_EVENT_SOURCE_NOT_SET	0xBFFA3005
		COM	E_IVILXISYNC_EVENT_SOURCE_NOT_SET	0x80043005
Invalid Event Source	The specified name is not a valid event source.			
		.NET	InvalidEventSourceException	N/A
		C	IVILXISYNC_INVALID_EVENT_SOURCE	0xBFFA3006
		COM	E_IVILXISYNC_INVALID_EVENT_SOURCE	0x80043006
Alarm Exists	The alarm is already a member of the collection.			
		.NET	AlarmExistsException	N/A
		C	IVILXISYNC_ALARM_EXISTS	0xBFFA3007
		COM	E_IVILXISYNC_ALARM_EXISTS	0x80043007

Table 11-1. IviLxiSync Error and Completion Codes

Error Name	*Description*			
		Language	*Identifier*	*Value(hex)*
Alarm Does Not Exist	The specified alarm has not been defined.			
		.NET	AlarmDoesNotExistException	N/A
		C	IVILXISYNC_ALARM_DOES_NOT_EXIST	0xBFFA3008
		COM	E_IVILXISYNC_ALARM_DOES_NOT_EXIST	0x80043008
Wired OR Mode Invalid	The event source cannot operate in driven mode while serving as the wired-OR bias.			
		.NET	WiredOrModeInvalidException	N/A
		C	IVILXISYNC_WIRED_OR_MODE_INVALID	0xBFFA3009
		COM	E_IVILXISYNC_WIRED_OR_MODE_INVALID	0x80043009
Attempt To Remove Reserved Repeated Capability	The repeated capability name is reserved and cannot be removed.			
		.NET	CannotRemoveReservedRepeatedCapabilityException	N/A
		C	IVILXISYNC_CANT_REMOVE_RESERVED_REPEATED_CAPABILITY	0xBFFA3010
		COM	E_IVILXISYNC_CANT_REMOVE_RESERVED_REPEATED_CAPABILITY	0x80043010

Table 10-2 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.

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

Table 11-2. IviLxiSync Error Message Strings

Name	Message String
Alarm Time Invalid	"%s: The alarm time is invalid"
Event Source Exists	"%s: The event source already exists"
Out of Event Resources	"%s: Out of event resources"
Event Source Does Not Exist	"%s: The specified event source does not exist"
Event Source Not Set	"%s: The event source has not been specified"
Invalid Event Source	"%s: The specified event source is not valid"
Alarm Exists	"%s: The alarms already exists"
Alarm Does Not Exist	"%s: The specified alarm has not been defined"
Wired OR Mode Invalid	"%s: The event source cannot operate in driven mode while serving as the wired-OR bias"

Table 11-2. IviLxiSync Error Message Strings

Name	Message String
Attempt to Remove Reserved Repeated Capability	"%s: The reserved repeated capability cannot be removed from the collection"

11.1 IVI.NET IviLxiSync Exceptions and Warnings

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

- AlarmDoesNotExistException
- AlarmExistsException
- AlarmTimeInvalidException
- CannotResolveReservedRepeatedCapabilityException
- EventSourceDoesNotExistException
- EventSourceExistsException
- EventSourceNotSetException
- InvalidEventSourceException
- OutOfEventResourcesException
- WiredOrModeInvalidException

11.1.1 AlarmDoesNotExistException

Description

This exception is used when the driver finds that a specified alarm has not been defined.

Constructors

```
Ivi.LxiSync.AlarmDoesNotExistException(String message
                                       String alarmName);

Ivi.LxiSync.AlarmDoesNotExistException();

Ivi.LxiSync.AlarmDoesNotExistException(String message);

Ivi.LxiSync.AlarmDoesNotExistException(String message,
                                       System.Exception innerException);
```

Message String

```
The specified alarm has not been defined.
Alarm name: <alarmName>
```

Parameters

Inputs	Description	Base Type
`alarmName`	The name of the alarm that is undefined.	`String`

Usage

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

11.1.2 AlarmExistsException

Description

This exception is used when the driver finds that a specified alarm already exists.

Constructors

```
Ivi.LxiSync.AlarmExistsException(String message
                                 String alarmName);

Ivi.LxiSync.AlarmExistsException();

Ivi.LxiSync.AlarmExistsException(String message);

Ivi.LxiSync.AlarmExistsException(String message,
                                 System.Exception innerException);
```

Message String

```
The specified alarm already exists.
Alarm name: <alarmName>
```

Parameters

Inputs	Description	Base Type
`alarmName`	The name of the alarm.	`String`

Usage

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

11.1.3 AlarmTimeInvalidException

Description

This exception is thrown when an alarm time is invalid.

Constructors

```
Ivi.LxiSync.AlarmTimeInvalidException(String alarmName,
                                      String alarmTime);

Ivi.LxiSync.AlarmTimeInvalidException();

Ivi.LxiSync.AlarmTimeInvalidException(String message);

Ivi.LxiSync.AlarmTimeInvalidException(String message,
                                      System.Exception innerException);
```

Message String

```
The alarm time is invalid.
Alarm name: <alarmName>.
Alarm time: <alarmTime>.
```

Parameters

Inputs	Description	Base Type
`alarmName`	The name of the alarm.	`String`
`alarmTime`	The invalid alarm time	`String`

Usage

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

11.1.4 CannotResolveReservedRepeatedCapabilityException

Description

This exception is thrown when a reserved repeated capability cannot be removed from one of the IviLxiSync collections.

Constructors

```
Ivi.LxiSync.CannotResolveReservedRepeatedCapabilityException(
                                    String repeatedCapability,
                                    String repeatedCapabilityInstance);

Ivi.LxiSync.CannotResolveReservedRepeatedCapabilityException();

Ivi.LxiSync.CannotResolveReservedRepeatedCapabilityException(
                                    String message);

Ivi.LxiSync.CannotResolveReservedRepeatedCapabilityException(
                                    String message,
                                    System.Exception innerException);
```

Message String

```
The reserved repeated capability cannot be removed from the collection.
Repeated capability name: <repeatedCapability>.
Repeated capability instance: <repeatedCapabilityInstance>.
```

Parameters

Inputs	Description	Base Type
`repeatedCapability`	The name of the repeated capability.	`String`
`repeatedCapabilityInstance`	The name of the repeated capability instance that cannot be removed from its collection.	`String`

Usage

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

11.1.5 EventSourceDoesNotExistException

Description

This exception is used when the driver finds that a specified event source is not defined.

Constructors

```
Ivi.LxiSync.EventSourceDoesNotExistException(String message,
                                             String eventSourceName);

Ivi.LxiSync.EventSourceDoesNotExistException();

Ivi.LxiSync.EventSourceDoesNotExistException(String message);

Ivi.LxiSync.EventSourceDoesNotExistException(String message,
                                             System.Exception innerException);
```

Message String

```
The specified event source has not been defined.
Event source name: <eventSourceName>
```

Parameters

Inputs	Description	Base Type
`eventSourceName`	The name of the event source that is undefined.	`String`

Usage

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

11.1.6 EventSourceExistsException

Description

This exception is used when the driver finds that a specified event source already exists.

Constructors

```
Ivi.LxiSync.EventSourceExistsException(String message,
                                       String eventSourceName);

Ivi.LxiSync.EventSourceExistsException();

Ivi.LxiSync.EventSourceExistsException(String message);

Ivi.LxiSync.EventSourceExistsException(String message,
                                       System.Exception innerException);
```

Message String

```
The specified event source already exists.
Event source name: <eventSourceName>
```

Parameters

Inputs	Description	Base Type
`eventSourceName`	The name of the event source that already exists.	`String`

Usage

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

11.1.7 EventSourceNotSetException

Description

This exception is used when the driver finds that the event source has not been specified.

Constructors

```
Ivi.LxiSync.EventSourceNotSetException();

Ivi.LxiSync.EventSourceNotSetException(String message);

Ivi.LxiSync.EventSourceNotSetException(String message,
                                       System.Exception innerException);
```

Message String

```
The event source has not been specified.
```

Usage

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

11.1.8 InvalidEventSourceException

Description

This exception is used when the driver finds that a specified event source is not valid.

Constructors

```
Ivi.LxiSync.InvalidEventSourceException(String message,
                                        String eventSourceName);

Ivi.LxiSync.InvalidEventSourceException();

Ivi.LxiSync.InvalidEventSourceException(String eventSourceName);

Ivi.LxiSync.InvalidEventSourceException(String eventSourceName,
                                        System.Exception innerException);
```

Message String

```
The specified event source is not valid.
Event source name: <eventSourceName>
```

Parameters

Inputs	Description	Base Type
`eventSourceName`	The name of the invalid event source.	`String`

Usage

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

11.1.9 OutOfEventResourcesException

Description

This exception is used when the driver is out of event resources.

Constructors

```
Ivi.LxiSync.OutOfEventResourcesException();

Ivi.LxiSync.OutOfEventResourcesException(String message);

Ivi.LxiSync.OutOfEventResourcesException(String message,
                                         System.Exception innerException);
```

Message String

```
Out of event resources.
```

Usage

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

11.1.10 WiredOrModeInvalidException

Description

This exception is used when the event source cannot operate in driven mode while serving as the wired-OR bias.

Constructors

```
Ivi.LxiSync.WiredOrModeInvalidException();

Ivi.LxiSync.WiredOrModeInvalidException(String message);

Ivi.LxiSync.WiredOrModeInvalidException(String message,
                                        System.Exception innerException);
```

Message String

```
The event source cannot operate in driven mode while serving as the wired-OR
bias.
```

Usage

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

12. Hierarchies

12.1 .NET Hierarchy

Table 12-1. .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Arm		
ArmCount	Arm Count	P
Delay	Arm Delay	P
Sources		
Add	Add Arm Source	M
DisableAll	Disable All Arm Sources	M
OrEnabled	Arm Source Or Enabled	P
Remove	Remove Arm Source	M
RemoveAllCustomArmSources	Remove All Custom Arm Sources	M
Count	Arm Source Count	P
[]	Item	
Configure	Configure Arm Source	M
Detection	Arm Source Detection	P
Enabled	Arm Source Enabled	P
EventID	Arm Source Event ID	P
Filter	Arm Source Filter	P
Name	Arm Source Name	P
Alarms		
Add	Add Arm Alarm	M
DisableAll	Disable All Arm Alarms	M
Remove	Remove Arm Alarm	M
RemoveAllArmAlarms	Remove All Arm Alarms	M
Count	Arm Alarm Count	P
[]	Item	
Configure	Configure Arm Alarm	M
Enabled	Arm Alarm Enabled	P
Name	Arm Alarm Name	P
Period	Arm Alarm Period	P
RepeatCount	Arm Alarm Repeat Count	P
Time	Arm Alarm Time	P
Trigger		
TriggerCount	Trigger Count	P
TriggerSource	Trigger Source	P

Table 12-1. .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Alarms		
Add	Add Trigger Alarm	M
DisableAll	Disable All Trigger Alarms	M
Remove	Remove Trigger Alarm	M
RemoveAllTriggerAlarms	Remove All Trigger Alarms	M
Count	Trigger Alarm Count	P
[]	Item	
Configure	Configure Trigger Alarm	M
Enabled	Trigger Alarm Enabled	P
Name	Trigger Alarm Name	P
Period	Trigger Alarm Period	P
RepeatCount	Trigger Alarm Repeat Count	P
Time	Trigger Alarm Time	P
Sources		
Add	Add Trigger Source	M
Remove	Remove Trigger Source	M
RemoveAllCustomTriggerSources	Remove All Custom Trigger Sources	M
Count	Trigger Source Count	P
[]	Item	
Configure	Configure Trigger Source	M
Delay	Trigger Source Delay	P
Detection	Trigger Source Detection	P
EventID	Trigger Source Event ID	P
Filter	Trigger Source Filter	P
Name	Trigger Source Name	P
Events		
Add	Add Event	M
DisableAll	Disable All Events	M
Remove	Remove Event	M
RemoveAllCustomEvents	Remove All Custom Events	M
Count	Event Count	P
WiredOrBiasMode	Wired OR Bias Mode	P
[]	Item	
Configure	Configure Event	M
DestinationPath	Event Destination Path	P
DriveMode	Event Drive Mode	P
Name	Event Name	P
Source	Event Source	P

Table 12-1. .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Slope	Event Slope	P
EventLog		
Clear	Clear Event Log	M
Enabled	Event Log Enabled	P
EntryCount	Event Log Entry Count	P
GetNextEntry	Get Next Event Log Entry	M
Time		
SystemTime	System Time	M
IsMaster	Is Master	P
IsSynchronized	Is Synchronized	P

12.1.1 IviLxiSync .NET Interfaces

IviLxiSync-interfaces contain interface reference properties for accessing the following IviLxiSync interfaces:

- IIviLixSyncArm
- IIviLixSyncEventLog
- IIviLixSyncEvents
- IIviLixSyncTime
- IIviLixSyncTrigger

The IIviLxiSyncArm interface contains interface reference properties for accessing additional the following IviLxiSync interfaces:

- IIviLxiSyncArmAlarms
- IIviLxiSyncArmSources

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

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

The IIviLxiSyncTrigger interface contains interface reference properties for accessing additional the following IviLxiSync interfaces:

- IIviLxiSyncTriggerAlarms
- IIviLxiSyncTriggerSources

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

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

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

12.1.2 Interface Reference Properties

Interface reference properties are used to navigate the IviLxiSync .NET hierarchy. This section describes the interface reference properties that the IIviLxiSync, IIviLxiSyncArm, IIviLxiSyncArmAlarms, IIviLxiSyncArmSources, IIviLxiSyncTriggerAlarms, IIviLxiSyncTriggerSources, and IIviLxiSyncEvents interfaces define. All interface reference properties are read-only.

Table 12-2. Interface Referece Properties

Data Type	Access
IIviLxiSyncArm	Arm
IIviLxiSyncArmAlarm	Alarms[]
IIviLxiSyncArmAlarms	Alarms
IIviLxiSyncArmSource	Sources[]
IIviLxiSyncArmSources	Sources
IIviLxiSyncEvent	Events[]
IIviLxiSyncEventLog	EventLog
IIviLxiSyncEvents	Events
IIviLxiSyncTime	Time
IIviLxiSyncTrigger	Trigger
IIviLxiSyncTriggerAlarm	Alarms[]
IIviLxiSyncTriggerAlarms	Alarms
IIviLxiSyncTriggerSource	Sources[]
IIviLxiSyncTriggerSources	Sources

12.2 COM Hierarchy

Table 12-3. COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Arm		
ArmCount	Arm Count	P
Delay	Arm Delay	P
Sources		
Add	Add Arm Source	M
DisableAll	Disable All Arm Sources	M
OrEnabled	Arm Source Or Enabled	P
Remove	Remove Arm Source	M
RemoveAllCustomArmSources	Remove All Custom Arm Sources	M
Count	Arm Source Count	P
Name	Arm Source Name	P

Table 12-3. COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Item		
Configure	Configure Arm Source	M
Detection	Arm Source Detection	P
Enabled	Arm Source Enabled	P
Event ID	Arm Source Event ID	P
Filter	Arm Source Filter	P
Alarms		
Add	Add Arm Alarm	M
DisableAll	Disable All Arm Alarms	M
Remove	Remove Arm Alarm	M
RemoveAllArmAlarms	Remove All Arm Alarms	M
Count	Arm Alarm Count	P
Name	Arm Alarm Name	P
Item		
Configure	Configure Arm Alarm	M
Enabled	Arm Alarm Enabled	P
Period	Arm Alarm Period	P
RepeatCount	Arm Alarm Repeat Count	P
TimeSeconds	Arm Alarm Time Seconds	P
TimeFraction	Arm Alarm Time Fraction	P
Trigger		
TriggerCount	Trigger Count	P
TriggerSource	Trigger Source	P
Alarms		
Add	Add Trigger Alarm	M
DisableAll	Disable All Trigger Alarms	M
Remove	Remove Trigger Alarm	M
RemoveAllTriggerAlarms	Remove All Trigger Alarms	M
Count	Trigger Alarm Count	P
Name	Trigger Alarm Name	P
Item		
Configure	Configure Trigger Alarm	M
Enabled	Trigger Alarm Enabled	P
Period	Trigger Alarm Period	P
RepeatCount	Trigger Alarm Repeat Count	P
TimeSeconds	Trigger Alarm Time Seconds	P
TimeFraction	Trigger Alarm Time Fraction	P
Sources		

Table 12-3. COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Add	Add Trigger Source	M
Remove	Remove Trigger Source	M
RemoveAllCustomTriggerSources	Remove All Custom Trigger Sources	M
Count	Trigger Source Count	P
Name	Trigger Source Name	P
Item		
Configure	Configure Trigger Source	M
Delay	Trigger Source Delay	P
Detection	Trigger Source Detection	P
Event ID	Trigger Source Event ID	P
Filter	Trigger Source Filter	P
Events		
Add	Add Event	M
DisableAll	Disable All Events	M
Remove	Remove Event	M
RemoveAllCustomEvents	Remove All Custom Events	M
Count	Event Count	P
Name	Event Name	P
WiredOrBiasMode	Wired OR Bias Mode	P
Item		
Configure	Configure Event	M
DestinationPath	Event Destination Path	P
Drive Mode	Event Drive Mode	P
Source	Event Source	P
Slope	Event Slope	P
EventLog		
Clear	Clear Event Log	M
Enabled	Event Log Enabled	P
EntryCount	Event Log Entry Count	P
GetNextEntry	Get Next Event Log Entry	M
Time		
GetSystemTime	Get System Time	M
IsMaster	Is Master	P
IsSynchronized	Is Synchronized	P

12.2.1 IviLxiSync COM Interfaces

IviLxiSync-interfaces contain interface reference properties for accessing the following IviLxiSync interfaces:

- IIviLixSyncArm
- IIviLixSyncEventLog
- IIviLixSyncEvents
- IIviLixSyncTime
- IIviLixSyncTrigger

The IIviLxiSyncArm interface contains interface reference properties for accessing additional the following IviLxiSync interfaces:

- IIviLxiSyncArmAlarms
- IIviLxiSyncArmSources

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

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

The IIviLxiSyncTrigger interface contains interface reference properties for accessing additional the following IviLxiSync interfaces:

- IIviLxiSyncTriggerAlarms
- IIviLxiSyncTriggerSources

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

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

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

12.2.2 COM Interfaces

Table 12-4. Interface GUIDs lists the interfaces that this specification defines and their GUIDs.

Table 12-4. Interface GUIDs

Interface	GUID
IIviLxiSync	47ed5346-a398-11d4-ba58-000064657374
IIviLxiSyncArm	47ed5347-a398-11d4-ba58-000064657374
IIviLxiSyncArmAlarm	47ed5348-a398-11d4-ba58-000064657374
IIviLxiSyncArmAlarms	47ed5349-a398-11d4-ba58-000064657374
IIviLxiSyncArmSource	47ed534a-a398-11d4-ba58-000064657374
IIviLxiSyncArmSources	47ed534b-a398-11d4-ba58-000064657374
IIviLxiSyncEvent	47ed534c-a398-11d4-ba58-000064657374
IIviLxiSyncEventLog	47ed534d-a398-11d4-ba58-000064657374

Table 12-4. Interface GUIDs

Interface	GUID
IIviLxiSyncEvents	47ed534e-a398-11d4-ba58-000064657374
IIviLxiSyncTime	47ed534f-a398-11d4-ba58-000064657374
IIviLxiSyncTrigger	47ed5350-a398-11d4-ba58-000064657374
IIviLxiSyncTriggerAlarm	47ed5351-a398-11d4-ba58-000064657374
IIviLxiSyncTriggerAlarms	47ed5352-a398-11d4-ba58-000064657374
IIviLxiSyncTriggerSource	47ed5353-a398-11d4-ba58-000064657374
IIviLxiSyncTriggerSources	47ed5354-a398-11d4-ba58-000064657374

12.2.3 COM Interface Reference Properties

Interface reference properties are used to navigate the IviLxiSync COM hierarchy. This section describes the interface reference properties that the IIviLxiSync, IIviLxiSyncArm, IIviLxiSyncTrigger, IIviLxiSyncEvents, and IIviLxiSyncEventLog.

Table 12-5. Interface Reference Properties

DataType	Access
IIviLxiSyncArm	Arm
IIviLxiSyncArmAlarm	Alarm.Item()
IIviLxiSyncArmAlarms	Alarms
IIviLxiSyncArmSource	Sources.Item()
IIviLxiSyncArmSources	Sources
IIviLxiSyncEvent	Events.Item()
IIviLxiSyncEventLog	EventLog
IIviLxiSyncEvents	Events
IIviLxiSyncTime	Time
IIviLxiSyncTrigger	Trigger
IIviLxiSyncTriggerAlarm	Alarms.Item()
IIviLxiSyncTriggerAlarms	Alarms
IIviLxiSyncTriggerSource	Sources.Item()
IIviLxiSyncTriggerSources	Sources

12.2.4 COM Category

The IviLxiSync COM Category shall be “IviLxiSync”, and the Category ID (CATID) shall be {47ed515c-a398-11d4-ba58-000064657374}.

12.2.5 COM Interface Accessibility

When an IviLxiSync-compliant IVI-COM driver is instantiated, a reference to the main driver class is returned. A call to QueryInterface on this main class shall succeed for all IviLxiSync interfaces, except for interfaces that implement repeated capabilities as collections.

See *IVI-3.1: Driver Architecture Specification* for details on the relationship between IVI-COM driver classes, interfaces, and the QueryInterface function.

12.3 C Function Hierarchy

The C function hierarchy is shown in the following table.

Name or Class	Function Name
Configuration…	
Arm…	
Alarm…	
Add Arm Alarm	`IviLxiSync_AddArmAlarm`
Configure Arm Alarm	`IviLxiSync_ConfigureArmAlarm`
Disable All Arm Alarms	`IviLxiSync_DisableAllArmAlarms`
Get Arm Alarm Name	`IviLxiSync_GetArmAlarmName`
Remove All Custom Arm Alarms	`IviLxiSync_RemoveAllCustomArmAlarms`
Remove Arm Alarm	`IviLxiSync_RemoveArmAlarm`
Source…	
Add Arm Source	`IviLxiSync_AddArmSource`
Configure Arm Source	`IviLxiSync_ConfigureArmSource`
Disable All Arm Sources	`IviLxiSync_DisableAllArmSources`
Get Arm Source Name	`IviLxiSync_GetArmSourceName`
Remove All Custom Arm Sources	`IviLxiSync_RemoveAllCustomArmSources`
Remove Arm Source	`IviLxiSync_RemoveArmSource`
Trigger…	
Alarm…	
Add Trigger Alarm	`IviLxiSync_AddTriggerAlarm`
Configure Trigger Alarm	`IviLxiSync_ConfigureTriggerAlarm`
Disable All Trigger Alarms	`IviLxiSync_DisableAllTriggerAlarms`
Get Trigger Alarm Name	`IviLxiSync_GetTriggerAlarmName`
Remove All Trigger Alarms	`IviLxiSync_RemoveAllTriggerAlarms`
Source…	
Add Trigger Source	`IviLxiSync_AddTriggerSource`
Configure Trigger Source	`IviLxiSync_ConfigureTriggerSource`
Get Trigger Source Name	`IviLxiSync_GetTriggerSourceName`
Remove All Custom Trigger Sources	`IviLxiSync_RemoveAllCustomTriggerSources`
Remove Trigger Source	`IviLxiSync_RemoveTriggerSource`

Name or Class	Function Name
Event…	
Add Event	IviLxiSync_AddEvent
Configure Event	IviLxiSync_ConfigureEvent
Disable All Events	IviLxiSync_DisableAllEvents
Get Event Name	IviLxiSync_GetEventName
Remove All Custom Events	IviLxiSync_RemoveAllCustomEvents
Remove Event	IviLxiSync_RemoveEvent
Event Log…	
Clear Event Log	IviLxiSync_ClearEventLog
Get Next Event Log Entry	IviLxiSync_GetNextEventLogEntry
Time…	
Get System Time	IviLxiSync_GetSystemTime

12.4 C Attribute Hierarchy

The IviLxiSync attribute hierarchy is shown in the following table.

Table 12-3. C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Arm	
Arm Count	IVILXISYNC_ATTR_ARM_COUNT
Arm Delay	IVILXISYNC_ATTR_ARM_DELAY
Alarm	
Arm Alarm Count	IVILXISYNC_ATTR_ARM_ALARM_COUNT
Arm Alarm Enabled	IVILXISYNC_ATTR_ARM_ALARM_ENABLED
Arm Alarm Period	IVILXISYNC_ATTR_ARM_ALARM_PERIOD
Arm Alarm Repeat Count	IVILXISYNC_ATTR_ARM_ALARM_REPEAT_COUNT
Arm Alarm Time Seconds	IVILXISYNC_ATTR_ARM_ALARM_TIME_SECONDS
Arm Alarm Time Fraction	IVILXISYNC_ATTR_ARM_ALARM_TIME_FRACTION
Source	
Arm Source Count	IVILXISYNC_ATTR_ARM_SOURCE_COUNT
Arm Source Detection	IVILXISYNC_ATTR_ARM_SOURCE_DETECTION
Arm Source Enabled	IVILXISYNC_ATTR_ARM_SOURCE_ENABLED
Arm Source Or Enabled	IVILXISYNC_ATTR_ARM_SOURCE_OR_ENABLED
Arm Source Event ID	IVILXISYNC_ATTR_ARM_SOURCE_EVENTID
Arm Source Filter	IVILXISYNC_ATTR_ARM_SOURCE_FILTER
Trigger	

Table 12-3. C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Trigger Count	IVILXISYNC_ATTR_TRIGGER_COUNT
Trigger Source	IVILXISYNC_ATTR_TRIGGER_SOURCE
Alarm	
Trigger Alarm Count	IVILXISYNC_ATTR_TRIGGER_ALARM_COUNT
Trigger Alarm Enabled	IVILXISYNC_ATTR_TRIGGER_ALARM_ENABLED
Trigger Alarm Period	IVILXISYNC_ATTR_TRIGGER_ALARM_PERIOD
Trigger Alarm Repeat Count	IVILXISYNC_ATTR_TRIGGER_ALARM_REPEAT_COUNT
Trigger Alarm Time Seconds	IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_SECONDS
Trigger Alarm Time Fraction	IVILXISYNC_ATTR_TRIGGER_ALARM_TIME_FRACTION
Source	
Trigger Source Count	IVILXISYNC_ATTR_TRIGGER_SOURCE_COUNT
Trigger Source Delay	IVILXISYNC_ATTR_TRIGGER_SOURCE_DELAY
Trigger Source Detection	IVILXISYNC_ATTR_TRIGGER_SOURCE_DETECTION
Trigger Source Event ID	IVILXISYNC_ATTR_TRIGGER_SOURCE_EVENTID
Trigger Source Filter	IVILXISYNC_ATTR_TRIGGER_SOURCE_FILTER
Event	
Event Count	IVILXISYNC_ATTR_EVENT_COUNT
Event Wired OR Bias Mode	IVILXISYNC_ATTR_EVENT_WIRED_OR_BIAS_MODE
Event Destination Path	IVILXISYNC_ATTR_EVENT_DESTINATION_PATH
Event Drive Mode	IVILXISYNC_ATTR_EVENT_DRIVE_MODE
Event Slope	IVILXISYNC_ATTR_EVENT_SLOPE
Event Source	IVILXISYNC_ATTR_EVENT_SOURCE
Event Log	
Event Log Enabled	IVILXISYNC_ATTR_EVENT_LOG_ENABLED
Event Log Entry Count	IVILXISYNC_ATTR_EVENT_LOG_ENTRY_COUNT
Time	
Is Time Master	IVILXISYNC_ATTR_IS_TIME_MASTER
Is Time Synchronized	IVILXISYNC_ATTR_IS_TIME_SYNCHRONIZED