

C#	
	<u> </u>

[This is preliminary documentation and is subject to change.]

Welcome to Lecroy Aribtrary Waveform Generator Series control library. A series of object and classes to control ArbStudio instruments. As such, consider this work in progress. If you have any information, tips, updates, or corrections that you would like to see added to the guide, feel free to submit them to the author using the Send Feedback to the e-mail link in the page footer.

Brief Description

AWG4000Control.dll contains a list of classes created in C# to control any Arbitrary Waveform Generator Series instruments like **ArbStudio 1104** or **ArbStudio 1102**.

In the next chapter a flow diagram explain how a ArbStudio instruments is represented within the library and wich classes are involved to control it.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



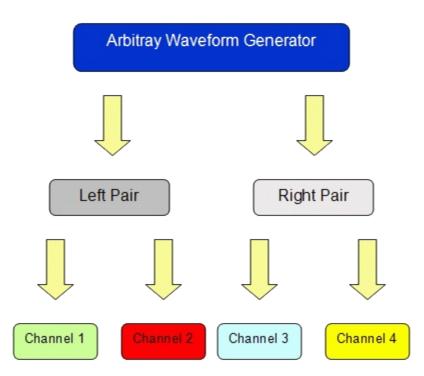
C#

[This is preliminary documentation and is subject to change.]

AWG4000Control contains a list of classes to control ArbStudio instruments. control the device all classes reflects the following diagrams

Class Diagram

The following diagram explain how the ArbStudio device is represented.



Class Representation

Each device level in the diagram is represented as:

- Arbitrary Waveform Generator: Class Device
- Left or Right Pair: Class PairChannels
- Channel 1..4: Class Channels

✓ Note:

For ArbStudio 1102 Right Pair is not defined

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



C#

Cop'

[This is preliminary documentation and is subject to change.]

Class Device can be created only getting item from class DeviceSet. With creation of a new class DeviceSet the low-level driver perform a search of all ArbStudio usb connected devices. Property DeviceList of class DeviceSet return a list of Device classes.

Create device

- 1. Create a new class DeviceSet. During initialization this class search for all connected devices.
- 2. From property DeviceList of class DeviceSet get the device that will be controlled.

Example

C#

```
DeviceSet deviceSet = new DeviceSet();
//Check if any errors
ATError result = deviceSet.ErrorResult;
//Display error if present
if (result.ErrorSource != ErrorCodes.RES SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);
//Get Device
Device device = deviceSet.DeviceList[0];
```

See Also

ActiveTechnologies.Instruments.AWG4000.Control.DeviceSet ActiveTechnologies.Instruments.AWG4000.Control.Device

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved

C#

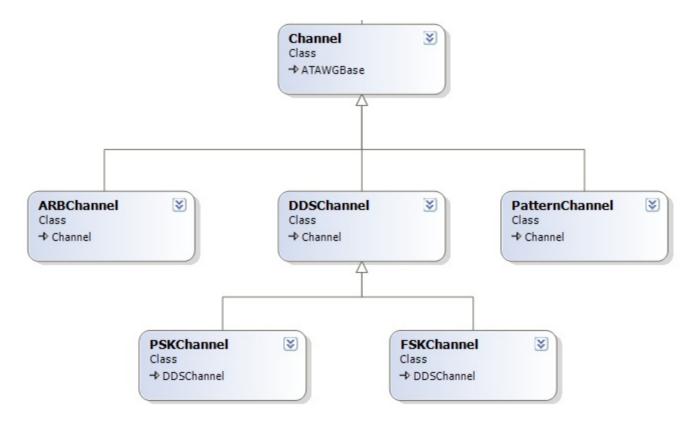
[This is preliminary documentation and is subject to change.]

Once channels list is created on device, each Channel class expose methods upload waveforms data and properties to set channel configuration.

For available methods and properties please refer to base class Channel and its derived class

Channel Class Diagram

The following diagram explain Channel class and derived classes.



■ Create Channels

- 1. To create channels list for device simply construct an array of 4 (2 for ArbStudio 1102) elements of Functionality type.
- 2. Call method Initialize of class Device the instruments will be prepare for the given channels list type.

Example

C# Cop⁺

```
Functionality[] funct = new Functionality[4];
funct[0] = Functionality.ARB;
funct[1] = Functionality.ARB;
funct[2] = Functionality.FSK;
funct[3] = Functionality.PSK;
ATError result = device.Initialize(funct);
```

■ Get Channels from Device

• Once channels are successfully created on Device class you need to get Channel class from Device to upload settings and waveforms data. To get Channel use GetChannel method of Device class giving the channel identification (0..3).

All specific channels derived from base class Channel. Specific channels are represented by:

- ARBChannel for arbitrary channel definition
- FSKChannel for DDS channel in Frequency modulated channel
- PSKChannel for DDS channel in Phase modulated channel

Example

```
C# Copy
```

```
ARBChannel channelArb1 = (ARBChannel)device.GetChannel(0);
ARBChannel channelArb2 = (ARBChannel)device.GetChannel(1);
FSKChannel channelFSK = (FSKChannel)device.GetChannel(2);
PSKChannel channelPSK = (FSKChannel)device.GetChannel(3);
```

See Also

ActiveTechnologies.Instruments.AWG4000.Control.Device

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Working with

Arbitrary Channel

C#	~ 1
C 11	

[This is preliminary documentation and is subject to change.]

Arbitrary channel can be used to create a signal for standard waveforms or fc amplitude modulation.

For a complete list of available method and properties please refer to base class Channel and its derived ARBChannel class.

Loading waveforms to an Arbitrary Channel

• To upload waveforms data to arbitrary channel use LoadWaveforms method.

WaveformStruct[] parameter is array of waveforms definition.

Each element WaveformStruct is composed by array of samples that design the waveform and array of markers.

✓ Note:

Marker is the ordinal number of sample in the samples array. For example if marker is located to 100th sample marker value must be set to 100

Example

C#

Cop'

Loading generation sequence to an Arbitrary Channel

 After waveforms definition is loaded on channel a sequence of how waveforms are generated must be defined.

To load generation sequence use LoadGenerationSequence method.

GenerationSequenceStruct[] parameter is array of waveform
identification with number of repetitions

mode parameter is method of transfer data to device. Use ReEntrant TransferMode if you want to upload dinamically sequence.

ClearMemory parameter specify if memory must be cleared before uploanew data. This parameter is set to false if in single TriggerMode you wanto upload generation sequence dinamically.

Example

```
GenerationSequenceStruct[] sequence = new GenerationSequenceStruct[1];
ARBChannel channelArb1 = (ARBChannel)device.GetChannel(0);

//Construct generation sequence. WaveformIndex is 0-based
sequence[0].WaveformIndex = 0;
sequence[0].Repetitions = 1;

//Load generation sequence to channel
ATError result = channelArb1.LoadGenerationSequence(sequence, TransferNot)

//Display error if present
if (result.ErrorSource != ErrorCodes.RES SUCCESS)
```

Console.WriteLine("Error:" + result.ErrorDescription);

See Also

ActiveTechnologies.Instruments.AWG4000.Control.ARBChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

41



ArbStudio Control Classes Library Documentation Working with DDS

Channel

C#		-
	4	_

[This is preliminary documentation and is subject to change.]

DDS channel can be used to create frequency modulated or phase modulated signal.

For frequency modulated signal use FSKChannel channel class derived from DDSChannel base class.

For phase modulated signal use PSKChannel channel class derived from DDSChannel base class.

For a complete list of available method and properties please refer to base class Channel and its derived.

■ Loading waveforms to a DDS (FSK or PSK) Channel

- To upload waveform (carrier) data to a DDS channel use LoadCarrier method.
 - WaveformStruct parameter is structure with waveform definition.

WaveformStruct is composed by array of samples that design the waveform and one marker.

✓ Note:

Unlike Arbitrary channel marker is represented by amplitude value in Volt

Example



```
WaveformStruct wave = new WaveformStruct[1];
FSKChannel channelFSK = (FSKChannel)device.GetChannel(2);
//construct array of samples and markers
double[] samples = new double[10];
```

```
double[] marker = new double[1];
for (int i = 0; i < 10; i++)
    samples[i] = i;
marker[0] = 2.5F;

//set WaveformStruct
wave[0].Sample = samples;
wave[0].Marker = marker;

//Load waveforms to channel
ATError result = channelFSK.LoadCarrier(waves);

//Display error if present
if (result.ErrorSource != ErrorCodes.RES_SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);</pre>
```

■ Load FSK modulation law to a DDS FSK Channel

- After waveforms carrier is loaded on channel a modulation law must be defined. To load modulation law use LoadFSKModulationLaw method.
 - FrequencyModulationLawStruct[] parameter is array of structure with sample value and number of repetitions for interval to be maintained.
 - mode parameter is method of transfer data to device
 - ClearMemory parameter specify if memory must be cleared before upload new data. This parameter is set to false if in single TriggerMode you want to upload modulation law dinamically.

✓ Note:

Maximum value for number of repetitions for time interval is $(2^20)-1$

Example

```
Cop¹
```

```
FrequencyModulationLawStruct[] modulation = new FrequencyModulationLawS
FSKChannel channelFSK = (FSKChannel)device.GetChannel(2);
double[] samples = new double[1];
double[] duration = new double[1];
```

Load amplitude profile to a DDS Channel

 Amplitude Profile tool allows setting the amplitude profile of the generate signal as a function of its frequency.

To load amplitude profile use SetAmplitudeProfile method.

AmplitudeProfileStruct structure define all profile parameters.
 Refer to SetAmplitudeProfile in DDSChannel topic for detailed information.

Example

```
AmplitudeProfileStruct ampProfile = new AmplitudeProfileStruct;
FSKChannel channelFSK = (FSKChannel)device.GetChannel(2);
double[] samples = new double[256];

for (int i = 0; i < 256; i++)
    samples[i] = 1;

//Set amplitude profile struct
ampProfile.AmplitudeInf = 1;
ampProfile.AmplitudeSup = 1;
ampProfile.FrequencyInf = 0;
amplitudeStruct.Samples = samples;
amplitudeStruct.BandWidth = 0;

//Load generation sequence to channel</pre>
```

```
ATError result = channelFSK.SetAmplitudeProfile(ampProfile);

//Display error if present
if (result.ErrorSource != ErrorCodes.RES_SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);
```

See Also

ActiveTechnologies.Instruments.AWG4000.Control.DDSChannel ActiveTechnologies.Instruments.AWG4000.Control.FSKChannel ActiveTechnologies.Instruments.AWG4000.Control.PSKChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

41



C#

[This is preliminary documentation and is subject to change.]

Once channels list is created on device and waveform definitions are uploaded it's possibile to start signals generation. Methods RUN and STOP of class Device instruct instrument to generate and terminate this.

For available methods and properties please refer to class Device.

Start and Stop Generation

1. Prepare an array of byte with list of channels number to be started or stopped

Id	Pair	Channel
1	Left	1
2	Left	2
3	Right	3
4	Right	4

✓ Note:

For ArbStudio 1102 only channel id's 1 and 2 are available.

2. Call the RUN method of the Device class giving the array of channels to start channels.

Call the STOP method of the Device class to stop instrument.

Example



```
ATError result = null;
byte[] channels = new byte[4];
channels[0] = 1;
channels[1] = 2;
channels[2] = 3;
```

```
channels[3] = 4;
//Start channels
result = device.RUN(channels);
//Display error if present
if (result.ErrorSource != ErrorCodes.RES SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);
//Stop channels
result = device.STOP();
//Display error if present
if (result.ErrorSource != ErrorCodes.RES SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);
//Start channel 2
channels = new byte[1];
channels[\emptyset] = 2;
result = device.RUN(channels);
if (result.ErrorSource != ErrorCodes.RES SUCCESS)
    Console.WriteLine("Error:" + result.ErrorDescription);
```

■ Robust Programming

Please check always ATError class that contain the result of last operation. RUN or STOP channels may be generate error due for not initialized device or channel in busy status. Use ATError.ErrorDescription to obtain a brief description of occured error. Refer to ErrorCodes class for a complete list of errors.

See Also

ActiveTechnologies.Instruments.AWG4000.Control.Device ActiveTechnologies.Instruments.AWG4000.Control.ATError ActiveTechnologies.Instruments.AWG4000.Control.ErrorCodes

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



C#	- 1
U 11	

[This is preliminary documentation and is subject to change.]

ArbStudio can be configured to work as a powerful Digital Pattern Generator-Sampler (optional). In this working mode ArbStudio provides the capability to emulate standard serial or parallel bus transitions or custom digital interfaces for system debugging and characterization.

Its architecture is based on a vector/command memory and a powerful sequencer that defines the vector/command execution flow.

The sequencer unit is a programmable finite state machine with a microcode memory that, combined with the vector/command memory, can generate verlong digital patters and control signals at a very high speed: up to 125 MS/s.

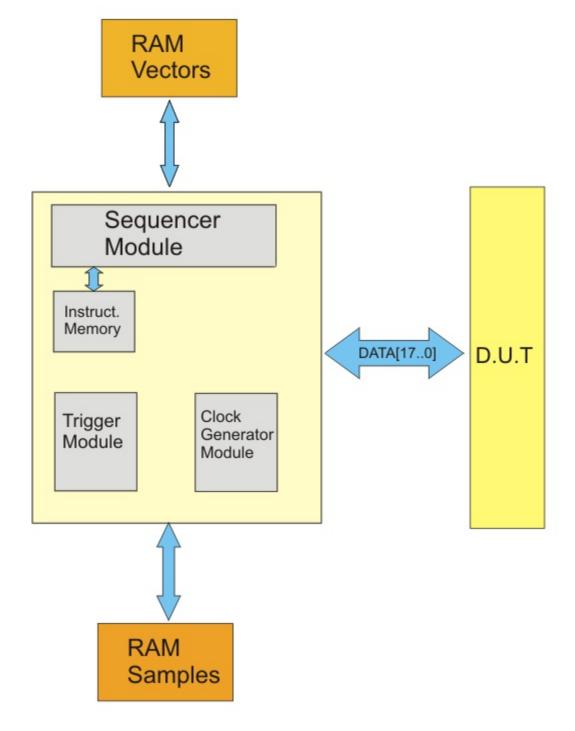
For a complete list of available method and properties please refer to base class PatternGenerator.

Please refer to Digital Pattern Generator-Sampler SDK example application for a correct usage of methods, properties and modules descripted below.

Please refer to ArbStudio 1104/1102 User Manual for in depth explanation about Digital Pattern Generator modules and instrument usage.

Digital Pattern Generator Architecture

The following diagram explains Digital Pattern Generator architecture and modules.



■ Clock Generator Module

- Using the Clock Generator module, the user can generate a clock signal with a frequency up to 125 MHz.
- The Ch15 is the output channel for the generated clock
- It is possible to logically combine the clock signal with the ExtendedTriggerIN (the VECTOR(16)) using the strobe function:NOP,OR,AND,XOR,NOT, NOR, NAND, XNOR are available.
- It provides a sync signal to the Sequencer Module to generate vectors syncronized with the output clock.

• The duty cycle is adjustable

Sequencer Module

- The sequencer instructions allow to:
- Repeat vectors stored inside the vector memory at each of several addresses
- Make conditional and unconditional jumps
- Monitor input signals combining them with a mask
- Modify the mask on input signals
- Insert wait states
- Wait for trigger events
- Reset Trigger Events

Trigger Module

- Using the Trigger Module, the user can set a trigger condition on Ch (17..0) and on Vector(16) to generate a trigger event like a logic analyze
- **Conditions on edges:** for each signal it is possible to indicate whether the trigger condition must be detected on a rising edge, on a falling edge or for any of the two edges. When the trigger conditions have been set or edges of several signals, the trigger event will be determined by the occurrence of at least one condition. If no trigger conditions on edges hav been set for any signals, trigger events on edges will never occur.
- **Conditions on logic levels:** for each signal it is possible to indicate whether the trigger condition must be detected on a high or a low logic level. When the trigger conditions have been set on logic levels of severa signals, the trigger event will be determined by the occurrence of all conditions simultaneously. If no trigger conditions on logic levels have been set for any signal, the trigger conditions on logic levels are always considered as verified. The trigger condition set on logic levels is useful when the trigger event must be detected for a specific pattern.
- Relationship between trigger conditions A relationship between

trigger conditions on edges and trigger conditions on logic levels can be s by the user, by using the Trigger condition menu ring. Several options are available: Edge OR Level, Edge AND Level, Edge BEFORE Level, Edge AFTER Level, Always Trigger, Never Trigger

• **Trigger Levels** Digital Pattern Generator-Sampler allows editing up to 3: trigger levels. Each trigger level has 2 trigger conditions "A" and "B". Each condition can be set with Action parameter: None the happening condition will be ignored (Never Trigger); Trigger: the condition happening leads to Trigger event; Jump to Leveln: the condition happening leads to a jump a trigger Level n, where n is the number of one of the present trigger levels.

■ Memory Modules

- **Instruction Memory**: it is the memory that stores the Sequencer instructions (Maximum number of available instructions is 1024). Memory Depth is 1024.
- **SRAM Vectors**: it is the memory that stores the vectors to generate (Maximum number of vectors is 1048576). Memory Depth is 1M.
- **SRAM Samples**: it is the memory that stores the samples acquired from the D.U.T.Memory Depth is 1M.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation Active Technologies. Instruments. AV

Namespace

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Co

[This is preliminary documentation and is subject to change.]

ArbStudio Control Classes Library contains all object to manage LeCroy Arbitrary Waveform Generator Series ArbStudio 1104 and 1102

Declaration Syntax

C#	Visual Basic	Visual C++
namespace ActiveTe	chnologies.Instrume	nts.AWG4000.Control
Namespace ActiveTe	chnologies.Instrume	nts.AWG4000.Control
namespace ActiveTe	chnologies.Instrume	nts.AWG4000.Control

Types

, ,					
All Ty	pes	Classes	Str	ructures	Enumerations
Icon	Туре			Description	
	ActionType			Pattern generated action type. Non condition will be Trigger); Trigger: happening leads Jump to Leveln: happening leads trigger Level n, wonumber of one of trigger levels.	e the happening ignored (Never the condition to a Trigger event the condition to a jump at where n is the
\$	<u>Amplitude</u> M	<u>IodulationLawStru</u>	<u>ct</u>	Struct of amplitu	ıde modulation law

	<u>AmplitudeProfileStruct</u>	Struct for amplitude profile.
₃ 2	<u>ARBBufferType</u>	Arbitrary channel clear memory buffer type.
9 \$	ARBChannel	Class to manage an arbitrary channel on Arbitrary Waveform Generator device.
9 \$	ATAWGBase	Provide a base class for AWG4000Control classes
4 \$	ATError	Class for error handling
₃ 9	ATXSSEvent	Line source for TriggerOUT channe trigger and handled event signal.
9 \$	AWG4000_PatternRegsStruct	Struct-Class for ArbStudio_Pattern Registers
4 \$	Channel	Base class to manage Arbitrary or DDS channel on Arbitrary Waveform Generator device.
	<u>ChannelOutLogicOperation</u>	Trigger combination between channels and pairs
	<u>ChannelStatus</u>	Status of channel

9 \$	ClockGenStruct	Pattern generator struct for clock generator.
	ClockSource	Clock source for channel.
(F)	ClockStrobeOpType	Pattern generators sets parameters for the strobe operation of the Clock Generator Module.
9 \$	Conversions	Class to manage conversion between data and instruments data
2	DDSBufferType	DDS channel clear memory buffer type.
%	DDSChannel	Base class to manage an phase modulated or frequency modulated DDS channel on Arbitrary Waveform Generator device.
%	Device	Class to manage an Arbitrary Waveform Generator device.
%	DeviceSet	Provide a class to manage all AWG4000 usb connectable devices
%	ErrorCodes	Provide a class with AWG4000Control errod codes.
	FrequencyInterpolation	Frequency interpolation for DAC.

%	<u>FrequencyModulationLawStruct</u>	Struct of frequency modulation law
₹ \$	<u>FrequencyRange</u>	Frequency range for flatness
4\$	<u>FSKChannel</u>	Class to manage an phase frequency modulated DDS channel on Arbitrary Waveform Generator device.
	<u>Functionality</u>	Channel functionality: Arbitrary, PSK, FSK, PatternGen. Other values are for internal use.
⋄	GenerationSequenceStruct	Struct for waveform generation sequence
₽ ţ	Instrument	Instrument Utilities Class
23	<u>InstrumentType</u>	Type of USB connected device.
₂ 3	<u>MemorySize</u>	Amount of installed memory
	<u>ModulationTriggerMode</u>	Trigger mode for arbitrary waveform modulation.
3	<u>ModulationType</u>	Modulation type of channel.

	<u>OutputImpedance</u>	Output channel impedance.
	<u>PairChannelAddSubsOperation</u>	Specify sign in add operation between channels.
3 3	<u>PairChannelDirection</u>	Link type between pair channels.
8	<u>PairChannelMathOperation</u>	Math operation type for pair channel.
%	<u>PairChannels</u>	Class to manage an pair of channels (left: channels 1, 2 or right: channels 3, 4) on Arbitrary Waveform Generator device.
	Pattern_ControlRegisterType	
	Pattern_SC_Connect_Pin	Expansion Bus Pin Connect
	Pattern_SelectionRegisterType	Parameter for Selection Register
9 \$	<u>PatternGenerator</u>	Class to manage a Digital Pattern Generator-Sampler device
% >	<u>PhaseModulationLawStruct</u>	Struct of phase modulation law.
₂ 2	<u>PLLStatus</u>	Status of PLL.

%	<u>PSKChannel</u>	Class to manage an phase modulated DDS channel on Arbitrary Waveform Generator device.
%	ReEntrantQueueStruct	Struct for ReEntrant bulk transfers
₂ 3	<u>RunningState</u>	Run state of instrument
a	<u>SensitivityEdge</u>	Sensitive edge for the IN/OUT trigger.
9 \$	<u>SEQInstructionCodeType</u>	Pattern generator struct for Sequencer.
3	<u>SequencerLoopType</u>	Pattern generator sequencer loop type
3	<u>SequencerOpcodeType</u>	Pattern generator sequencer instruction.
3	Setup_LevelCondType	Pattern generator indicate whether the trigger condition must be detected on a high or a low logic level.
3	Setup_TriggerCondType	Pattern generator relationship between trigger conditions on edges and trigger conditions on logic levels.

<u>SuperCableStruct</u>	Pattern generator class for Expansion Bus
<u>TransferMode</u>	Bulk transfer mode to USB.
<u>TransferStatus</u>	Status of bulk tranfers to the channel.
<u>TransferTypeDef</u>	Specify type of bulk transfer.
<u>TriggerAction</u>	Channel action for trigger operation.
<u>TriggerConditionType</u>	Pattern generator trigger condition type.
<u>TriggerMode</u>	Trigger mode in waveform sequence.
<u>TriggerSource</u>	Line source for TriggerOUT channe trigger and handled event signal.
<u>TriggerStruct</u>	Pattern generator struct for trigger
WaveformStruct	Struct of Waveform reprepresentation.
	TransferMode TransferStatus TransferTypeDef TriggerAction TriggerConditionType TriggerMode TriggerSource TriggerStruct

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Action Type

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

ActionType

 C#	- 1
U 11	4

[This is preliminary documentation and is subject to change.]

Pattern generator levels trigger action type. None the happening condition wi be ignored (Never Trigger); Trigger: the condition happening leads to a Trigger event; Jump to Leveln: the condition happening leads to a jump at trigger Level n, where n is the number of one of the present trigger levels.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum ActionType			
Public Enumeration ActionType			
public enum class	ActionType		

Members

Member	Description
None	
Trigger	
JumpToLevel0	
JumpToLevel1	
JumpToLevel2	
JumpToLevel3	
JumpToLevel4	
JumpToLevel5	
JumpToLevel6	
JumpToLevel7	

JumpToLevel8	
JumpToLevel9	
JumpToLevel10	
JumpToLevel11	
JumpToLevel12	
JumpToLevel13	
JumpToLevel14	
JumpToLevel15	
JumpToLevel16	
JumpToLevel17	
JumpToLevel18	
JumpToLevel19	
JumpToLevel20	
JumpToLevel21	
JumpToLevel22	
JumpToLevel23	
JumpToLevel24	
JumpToLevel25	
JumpToLevel26	
JumpToLevel27	
JumpToLevel28	
JumpToLevel29	
JumpToLevel30	
JumpToLevel31	

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Amplitude Modulation Law Struct

Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

AmplitudeModulationLawStruct

C# _

[This is preliminary documentation and is subject to change.]

Struct of amplitude modulation law.

■ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

public struct AmplitudeModulationLawStruct

Public Structure AmplitudeModulationLawStruct

public value class AmplitudeModulationLawStruct

Members

All Members	Methods	Properties	
Public Public		Instance	□ Declared
		Static	Inherited

]	Icon	Member	Description
		<u>AmplitudeValue</u>	Value of amplitude.
-	≘ ∲	Equals(Object)	Indicates whether this instance and a specified object are equal. (Inherited from ValueType .)
	=∳	GetHashCode()	Returns the hash code for this instance. (Inherited from ValueType .)
=	≡	GetType()	Gets the <u>Type</u> of the current instance.

		(Inherited from Object.)
	Repetitions	Number of repetitions to be mantained.
≡	ToString()	Returns the fully qualified type name of this instance.
		(Inherited from <u>ValueType</u> .)

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Amplitude Value

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AmplitudeModulationLawStruct</u> ► **AmplitudeValue**

C# 🔽

[This is preliminary documentation and is subject to change.]

Value of amplitude.

Declaration Syntax

```
C# Visual Basic Visual C++
public double AmplitudeValue { get; set; }

Public Property AmplitudeValue As Double

public:
property double AmplitudeValue {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Repetitions

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► AmplitudeModulationLawStruct ► Repetitions

C# -

[This is preliminary documentation and is subject to change.]

Number of repetitions to be mantained.

Declaration Syntax

```
C# Visual Basic Visual C++

public ulong Repetitions { get; set; }

Public Property Repetitions As ULong

public:
property unsigned long long Repetitions {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library

Documentation Amplitude Profile Struct Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

AmplitudeProfileStruct

C#

[This is preliminary documentation and is subject to change.]

Struct for amplitude profile.

Equals(Object)

Declaration Syntax

C# Visual Basic Visual C++

public struct AmplitudeProfileStruct

Public Structure AmplitudeProfileStruct

public value class AmplitudeProfileStruct

Members

Icon Member	1	Description	
		Static	Inherited
Public Public			□ Declared
All Members	Methods	Properties	

		▼ Static	Innerited
Icon	Member	Description	
	<u>AmplitudeInf</u>	Low frequency constant v	alue.
	<u>AmplitudeSup</u>	High frequency constant v	/alue.
	<u>BandWidth</u>	Bandwidth. Range 027. 62Mhz 2: 32.25Mhz 20: 1	

Indicates whether this instance and a

specified object are equal.

		(Inherited from <u>ValueType</u> .)
	<u>FrequencyInf</u>	Low frequency limit.
=	GetHashCode()	Returns the hash code for this instance. (Inherited from ValueType .)
=	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
	Samples	Array of 256 samples of amplitude segment profile
∄	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Amplitude Inf

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>AmplitudeProfileStruct</u> ► **AmplitudeInf**

C#

[This is preliminary documentation and is subject to change.]

Low frequency constant value.

Declaration Syntax

```
C# Visual Basic Visual C++

public double AmplitudeInf { get; set; }

Public Property AmplitudeInf As Double

public:
property double AmplitudeInf {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Amplitude Sup

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>AmplitudeProfileStruct</u> ► **AmplitudeSup**

C# -

[This is preliminary documentation and is subject to change.]

High frequency constant value.

Declaration Syntax

```
C# Visual Basic Visual C++
public double AmplitudeSup { get; set; }

Public Property AmplitudeSup As Double

public:
property double AmplitudeSup {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation BandWidth

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>AmplitudeProfileStruct</u> ► BandWidth

C#

[This is preliminary documentation and is subject to change.]

Bandwidth. Range 0..27. 0: 125Mhz 1: 62Mhz 2: 32.25Mhz 20: 119.2092Hz

■ Declaration Syntax

```
C# Visual Basic Visual C++

public byte BandWidth { get; set; }

Public Property BandWidth As Byte

public:
property unsigned char BandWidth {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation FrequencyInf

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>AmplitudeProfileStruct</u> ► FrequencyInf

C# -

[This is preliminary documentation and is subject to change.]

Low frequency limit.

Declaration Syntax

```
C# Visual Basic Visual C++

public double FrequencyInf { get; set; }

Public Property FrequencyInf As Double

public:
property double FrequencyInf {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Samples Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>AmplitudeProfileStruct</u> ► Samples

C# _

[This is preliminary documentation and is subject to change.]

Array of 256 samples of amplitude segment profile

Declaration Syntax

```
C# Visual Basic Visual C++
public double[] Samples { get; set; }

Public Property Samples As Double()

public:
property array<double>^ Samples {
    array<double>^ get ();
    void set (array<double>^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ARBBufferType

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBBufferType

C#	~
C 11	

[This is preliminary documentation and is subject to change.]

Arbitrary channel clear memory buffer type.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum ARBBuf	ferType		
Public Enumeration	ARBBufferType		
public enum class	ARBBufferType		

Members

Member	Description
WaveformsSamples	Clear loaded waveforms samples.
GenerationList	Clear waveforms generation list.
ModulatingSamples	Clear modulating samples.
MathOperations	Not Used: Clear math operations between pair channels.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ARBChannel Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel

C#	_
----	---

factor

[This is preliminary documentation and is subject to change.]

Class to manage an arbitrary channel on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class ARBChannel : Channel

Public Class ARBChannel _ Inherits Channel

public ref class ARBChannel: public Channel

Members

All Me	embers	Constructors	Methods	Properties	
				Declare Inherite	
Icon	Member				Desci
∃ ₩	ARBChann	nel()			Initial new ir of the ARBCI class.
	Amplitude	CorrectionFactor			Set amplit correc

		(Inher from Chanr
∃	ATXSSSlaveDisableStartCondition()	Disabl from A cable chann slave (Inher
		from Chanr
≟	ATXSSSlaveDisableStopCondition()	Disabl from / cable chann slave (Inher
		from Chanr
≟	ATXSSSlaveEnableStartCondition(ATXSSEvent)	Enable from A cable chann slave and specific check condit
		(Inher from Chanr
≡ ₩	ATXSSSlaveEnableStopCondition(ATXSSEvent)	Enable from / cable chann slave and sp

		the lir check condit
		(Inher from <u>Chanr</u>
	ChannelFunctionality	Get che function (Arbit DDS, Patter
		(Inher from <u>Chanr</u>
∃ ₩	ClearBuffer(ARBBufferType)	Reset FPGA regist
≅◊	EnableDisable(Boolean)	Enable disable chann switch output off.
		(Inher from <u>Chanr</u>
∃	EnableDisableModulationLaw(Boolean)	Enable Disable module law in FIFO
≓	Equals(Object)	Deteri wheth

		specifi Object equal currer Object
		(Inhei from <u>(</u>
	ErrorResult	Conta result of erro
		(Inher from ATAW
∄	GetHashCode()	Serve hash f for a particitype.
		(Inher from (
Ξ ◊	GetType()	Gets t Type c currer instan
		(Inher from (
≓	GetVersion(String)	Read 1 FPGA versio firmwa loaded
		(Inher from Chanr
=•	<u>LoadAmplitudeModulationLaw(AmplitudeModulationLawStruct[],</u>	Load

	TransferMode, Boolean)	modul law in FIFO.
≅	LoadGenerationSequence(GenerationSequenceStruct[], TransferMode, Boolean)	Load wavef gener seque the FI
- II	LoadWaveforms(WaveformStruct[])	Load wavef and m to cha memo
∃ 🍑	ReadMode(Functionality)	Get the Function of charter (Inhermunit Chanter Charter)
=	ReadRunningWaveformIndex(Int32)	Read i runnir wavef
=•	ReadStatus(ChannelStatus)	Read sof characteristics of char
≡	ResetSoftware()	Make

softwa

		FPGA
		(Inher from Chanr
	SampligRatePrescaler	Set sa rate presca chann Availa only for arbitra chann
		(Inher from Chanr
■	<u>SetAmplitudeVsFrequencyCorrectionFactor(Double)</u>	Set th amplit correct factor output freque
≅	<u>SetExternalTrigger(TriggerSource, SensitivityEdge, TriggerAction)</u>	Set Tr to Ext (Inher from Chanr
≅	SetInternalTrigger()	Set Tr to Into (Inher from Chanr
∃	SetModulationFrequencyPrescaler(Int32)	Set modul freque

		divisic factor.
₫ ◊	SetModulationTriggerMode(ModulationTriggerMode)	Set tr mode modul
≅∳	SetOutputImpedance(OutputImpedance)	Set ou imped for ch (Inheur from Change)
≅	SetOutputVoltage(Single)	Set ou voltag chann chann be in status (Inher from Chann
⊒	SetTriggerDelay(UInt32)	Set a delay chann (Inher from Chanr
∃	SetTriggerMode(TriggerMode)	Set tr mode wavef seque
∉	SetTriggerOut(TriggerSource[], SensitivityEdge)	Set Tr

		out ev (Inher from Chanr
⊕	SetTriggerOutDelay(UInt32)	Set a out de chann (Inher from Chann
∃	ToString()	Return String repres the cu Object (Inher from (

■ Inheritance Hierarchy

Object

ATAWGBase

<u>Channel</u>

ARBChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ARBChannel

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ► **ARBChannel()**

C#	-

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ARBChannel class.

■ Declaration Syntax

C# Visual Basic Visual C++	
<pre>public ARBChannel()</pre>	
Public Sub New	
<pre>public: ARBChannel()</pre>	

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ClearBuffer

Method (buffertype)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ► ClearBuffer(ARBBufferType)

C# _

[This is preliminary documentation and is subject to change.]

Reset the FPGA register.

Declaration Syntax

■ Parameters

buffertype (ARBBufferType)

BufferType to be reset.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation Enable Disable Modulation Law

Method (enable)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ARBChannel ► EnableDisableModulationLaw(Boolean)

C#

[This is preliminary documentation and is subject to change.]

Enable or Disable modulation law in the FIFO

Declaration Syntax

Parameters

enable (Boolean)

True to enable modulation law.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation Load Amplitude Modulation Law Meth (samples, mode, clear Memory)

Namespaces ActiveTechnologies.Instruments.AWG4000.Control

LoadAmplitudeModulationLaw(AmplitudeModulationLawStrutransferMode, Boolean)

C#

[This is preliminary documentation and is subject to change.]

Load modulation law in the FIFO.

Declaration Syntax

```
C#
                     Visual Basic
                                          Visual C++
public ATError LoadAmplitudeModulationLaw(
        AmplitudeModulationLawStruct[] samples,
        TransferMode mode,
         bool clearMemory
)
Public Function LoadAmplitudeModulationLaw (
         samples As AmplitudeModulationLawStruct(),
        mode As <a href="mailto:TransferMode">TransferMode</a>,
         clearMemory As Boolean
) As ATError
public:
ATError^ LoadAmplitudeModulationLaw(
         array<<u>AmplitudeModulationLawStruct</u>>^ samples,
         TransferMode mode,
         bool clearMemory
)
```

■ Parameters

samples (AmplitudeModulationLawStruct [])

Array of AmplitudeModulationLawStruct with sample value and number of time interval to be maintained.

mode (TransferMode)

Transfer mode. ReEntrant for dynamic load of modulation law.

clearMemory (Boolean)

True to clear memory before load sequence.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

ArbStudio Control Classes Library

Documentation Load Generation Sequence Method

(sequence, mode, clear Memory)

Namespaces

Active Technologies Instruments AWG4000 Control

Active Technologies Sequence

<u>ARBChannel</u> ► LoadGenerationSequence(GenerationSequenceStruct[], TransferMode, Boolean)

C#

[This is preliminary documentation and is subject to change.]

Load waveform generation sequence in the FIFO.

Declaration Syntax

```
C#
                     Visual Basic
                                           Visual C++
public ATError LoadGenerationSequence(
         GenerationSequenceStruct[] sequence.
         TransferMode mode,
         bool clearMemory
)
Public Function LoadGenerationSequence (
         sequence As GenerationSequenceStruct(),
         mode As <a href="mailto:TransferMode">TransferMode</a>,
         clearMemory As Boolean
) As ATError
public:
ATError^ LoadGenerationSequence(
         array<<u>GenerationSequenceStruct</u>>^ sequence,
         <u>TransferMode</u> mode,
         bool clearMemory
```

Parameters

sequence (GenerationSequenceStruct [])

Array of GenerationSequenceStruct with 0-based waveform index and number of times to be generated.

mode (TransferMode)

Transfer mode: ReEntrant for dynamic load of waveform sequence.

clearMemory (Boolean)

True to clear memory before load sequence.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation LoadWaveforms

Method (waveforms)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ► LoadWaveforms(WaveformStruct[])

C# _

[This is preliminary documentation and is subject to change.]

Load waveforms and markers to channel memory.

Declaration Syntax

■ Parameters

waveforms (WaveformStruct [])

Array of WaveformStruct with waveform definition.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation ReadRunning Waveform Index

Method (waveformIndex)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ► **ReadRunningWaveformIndex(Int32)**

C#

[This is preliminary documentation and is subject to change.]

Read index of running waveform

Declaration Syntax

Parameters

waveformIndex (Int32)

out Index of running waveform according to waveform list uploaded. Return -1 if channels is in idle mode

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved

LeCroy

ArbStudio Control Classes Library

Documentation Set Amplitude Vs Frequency Correction

eCroy Method (Frequency)

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

<u>ARBChannel</u> ► SetAmplitudeVsFrequencyCorrectionFactor(De

C#

[This is preliminary documentation and is subject to change.]

Set the amplitude correction factor versus output frequency.

Declaration Syntax

Parameters

Frequency (Double)

Output waveform frequency [Hz]. When Frequency = 0 the correction factor is disabled.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright</u> © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library

Documentation Set Modulation Frequency Prescaler Method (modfrequiv factor)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ARBChannel ► **SetModulationFrequencyPrescaler(Int32)**

C#

[This is preliminary documentation and is subject to change.]

Set modulation frequency division factor.

Declaration Syntax

Parameters

modfreqdivfactor (Int32)

Modulation frequency division factor. Range 0 to 16383. 0: no division; 1 frequency / 4; 2: frequency / 8 and so on.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Set Modulation Trigger Mode Method

(triggerMode)

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ►

SetModulationTriggerMode(ModulationTriggerMode)

C#

[This is preliminary documentation and is subject to change.]

Set trigger mode for modulation.

Declaration Syntax

```
C#
                    Visual Basic
                                         Visual C++
public ATError SetModulationTriggerMode(
        <u>ModulationTriggerMode</u> triggerMode
Public Function SetModulationTriggerMode (
        triggerMode As ModulationTriggerMode
) As ATError
public:
ATError^ SetModulationTriggerMode(
        <u>ModulationTriggerMode</u> triggerMode
)
```

Parameters

triggerMode (ModulationTriggerMode)

TriggerMode to be set: Single, Continuous.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetTriggerMode

Method (triggerMode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ARBChannel ► **SetTriggerMode(TriggerMode)**

C# _

[This is preliminary documentation and is subject to change.]

Set trigger mode for waveform sequence.

Declaration Syntax

■ Parameters

triggerMode (TriggerMode)

TriggerMode to be set: Single, Continuous, Stepped, Burst.

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation ATAWGBase Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ATAWGBase

C#	_	

[This is preliminary documentation and is subject to change.]

Provide a base class for AWG4000Control classes

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public class A	ΓΑWGBase		
Public Class A	ΓΑWGBase		

public ref class ATAWGBase

Members

All Me	embers	Constructors	Methods	Properties	
Publ Prot			✓ Instance ✓ Static		Declare Inherite
Icon	Member		Description		
∄	<u>ATAWGBas</u>	se()	Initializes a new ATAWGBase cla	w instance of the ss.	
∃∲	Equals(Ob	vject)	Determines who is equal to the of the office	•	d <u>Object</u>
	<u>ErrorResu</u>	<u>lt</u>	Contain the res	ult status of erro	7.
≡	GetHashC	ode()	Serves as a has	sh function for a p	articular

type.

		(Inherited from Object.)
≟	GetType()	Gets the <u>Type</u> of the current instance.
		(Inherited from Object.)
≡	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> .
		(Inherited from Object.)

■ Inheritance Hierarchy

Object

ATAWGBase

- Channel
- Conversions
- Device
- DeviceSet
- <u>PairChannels</u>
- <u>PatternGenerator</u>

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ATAWGBase

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>ATAWGBase</u> ► **ATAWGBase()**

C#	-

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ATAWGBase class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public ATAWGBase()</pre>			
Public Sub New			
<pre>public: ATAWGBase()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation **ErrorResult**

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ATAWGBase ► **ErrorResult**

C# -

[This is preliminary documentation and is subject to change.]

Contain the result status of error.

■ Declaration Syntax

```
C# Visual Basic Visual C++
public ATError ErrorResult { get; set; }

Public Property ErrorResult As ATError

public:
property ATError^ ErrorResult {
        ATError^ get ();
        void set (ATError^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ATError Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► **ATError**

C#

[This is preliminary documentation and is subject to change.]

Class for error handling

■ Declaration Syntax

C#	Visual Basic	Visual C++		
public class ATError				
Public Class ATErr	ror			
nublic nof class A	TEnnon			
public ref class A	() E. l.Ol.			

Members

inclination						
All Members Constructors		Methods	Properties			
			☑ Instance ☑ Static		Declare Inherite	
Icon Member			Description			
€	ATError()		Initializes a new instance of the ATError class.			
- II 🍑	ClearError()		Clear error to RES_SUCCESS level			
∃	Equals(Object)		Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> . (Inherited from <u>Object</u> .)			
	ErrorCode		Code of error or	ccured		

i i	<u>ErrorDescription</u>	Description of error occured. From 0 to 100000 are internal error
	<u>ErrorSource</u>	Error level
∄	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object.)
≅∳	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
∃	SetError(Int32, Int32)	Set error state
=•	SetError(Int32, Int32, String)	Set error state
≡	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

ATError

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ATError

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

ATError()

C#	~
U ''	4

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ATError class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public ATError()</pre>			
Public Sub New			
<pre>public: ATError()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ClearError Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

ClearError()

C#	•
•	d

[This is preliminary documentation and is subject to change.]

Clear error to RES_SUCCESS level

Declaration Syntax

C# Visual Basic Visual C++

public void ClearError

Public Sub ClearError

public:
void ClearError()

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ErrorCode Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

ErrorCode

C#

[This is preliminary documentation and is subject to change.]

Code of error occured

Declaration Syntax

```
C# Visual Basic Visual C++

public int ErrorCode { get; }

Public ReadOnly Property ErrorCode As Integer

public:
    property int ErrorCode {
        int get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Error Description

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

ErrorDescription

C# -

[This is preliminary documentation and is subject to change.]

Description of error occured. From 0 to 100000 are internal error

Declaration Syntax

```
C# Visual Basic Visual C++

public string ErrorDescription { get; }

Public ReadOnly Property ErrorDescription As String

public:
    property String^ ErrorDescription {
        String^ get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation **ErrorSource**

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

ErrorSource

C# -

[This is preliminary documentation and is subject to change.]

Error level

Declaration Syntax

```
C# Visual Basic Visual C++

public int ErrorSource { get; }

Public ReadOnly Property ErrorSource As Integer

public:
   property int ErrorSource {
        int get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetError Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ► **SetError()**

C#		,
•	4	

[This is preliminary documentation and is subject to change.]

Members

Icon	Member	Description
-= (SetError(Int32, Int32)	Set error state
≅∲	SetError(Int32, Int32, String)	Set error state

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation SetError Method

(errSource, errCode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

SetError(Int32, Int32)

C#

[This is preliminary documentation and is subject to change.]

Set error state

Declaration Syntax

```
Visual Basic
C#
                                            Visual C++
public void SetError(
         int errSource,
         int errCode
)
Public Sub SetError (
         errSource As Integer,
         errCode As <a href="Integer">Integer</a>
)
public:
void SetError(
         int errSource,
         int errCode
)
```

Parameters

errSource (Int32)

Source level for error: RES_SUCCESS = no error; RES_SYSTEM = system error not recoverable; RES_APPLICATION = application error recoverable

errCode (Int32)

Code for error. Refer to ErrorCodes class for list.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetError Method

(errSource, errCode, errDescription)

Namespaces ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ATError</u> ►

SetError(Int32, Int32, String)

C#

[This is preliminary documentation and is subject to change.]

Set error state

Declaration Syntax

```
Visual Basic
C#
                                          Visual C++
public void SetError(
         int errSource,
         int errCode,
         string errDescription
)
Public Sub SetError (
         errSource As Integer,
         errCode As <u>Integer</u>,
         errDescription As String
)
public:
void SetError(
         <u>int</u> errSource,
         int errCode,
         String<sup>^</sup> errDescription
)
```

Parameters

errSource (Int32)

Source level for error: RES_SUCCESS = no error; RES_SYSTEM = system error not recoverable; RES APPLICATION = application error recoverable

errCode (Int32)

Code for error. Refer to ErrorCodes class for list.

errDescription (String)

Brief description of error.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ATXSSEvent

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ATXSSEvent

C#	~
U 11	

[This is preliminary documentation and is subject to change.]

Line source for TriggerOUT channel trigger and handled event signal.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum ATXSSE	vent		
Public Enumeration	ATXSSEvent		
public enum class	ATXSSEvent		

Members

Member	Description
None	None.
Stop	Stop.
Start	Start.
AWGMarker	AWGMarker.
PatEvent	PatternEvent.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation AWG4000 Pattern Regs Struct

Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

AWG4000_PatternRegsStruct

C#

[This is preliminary documentation and is subject to change.]

Struct-Class for ArbStudio_Pattern Registers

Declaration Syntax

C#	Visual Basic	Visual C++	
public class AWG4	000_PatternRegsSt	ruct	

Public Class AWG4000_PatternRegsStruct

public ref class AWG4000_PatternRegsStruct

Members

All Me	embers	Constructors	Me	ethods	Properties	
✓ Publ ✓ Prot	_			instance Static		Declare Inherite
Icon	Member			Description		
≟ ∳	AWG4000	<u>PatternRegsStruc</u>	<u>t()</u>		ew instance of th PatternRegsStru	
	ClkGenRe	<u>gs</u>				
	ControlDo	C				
	ControlRe	<u>y</u>				
	DisableTRO					

		(Inherited from Object.)
	FreqDivision	
=	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object.)
≡	CotType()	
	<u>GetType()</u>	Gets the <u>Type</u> of the current instance.
		(Inherited from Object.)
	RepProcReg	
	ResetPLL	
	SCRegs	
	SelStartHizReg	
	<u>StartAddrBlk</u>	
	<u>StartAddrInstrSeq</u>	
∄	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)
	TrigINOUTSotup	(Innericed from <u>Object</u>)
	TrigINOUTSetup	
	TriaReas	

■ Inheritance Hierarchy

Object

AWG4000_PatternRegsStruct

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation AWG4000 Pattern Regs Struct

Constructor

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► AWG4000 PatternRegsStruct ►

AWG4000_PatternRegsStruct()

C#

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the AWG4000_PatternRegsStruct class

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public AWG4000_Pat</pre>	ternRegsStruct()		
Public Sub New			
<pre>public: AWG4000_PatternReg</pre>	sStruct()		

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation ClkGenRegs

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000_PatternRegsStruct</u> ► **ClkGenRegs**

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ClockGenStruct ClkGenRegs { get; set; }

Public Property ClkGenRegs As ClockGenStruct

public:
property ClockGenStruct^ ClkGenRegs {
        ClockGenStruct^ get ();
        void set (ClockGenStruct^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Control Reg

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000_PatternRegsStruct</u> ► **ControlReg**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte ControlReg { get; set; }

Public Property ControlReg As Byte

public:
property unsigned char ControlReg {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DisableTRG

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000 PatternRegsStruct</u> ► **DisableTRG**

C# 🔽

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte DisableTRG { get; set; }

Public Property DisableTRG As Byte

public:
property unsigned char DisableTRG {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Freq Division

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000 PatternRegsStruct</u> ► **FreqDivision**

C# 🔽

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ulong FreqDivision { get; set; }

Public Property FreqDivision As ULong

public:
property unsigned long long FreqDivision {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RepProcReg

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► AWG4000_PatternRegsStruct ► RepProcReg

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public ushort RepProcReg { get; set; }

Public Property RepProcReg As UShort

public:
property unsigned short RepProcReg {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ResetPLL Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► AWG4000 PatternRegsStruct ► ResetPLL

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte ResetPLL { get; set; }

Public Property ResetPLL As Byte

public:
property unsigned char ResetPLL {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SCRegs Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>AWG4000_PatternRegsStruct</u> ► **SCRegs**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public SuperCableStruct SCRegs { get; set; }

Public Property SCRegs As SuperCableStruct

public:
property SuperCableStruct^ SCRegs {
    SuperCableStruct^ get ();
    void set (SuperCableStruct^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SelStartHizReg

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000_PatternRegsStruct</u> ► **SelStartHizReg**

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public byte SelStartHizReg { get; set; }

Public Property SelStartHizReg As Byte

public:
property unsigned char SelStartHizReg {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation StartAddrBlk

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000 PatternRegsStruct</u> ► **StartAddrBlk**

C# 🔽

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ulong StartAddrBlk { get; set; }

Public Property StartAddrBlk As ULong

public:
property unsigned long long StartAddrBlk {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation StartAddrInstrSeq

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000_PatternRegsStruct</u> ► **StartAddrInstrSeq**

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ushort StartAddrInstrSeq { get; set; }

Public Property StartAddrInstrSeq As UShort

public:
property unsigned short StartAddrInstrSeq {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TrigINOUTSetup

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>AWG4000 PatternRegsStruct</u> ► **TrigINOUTSetup**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ushort TrigINOUTSetup { get; set; }

Public Property TrigINOUTSetup As UShort

public:
property unsigned short TrigINOUTSetup {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TrigRegs Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>AWG4000_PatternRegsStruct</u> ► **TrigRegs**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public TriggerStruct TrigRegs { get; set; }

Public Property TrigRegs As TriggerStruct

public:
property TriggerStruct^ TrigRegs {
          TriggerStruct^ get ();
          void set (TriggerStruct^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Channel Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► **Channel**

C#

[This is preliminary documentation and is subject to change.]

Base class to manage Arbitrary or DDS channel on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class Channel : ATAWGBase

Public Class Channel _

Inherits ATAWGBase

public ref class Channel : public ATAWGBase

Members

All Members	Constructors	Methods	Properties	
Public Protected Protected		☐ Instance ☐ Static		Declare

		IIIICIIC
Icon	Member	Description
≟	Channel()	Initializes a new instance of the base channel class.
	AmplitudeCorrectionFactor	Set amplitude correction factor

≡	ATXSSSlaveDisableStartCondition()	Disable start from AT- XSS cable for channel
		in slave mode.

≅◊	ATXSSSlaveDisableStopCondition()	Disable stop from AT- XSS cable for channel in slave mode.
≡ ♠	ATXSSSlaveEnableStartCondition(ATXSSEvent)	Enable start from AT- XSS cable for channel in slave mode and specify the line for check start condition.
≡••	ATXSSSlaveEnableStopCondition(ATXSSEvent)	Enable stop from AT- XSS cable for channel in slave mode and specify the line for check stop condition.
	ChannelFunctionality	Get channel functionality (Arbitrary DDS, Pattern)
≅∳	EnableDisable(Boolean)	Enable or disable channel switching output on or off.
≅♦	Equals(Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> . (Inherited from <u>Object</u> .)
	<u>ErrorResult</u>	Contain the result status of error.

		(Inherited from <u>ATAWGBase</u> .)
∃	GetHashCode()	Serves as a hash function for a particular type.
		(Inherited from Object.)
≡ ₩	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
≡	Catleraian (Ctring)	
	GetVersion(String)	Read from FPGA the version of firmware loaded.
≅∳	ReadMode(Functionality)	Get the Functionality of channel.
≅∳	ReadStatus(ChannelStatus)	Read status of channel Idle or Busy.
₫ ڼ	ResetSoftware()	Make a reset software in FPGA
	<u>SampligRatePrescaler</u>	Set sampling rate prescaler for channel. Available only for arbitrary channel
≡	SetExternalTrigger(TriggerSource, SensitivityEdge, TriggerAction)	Set Trigger to Externa

=•	SetInternalTrigger()	Set Trigger to Internal
≡	SetOutputImpedance(OutputImpedance)	Set output impedance for channel.
≅∳	SetOutputVoltage(Single)	Set output voltage for channel. The channel must be in stop status
∃∳	SetTriggerDelay(UInt32)	Set a start delay on channel
≅∳	SetTriggerOut(TriggerSource[], SensitivityEdge)	Set Trigger out events
≟	SetTriggerOutDelay(UInt32)	Set a trigger out delay on channel
∃ ∳	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

ATAWGBase

Channel

ARBChannel

DDSChannel

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Channel

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

Channel()

 C#	-
U 11	

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the base channel class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public Channel()</pre>			
Public Sub New			
<pre>public: Channel()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Amplitude Correction Factor

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

AmplitudeCorrectionFactor

C# -

[This is preliminary documentation and is subject to change.]

Set amplitude correction factor

Declaration Syntax

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation ATXSSS lave Disable Start Condition

Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ATXSSSlaveDisableStartCondition()

C#

[This is preliminary documentation and is subject to change.]

Disable start from AT-XSS cable for channel in slave mode.

Declaration Syntax

C# Visual Basic Visual C++

public ATError ATXSSSlaveDisableStartCondition()

Public Function ATXSSSlaveDisableStartCondition As ATError

public:

ATError^ ATXSSSlaveDisableStartCondition()

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation ATXSSS lave Disable Stop Condition

Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ATXSSSlaveDisableStopCondition()

C#

[This is preliminary documentation and is subject to change.]

Disable stop from AT-XSS cable for channel in slave mode.

■ Declaration Syntax

C# Visual Basic Visual C++

public ATError ATXSSSlaveDisableStopCondition()

Public Function ATXSSSlaveDisableStopCondition As <u>ATError</u>

public:

ATError^ ATXSSSlaveDisableStopCondition()

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation ATXSSS lave Enable Start Condition

Method (xssEvent)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ATXSSSlaveEnableStartCondition(ATXSSEvent)

C# _

[This is preliminary documentation and is subject to change.]

Enable start from AT-XSS cable for channel in slave mode and specify the line for check start condition.

Declaration Syntax

Parameters

xssEvent (ATXSSEvent)

Event for start condition.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation ATXSSS lave Enable Stop Condition

Method (xssEvent)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ATXSSSlaveEnableStopCondition(ATXSSEvent)

C# -

[This is preliminary documentation and is subject to change.]

Enable stop from AT-XSS cable for channel in slave mode and specify the line for check stop condition.

Declaration Syntax

Parameters

xssEvent (ATXSSEvent)

ATSSLine for stop condition.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Channel Functionality Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ChannelFunctionality

C# -

[This is preliminary documentation and is subject to change.]

Get channel functionality (Arbitrary, DDS, Pattern...)

Declaration Syntax

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation **EnableDisable**

Method (chanEnabled)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

EnableDisable(Boolean)

C# _

[This is preliminary documentation and is subject to change.]

Enable or disable channel switching output on or off.

Declaration Syntax

■ Parameters

chanEnabled (Boolean)

true to enable channel, false to disable.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



[This is preliminary documentation and is subject to change.]

Read from FPGA the version of firmware loaded.

Declaration Syntax

■ Parameters

```
version ( <u>String</u> ) out Version of firmware.
```

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadMode Method

(functionality)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ReadMode(Functionality)

C# _

[This is preliminary documentation and is subject to change.]

Get the Functionality of channel.

Declaration Syntax

■ Parameters

functionality (Functionality)

out functionality of channel.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadStatus

Method (status)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► Channel ►

ReadStatus(ChannelStatus)

C# _

[This is preliminary documentation and is subject to change.]

Read status of channel: Idle or Busy.

Declaration Syntax

■ Parameters

status (ChannelStatus)

out ChannelStatus of channel.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation ResetSoftware

Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

ResetSoftware()

\cap #	₩
$\subset \pi$	

[This is preliminary documentation and is subject to change.]

Make a reset software in FPGA

Declaration Syntax

C# Visual Basic Visual C++

public ATError ResetSoftware()

Public Function ResetSoftware As ATError

public: ATError^ ResetSoftware()

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Samplig Rate Prescaler Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

SampligRatePrescaler

C# -

[This is preliminary documentation and is subject to change.]

Set sampling rate prescaler for channel. Available only for arbitrary channel

Declaration Syntax

```
C# Visual Basic Visual C++
public ulong SampligRatePrescaler { get; set; }

Public Property SampligRatePrescaler As ULong

public:
property unsigned long long SampligRatePrescaler {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetExternalTrigger

Method (source, edge, action)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

Visual C++

SetExternalTrigger(TriggerSource, SensitivityEdge, **TriggerAction**)

C#



[This is preliminary documentation and is subject to change.]

Set Trigger to External

Declaration Syntax

```
C#
                    Visual Basic
public ATError SetExternalTrigger(
        TriggerSource, source,
        SensitivityEdge edge,
        <u>TriggerAction</u> action
)
Public Function SetExternalTrigger ( _
        source As TriggerSource,
        edge As SensitivityEdge,
        action As <u>TriggerAction</u>
) As ATError
public:
ATError^ SetExternalTrigger(
        TriggerSource, source,
        SensitivityEdge edge,
        TriggerAction action
)
```

Parameters

```
source (TriggerSource)
  Source for trigger
edge (SensitivityEdge)
  Edge of signal
action (TriggerAction)
```

Action to perform

■ Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetInternalTrigger

Method

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► Channel ►

SetInternalTrigger()

C#	▼1
U#	

[This is preliminary documentation and is subject to change.]

Set Trigger to Internal

Declaration Syntax

Visual Basic Visual C++ C#

public ATError SetInternalTrigger()

Public Function SetInternalTrigger As ATError

public:

ATError^ SetInternalTrigger()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation SetOutputImpedance Method

(outimpedance)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

SetOutputImpedance(OutputImpedance)

C# -

[This is preliminary documentation and is subject to change.]

Set output impedance for channel.

Declaration Syntax

Parameters

outimpedance (OutputImpedance)

OutputImpedance to be set.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetOutputVoltage

Method (voltage)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

SetOutputVoltage(Single)

C# -

[This is preliminary documentation and is subject to change.]

Set output voltage for channel. The channel must be in stop status.

Declaration Syntax

■ Parameters

```
voltage (Single)
```

Output voltage to be set (-6...+6)

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetTriggerDelay

Method (delaySamples)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

SetTriggerDelay(UInt32)

		-
C #		,
U ''	4	

[This is preliminary documentation and is subject to change.]

Set a start delay on channel

Declaration Syntax

■ Parameters

delaySamples (UInt32)

Samples of delay time

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetTriggerOut

Method (sources, edge)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► Channel ►

SetTriggerOut(TriggerSource[], SensitivityEdge)

C# _

[This is preliminary documentation and is subject to change.]

Set Trigger out events

Declaration Syntax

```
Visual C++
C#
                    Visual Basic
public ATError SetTriggerOut(
        TriggerSource[] sources,
        SensitivityEdge edge
)
Public Function SetTriggerOut (
        sources As TriggerSource(),
        edge As SensitivityEdge
) As <u>ATError</u>
public:
ATError^ SetTriggerOut(
        array<TriggerSource>^ sources,
        SensitivityEdge edge
)
```

Parameters

```
sources ( TriggerSource [])
```

List of sources which generates a trigger out event

edge (SensitivityEdge)

Signal polarity

Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation SetTriggerOutDelay Method

(delaySamples)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Channel</u> ►

SetTriggerOutDelay(UInt32)

C# -

[This is preliminary documentation and is subject to change.]

Set a trigger out delay on channel

Declaration Syntax

Parameters

delaySamples (UInt32)

Samples of delay time

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Channel Out Logic Operation

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ChannelOutLogicOperation

C#

[This is preliminary documentation and is subject to change.]

Trigger combination between channels and pairs

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Channel	lOutLogicOperati	on	

Public Enumeration ChannelOutLogicOperation

public enum class ChannelOutLogicOperation

Members

Member	Description
OperationDISABLED	Operation disabled.
OperationFALSE	Trigger operation is always FALSE.
OperationTRUE	Trigger operation is always TRUE.
OperationAND	Trigger operation AND.
OperationOR	Trigger operation OR.

OperationXOR	Trigger operation XOR.
OperationNAND	Trigger operation NAND.
OperationNOR	Trigger operation NOR.
OperationXNOR	Trigger operation XNOR.
OperationTriggerOut	Trigger operation TriggerOutForPattern.
SelectionAWG	On mixed couple select AWG out trigger for final operation
SelectionPattern	On mixed couple select Pattern out trigger fo final operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Channel Status

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ChannelStatus

C#	-
U 11	4

[This is preliminary documentation and is subject to change.]

Status of channel

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Chann	elStatus		
Public Enumeration ChannelStatus			
public enum class	ChannelStatus		

Members

Member	Description
Idle	Channel is idle mode and ready to execute commands.
Busy	Channel is busy and not ready to execute commands.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Clock GenStruct

Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockGenStruct

 C#	-
C II	

[This is preliminary documentation and is subject to change.]

Pattern generator struct for clock generator.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public class ClockGenStruct			
Public Class ClockGenStruct			
public ref class ClockGenStruct			

Members

All Me	embers	Constructors	Methods	Properties	
Pub Prot	lic ected		□ Instance □ Static □		Declare Inherite
Icon	Member		Description		
∄	ClockGenS	Struct()	Initializes a new ClockGenStruck	v instance of the ct class	
	ClkStrobe	Ор			
	<u>DelayClk</u>				
	<u>Enable</u>				
=	Equals(Ob	ject)	Determines who	ether the specified	d <u>Object</u>

		is equal to the current Object.
		(Inherited from Object.)
-= (GetHashCode()	Serves as a hash function for a particular type.
		(Inherited from Object.)
∃	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
	<u>HighSteps</u>	
	LowSteps	
≓	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

ClockGenStruct

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Clock GenStruct

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockGenStruct()

C#	_

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ClockGenStruct class

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public ClockGenStr	<pre>public ClockGenStruct()</pre>		
Public Sub New			
<pre>public: ClockGenStruct()</pre>			
CTOCKGEHSCH dCC()			

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation ClkStrobeOp

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockGenStruct ► ClkStrobeOp

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte ClkStrobeOp { get; set; }

Public Property ClkStrobeOp As Byte

public:
property unsigned char ClkStrobeOp {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DelayClk Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockGenStruct ► DelayClk

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte DelayClk { get; set; }

Public Property DelayClk As Byte

public:
property unsigned char DelayClk {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation Enable Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockGenStruct ► Enable

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte Enable { get; set; }

Public Property Enable As Byte

public:
property unsigned char Enable {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation HighSteps

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ClockGenStruct</u> ► **HighSteps**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte HighSteps { get; set; }

Public Property HighSteps As Byte

public:
property unsigned char HighSteps {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation LowSteps Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>ClockGenStruct</u> ► **LowSteps**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte LowSteps { get; set; }

Public Property LowSteps As Byte

public:
property unsigned char LowSteps {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation ClockSource

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ClockSource

C#		

[This is preliminary documentation and is subject to change.]

Clock source for channel.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum ClockSource			
Public Enumeration ClockSource			
public enum class	ClockSource		

■ Members

Member	Description
Internal	Clock source is internal.
External	Clock source is external.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation ClockStrobeOpType Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ClockStrobeOpType

C#	•
	4

[This is preliminary documentation and is subject to change.]

Pattern generators sets parameters for the strobe operation of the Clock Generator Module.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum ClockStrobeOpType			
Public Enumeration ClockStrobeOpType			
<pre>public enum class ClockStrobeOpType</pre>			

Members

Member	Description
NOP	No strobe operation.
OR	Logic operation OR.
AND	Logic operation AND.
XOR	Logic operation XOR.
FORCE_0	Force the clock to GND.

NOT	Logic operation NOT.
NOR	Logic operation NOR.
NAND	Logic operation NAND.
XNOR	Logic operation XNOR.
FORCE_1	Force the clock to VCC value.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Conversions Class

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

Conversions

C#	_

[This is preliminary documentation and is subject to change.]

Class to manage conversion between data and instruments data

Declaration Syntax

C# Visual Basic Visual C++

public class Conversions : ATAWGBase

Public Class Conversions _ Inherits ATAWGBase

public ref class Conversions : public ATAWGBase

Members

All Me	embers	Constructors	Methods	Properties	
Public Protected					Declare Inherite
Icon Member				Description	
∃	<u>Conversions()</u>		Initializes a new instance of the Conversions class		

ConvertFrequencyToBaseFrequency(Double, UInt64)
Convert a frequency in range 0...250Mhz to instrument frequency

Equals(Object)

Determines whether the specified Object is equal to the current Object.

		(Inherited from Object.)
	ErrorResult	Contain the result status of error. (Inherited from ATAWGBase.)
•		ATAWODase.)
≅	GetHashCode()	Serves as a hash function for a particular type.
		(Inherited from Object.)
= (GetType()	Gets the <u>Type</u> of the current instance.
		(Inherited from Object.)
≓	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> .
		(Inherited from Object.)

■ Inheritance Hierarchy

Object

<u>ATAWGBase</u>

Conversions

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Conversions

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Conversions ► Conversions()

C#	-

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the **Conversions** class

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public Conversi	ons()		
Public Sub New			
<pre>public:</pre>			
<pre>public: Conversions()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



C#

ArbStudio Control Classes Library

Documentation Convert Frequency To Base Frequency

Method (baseFrequency, prescaler)

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

Visual C++

▶ Conversions

ConvertFrequencyToBaseFrequency(Double, UInt64)

C#

[This is preliminary documentation and is subject to change.]

Convert a frequency in range 0...250Mhz to instrument frequency

Visual Basic

Declaration Syntax

Parameters

```
baseFrequency ( Double )
  Frequency calculated

prescaler ( UInt64 )
  Frequency division factor applied
```

Return Value

ATError result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DDSBufferType

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSBufferType

C#	•
C 11	4

[This is preliminary documentation and is subject to change.]

DDS channel clear memory buffer type.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum DDSBuf	ferType		
Public Enumeration	DDSBufferType		
public enum class	DDSBufferType		

Members

Member	Description
CarrierWaveform	Clear carrier waveform.
ModulatingSamples	Clear modulating samples.
AmplitudeProfile	Clear amplitude profile.
MathOperations	Not Used: Clear math operations between pair channels.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DDSChannel Class

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

DDSChannel

C#	
----	--

[This is preliminary documentation and is subject to change.]

Base class to manage an phase modulated or frequency modulated DDS channel on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class DDSChannel : Channel

Public Class DDSChannel _ Inherits Channel

public ref class DDSChannel : public Channel

ATXSSSlaveDisableStartCondition()

Members

All Members	Constructors	Methods	Properties	
Public Protected		✓ Instance ✓ Static		Declare Inherite

Prot	ected	▽ Static		Inherite
Icon	Member		Description	
€	DDSChannel()		Initializes a neinstance of the DDSChannel of	9
iii iii	<u>AmplitudeCorrectionFactor</u>		Set amplitude correction fact (Inherited from	or

Channel.)

Disable start from AT-

		XSS cable for channel in slave mode.
		(Inherited from Channel.)
≅ ©	ATXSSSlaveDisableStopCondition()	Disable stop from AT- XSS cable for channel in slave mode.
		(Inherited from Channel .)
∃	ATXSSSlaveEnableStartCondition(ATXSSEvent)	Enable start from AT- XSS cable for channel in slave mode and specify the line for check start condition.
		(Inherited from Channel .)
⊕	ATXSSSlaveEnableStopCondition(ATXSSEvent)	Enable stop from AT- XSS cable for channel in slave mode and specify the line for check stop condition.
		(Inherited from Channel.)
	ChannelFunctionality	Get channel functionality (Arbitrary DDS, Pattern)
		(Inherited from Channel .)
≅∳	ClearBuffer(DDSBufferType)	Reset the FPGA register.
≡	EnableDisable(Boolean)	Enable or disable channel switching output on or off.

		(Inherited from Channel.)
	EnableFlatnessCompensation	Enables/disbles the internal amplitude flatness compensation When true the user defined amplitude profile will be automatically compensated with the internal compensation factors. When false the user defined amplitude profile will be applied as is.
≅∳	Equals(Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
		(Inherited from Object.)
	ErrorResult	Contain the result status of error.
		(Inherited from <u>ATAWGBase</u> .)
=	GetHashCode()	Serves as a hash function for a particular type.
		(Inherited from Object.)
≅◊	GetType()	Gets the <u>Type</u> of the current instance.
		(Inherited from

		Object.)
∃	GetVersion(String)	Read from FPGA the version of firmware loaded.
		(Inherited from Channel .)
≘₩	LoadCarrier(WaveformStruct)	Load waveform and marker to channel memory.
	<u>MemorySamples</u>	Memory samples country for DDS carrier
≓	ReadMode(ModulationType)	Get the Modulation type of channel DDS.
∃ ₩	ReadMode(Functionality)	Get the Functionality of channel. (Inherited from Channel.)
∄	ReadStatus(ChannelStatus)	Read status of channe Idle or Busy. (Inherited from Channel.)
∃\	ResetSoftware()	Make a reset software in FPGA (Inherited from Channel.)
	<u>SampligRatePrescaler</u>	Set sampling rate prescaler for channel. Available only for arbitrary channel

		(Inherited from Channel.)
≓	SetAmplitudeProfile(AmplitudeProfileStruct)	Set amplitude profile of the generated signal a a function of its frequency.
≡	<u>SetExternalTrigger(TriggerSource,</u> <u>SensitivityEdge, TriggerAction)</u>	Set Trigger to Externa (Inherited from Channel.)
≡	SetInternalTrigger()	Set Trigger to Internal (Inherited from Channel.)
≝₩	SetOutputImpedance(OutputImpedance)	Set output impedance for channel. (Inherited from Channel.)
≡ ₩	SetOutputVoltage(Single)	Set output voltage for channel. The channel must be in stop status (Inherited from Channel.)
≡	SetTriggerDelay(UInt32)	Set a start delay on channel (Inherited from Channel.)
≓	SetTriggerMode(TriggerMode)	Set trigger mode for waveform sequence: Single, Continuous, Burst.
=		

	SetTriggerOut(TriggerSource[], SensitivityEdge)	Set Trigger out events (Inherited from Channel.)
≘ ₩	SetTriggerOutDelay(UInt32)	Set a trigger out delay on channel (Inherited from Channel.)
≅ 🍑	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

ATAWGBase

<u>Channel</u>

DDSChannel

FSKChannel

PSKChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DDSChannel

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ► **DDSChannel()**

_

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the DDSChannel class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public DDSChannel	()		
Public Sub New			
<pre>public: DDSChannel()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation ClearBuffer

Method (buffertype)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ► ClearBuffer(DDSBufferType)

C# _

[This is preliminary documentation and is subject to change.]

Reset the FPGA register.

Declaration Syntax

■ Parameters

buffertype (DDSBufferType)

BufferType to be reset.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved

<u>LeCroy</u>

ArbStudio Control Classes Library

Documentation Enable Flatness Compensation

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

DDSChannel ► **EnableFlatnessCompensation**

C# _

[This is preliminary documentation and is subject to change.]

Enables/disbles the internal amplitude flatness compensation. When true the user defined amplitude profile will be automatically compensated with the internal compensation factors. When false the user defined amplitude profile will be applied as is.

Declaration Syntax

```
C# Visual Basic Visual C++
public bool EnableFlatnessCompensation { get; set; }

Public Property EnableFlatnessCompensation As Boolean

public:
property bool EnableFlatnessCompensation {
    bool get ();
    void set (bool value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation LoadCarrier

Method (waveform)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ► LoadCarrier(WaveformStruct)

C# 🔽

[This is preliminary documentation and is subject to change.]

Load waveform and marker to channel memory.

Declaration Syntax

Parameters

waveform (WaveformStruct)

WaveformStruct with waveform definition

■ Return Value

ATError class result of operation.

Remarks

In DDS channel only 1 waveform and 1 marker is allowed. For DDS standard, number of samples is 2^20/memoryDivision, for DDS High Speed, number of samples is fixed to 2560 and no memory division is applied. All memory must be filled.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation MemorySamples

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ► **MemorySamples**

C#

[This is preliminary documentation and is subject to change.]

Memory samples count for DDS carrier

Declaration Syntax

```
C# Visual Basic Visual C++

public long MemorySamples { get; }

Public ReadOnly Property MemorySamples As Long

public:
   property long long MemorySamples {
        long long get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadMode Method

(modulation)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

DDSChannel ► ReadMode(ModulationType)

C#

[This is preliminary documentation and is subject to change.]

Get the Modulation type of channel DDS.

Declaration Syntax

■ Parameters

modulation (ModulationType)

out ModulationType of channel.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation Set Amplitude Profile Method

(inputProfile)

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ►

SetAmplitudeProfile(AmplitudeProfileStruct)

C#

[This is preliminary documentation and is subject to change.]

Set amplitude profile of the generated signal as a function of its frequency.

Declaration Syntax

```
C#
                    Visual Basic
                                         Visual C++
public ATError SetAmplitudeProfile(
        <u>AmplitudeProfileStruct</u> inputProfile
Public Function SetAmplitudeProfile (
        inputProfile As AmplitudeProfileStruct
) As ATError
public:
ATError^ SetAmplitudeProfile(
        <u>AmplitudeProfileStruct</u> inputProfile
)
```

Parameters

inputProfile (AmplitudeProfileStruct)

AmplitudeProfileStruct with amplitude profile data.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved

LeCroy

ArbStudio Control Classes Library Documentation SetTriggerMode

Method (triggerMode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

DDSChannel ► **SetTriggerMode(TriggerMode)**

C# _

[This is preliminary documentation and is subject to change.]

Set trigger mode for waveform sequence: Single, Continuous, Burst.

Declaration Syntax

■ Parameters

triggerMode (TriggerMode)

TriggerMode to be set.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Device Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► **Device**

		_
C#	_	•

queued bulk

[This is preliminary documentation and is subject to change.]

Class to manage an Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class Device : ATAWGBase

Public Class Device _

Inherits ATAWGBase

public ref class Device : public ATAWGBase

Members

All Members	Constructors	Methods	Properties	
Public Protected Protected		☑ Instance ☑ Static		✓Declare✓Inherite

W 1 1 0 C	ecteu	₩ Static		Inherite
Icon	Member		Description	1
⊕	Device()		Initializes a instance of t class.	_
∃	AbortPendingTransfers()		Break all per (NonReentra tranfers to the channels.	nt)
≡	AbortTransfers(TransferTypeI	<u>Def)</u>	Abort all pen	iding and

		transfers.
	<u>AWGModelType</u>	Get AWG model type
∃	<u>CloseCompletedTransfers()</u>	Close all completed bulk data transfers.
	<u>DeviceUsbId</u>	Get USB device enumeration.
∄	<u>DisableFPTriggerOut()</u>	Disable front panel global trigger out.
∃	EnableFPTriggerOut()	Enable front panel global trigger out.
≅₩	Equals(Object)	Determines whether the specified <u>Object</u> i equal to the current <u>Object</u> . (Inherited from <u>Object</u> .)
	ErrorResult	Contain the result status of error. (Inherited from ATAWGBase.)
∃ ₩	ForceStop(Byte[])	Send a stop trigger t given channels
=•	ForceTrigger(Byte[])	Send a trigger to

		given channels
≅	GetChannel(Int32)	Return device channel for the giver id
⊕	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object.)
≅ ∲	<u>GetPatternGenerator(Int32)</u>	Return device patteringenerator for the given id
≅₩	GetTransferStatus(Int32, TransferStatus, Int32)	Query the status of the (NonReentrant) transfers to the channels.
≅	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
∃	Initialize(Functionality[])	Initialize device with creation of channels with given channel Functionality.
	InstalledMemory	Get installed memory size of instrument

	<u>PairLeft</u>	Get or Set the left pair of channels
	<u>PairRight</u>	Get or Set the left pair of channels (only for AWG 4000)
=	PowerLedManage(Byte)	Control power led
≅∲	RUN(Byte[])	Run instrument
in the second	SerialId	Get Serial Id of connected device
€₩	SetATXSSDeSkewDelay(UInt32)	Set De-Skew delay for ATXSS device connections
∃	SetSamplingFrequency(Decimal, Decimal, ClockSource, Decimal)	Set sampling frequency for all pair
∃	SetupBNCTriggerOut(ChannelOutLogicOperation, ChannelOutLogicOperation, ChannelOutLogicOperation)	Set front panel trigger out FPGA combinations.
	STOP()	Stop instrument
	ToString()	Returns a <u>String</u> that represents the

current <u>Object</u>.

(Inherited from <u>Object</u>.)

■ Inheritance Hierarchy

Object

ATAWGBase

Device

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Device Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

Device()

C#		,
	4	-

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the Device class.

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public Device()</pre>			
Public Sub New			
<pre>public: Device()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Abort Pending Transfers Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

AbortPendingTransfers()

C# -

[This is preliminary documentation and is subject to change.]

Break all pending (NonReentrant) tranfers to the channels.

Declaration Syntax

C# Visual Basic Visual C++

public ATError AbortPendingTransfers()

Public Function AbortPendingTransfers As <u>ATError</u>

public:

ATError^ AbortPendingTransfers()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation AbortTransfers

Method (transferType)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

AbortTransfers(TransferTypeDef)

C# -

[This is preliminary documentation and is subject to change.]

Abort all pending and queued bulk transfers.

Declaration Syntax

■ Parameters

transferType (TransferTypeDef)

[Missing <param name="transferType"/> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.Device.Abo

■ Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>





ArbStudio Control Classes Library Documentation AWG Model Type

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

AWGModelType

C# _

[This is preliminary documentation and is subject to change.]

Get AWG model type

■ Declaration Syntax

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library

Documentation Close Completed Transfers Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

CloseCompletedTransfers()

C#

[This is preliminary documentation and is subject to change.]

Close all completed bulk data transfers.

Declaration Syntax

C# Visual Basic Visual C++

public ATError CloseCompletedTransfers()

Public Function CloseCompletedTransfers As ATError

public:

ATError^ CloseCompletedTransfers()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation DeviceUsbId

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

DeviceUsbId

C#

[This is preliminary documentation and is subject to change.]

Get USB device enumeration.

Declaration Syntax

```
C# Visual Basic Visual C++
public string DeviceUsbId { get; }

Public ReadOnly Property DeviceUsbId As String

public:
property String^ DeviceUsbId {
        String^ get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Disable FPT rigger Out Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

DisableFPTriggerOut()

C#	•
	4

[This is preliminary documentation and is subject to change.]

Disable front panel global trigger out.

Declaration Syntax

C# Visual Basic Visual C++

public ATError DisableFPTriggerOut()

Public Function DisableFPTriggerOut As ATError

public:

ATError^ DisableFPTriggerOut()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library

Documentation Enable FPT rigger Out Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

EnableFPTriggerOut()

C# -

[This is preliminary documentation and is subject to change.]

Enable front panel global trigger out.

Declaration Syntax

C# Visual Basic Visual C++

public ATError EnableFPTriggerOut()

Public Function EnableFPTriggerOut As ATError

public:

ATError^ EnableFPTriggerOut()

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation ForceStop Method

(channelId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

ForceStop(Byte[])

C# _

[This is preliminary documentation and is subject to change.]

Send a stop trigger to given channels

Declaration Syntax

■ Parameters

channelId (Byte [])

List of channels to be stopped

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation ForceTrigger

Method (channelId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

ForceTrigger(Byte[])

C# _

[This is preliminary documentation and is subject to change.]

Send a trigger to given channels

Declaration Syntax

■ Parameters

channelId (Byte [])

List of channels to be triggered

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation **GetChannel**

Method (channelId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

GetChannel(Int32)

C#

[This is preliminary documentation and is subject to change.]

Return device channel for the given id

Declaration Syntax

```
C# Visual Basic Visual C++

public Channel GetChannel(
        int channelId
)

Public Function GetChannel (
        channelId As Integer
) As Channel

public:
Channel^ GetChannel(
        int channelId
)
```

■ Parameters

channelId (Int32)

Id of channel to be returned (0..3)

■ Return Value

[Missing <returns> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.Device.GetChai

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation GetPatternGenerator Method

(patternId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

GetPatternGenerator(Int32)

C# -

[This is preliminary documentation and is subject to change.]

Return device pattern generator for the given id

Declaration Syntax

Parameters

patternId (Int32)

Id of pattern generator to be returned (0..1)

Return Value

[Missing <returns> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.Device.GetPatt

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

ArbStudio Control Classes Library Documentation GetTransferStatus

Method (transferId, status, LeCroy byteTransferred)

Namespaces ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

GetTransferStatus(Int32, TransferStatus, Int32)

C#

[This is preliminary documentation and is subject to change.]

Query the status of the (NonReentrant) transfers to the channels.

Declaration Syntax

```
Visual C++
C#
                   Visual Basic
public ATError GetTransferStatus(
        int transferId,
        out TransferStatus status,
        out int byteTransferred
)
Public Function GetTransferStatus (
        transferId As Integer,
        <OutAttribute> ByRef status As TransferStatus, _
        <OutAttribute> ByRef byteTransferred As Integer
) As <u>ATError</u>
public:
ATError^ GetTransferStatus(
        int transferId,
        [OutAttribute] TransferStatus% status,
        [OutAttribute] int% byteTransferred
```

Parameters

```
transferId (Int32)
```

Transfer id to be query.

status (TransferStatus)

out TransferStatus status of transfer.

byteTransferred (Int32)

out Number of bytes already transferred.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Initialize Method

(channelArray)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

Initialize(Functionality[])

C#

[This is preliminary documentation and is subject to change.]

Initialize device with creation of channels with given channel Functionality.

Declaration Syntax

Parameters

channelArray (Functionality [])

Array of 2 (AWG 2000 and AWGPower) or 4 (AWG 4000) elements with assigned Functionality.

¬ Return Value

ATError class result of operation.

Remarks

An error is returned if channel array dimensions is different from 2 or 4.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Installed Memory

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

InstalledMemory

C# -

[This is preliminary documentation and is subject to change.]

Get installed memory size of instrument

Declaration Syntax

```
C# Visual Basic Visual C++
public MemorySize InstalledMemory { get; }

Public ReadOnly Property InstalledMemory As MemorySize

public:
property MemorySize InstalledMemory {
          MemorySize get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PairLeft Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

PairLeft

C#	_	•

[This is preliminary documentation and is subject to change.]

Get or Set the left pair of channels

Declaration Syntax

```
C# Visual Basic Visual C++
public PairChannels PairLeft { get; set; }

Public Property PairLeft As PairChannels

public:
property PairChannels^ PairLeft {
    PairChannels^ get ();
    void set (PairChannels^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PairRight Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

PairRight

C# _-

[This is preliminary documentation and is subject to change.]

Get or Set the left pair of channels (only for AWG 4000)

Declaration Syntax

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation PowerLed Manage

Method (ledMode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

PowerLedManage(Byte)

C# _

[This is preliminary documentation and is subject to change.]

Control power led

Declaration Syntax

■ Parameters

ledMode (Byte)

0: off, 1: on, 2: blink

■ Return Value

[Missing <returns> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.Device.PowerL

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RUN Method

(channelId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

RUN(Byte[])

C#

[This is preliminary documentation and is subject to change.]

Run instrument

Declaration Syntax

■ Parameters

channelId (Byte [])

List of channel to be started

■ Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Serial Id Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

SerialId

C#

[This is preliminary documentation and is subject to change.]

Get Serial Id of connected device

Declaration Syntax

```
C# Visual Basic Visual C++

public string SerialId { get; set; }

Public Property SerialId As String

public:
property String^ SerialId {
    String^ get ();
    void set (String^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

<u>LeCroy</u>

ArbStudio Control Classes Library

Documentation Set ATXSSDe Skew Delay Method

(delay)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

SetATXSSDeSkewDelay(UInt32)

C# _-

[This is preliminary documentation and is subject to change.]

Set De-Skew delay for ATXSS device connections

Declaration Syntax

Parameters

delay (UInt32)

De-Skew delay expressed in pico seconds

Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Set Sampling Frequency Method (frequency Left Pair, frequency Right Pair, clock Type, frequency External Clock)

<u>Namespaces</u> ►

ArbStudio Control Classes Library

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

SetSamplingFrequency(Decimal, Decimal, ClockSource, Decimal)

C#

[This is preliminary documentation and is subject to change.]

Set sampling frequency for all pairs

Declaration Syntax

```
Visual C++
C#
                    Visual Basic
public ATError SetSamplingFrequency(
        ref <u>decimal</u> frequencyLeftPair,
        ref <u>decimal</u> frequencyRightPair,
        ClockSource clockType,
        decimal frequencyExternalClock
)
Public Function SetSamplingFrequency (
        ByRef frequencyLeftPair As Decimal,
        ByRef frequencyRightPair As Decimal,
        clockType As ClockSource,
        frequencyExternalClock As Decimal
) As <u>ATError</u>
public:
ATError^ SetSamplingFrequency(
        Decimal% frequencyLeftPair,
        Decimal% frequencyRightPair,
        ClockSource clockType,
```

Parameters

frequencyLeftPair (Decimal)

Decimal frequencyExternalClock

Frequency for left pair

frequencyRightPair (Decimal)

Frequency for right pair

clockType (ClockSource)

Clock type external or internal

frequencyExternalClock (Decimal)

Frequency for external clock

Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Setup BNCT rigger Out Method (leftOper, rightOper, globalOper)

Namespaces ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

SetupBNCTriggerOut(ChannelOutLogicOperation, ChannelOutLogicOperation, ChannelOutLogicOperation)

C#

[This is preliminary documentation and is subject to change.]

Set front panel trigger out FPGA combinations.

■ Declaration Syntax

```
C#
                   Visual Basic
                                      Visual C++
public ATError SetupBNCTriggerOut(
        ChannelOutLogicOperation leftOper,
        ChannelOutLogicOperation rightOper,
        ChannelOutLogicOperation globalOper
)
Public Function SetupBNCTriggerOut (
        leftOper As ChannelOutLogicOperation,
        rightOper As ChannelOutLogicOperation, _
        globalOper As ChannelOutLogicOperation
) As ATError
public:
ATError^ SetupBNCTriggerOut(
        ChannelOutLogicOperation LeftOper,
        ChannelOutLogicOperation rightOper,
        ChannelOutLogicOperation qlobalOper
)
```

Parameters

IeftOper (ChannelOutLogicOperation)

Logical operation of left pair channels.

rightOper (ChannelOutLogicOperation)

Logical operation of right pair channels.

globalOper (ChannelOutLogicOperation)

Logical operation of global front panel trigger out combination of left and right pairs. If at least one pair is pattern only SelectionAWG or SelectionPattern parameter is accepted. For all pattern parameter is ignored

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation STOP Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Device</u> ►

STOP()

C#	•
C 11	_

[This is preliminary documentation and is subject to change.]

Stop instrument

Declaration Syntax

C# Visual Basic Visual C++

public ATError STOP()

Public Function STOP As ATError

public: ATError^ STOP()

Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation DeviceSet Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► **DeviceSet**

C#

Get the list of usb connected devices

[This is preliminary documentation and is subject to change.]

Provide a class to manage all AWG4000 usb connectable devices

Declaration Syntax

Visual Basic Visual C++ C#

public class DeviceSet : ATAWGBase

Public Class DeviceSet Inherits ATAWGBase

public ref class DeviceSet : public ATAWGBase

DeviceList

	- Mei	mbers					
	All Me	embers	Constructors	Met	hods	Properties	
		✓ Instance ✓ Static		Declare Inherite			
	Icon	Member			Description	1	
	≘ڼ	DeviceSet()		Initialize class and generate connecte Device list			
	=	DeviceSet(Boolean)		Refresh connected devices list			
	≡	ATXSS_SetMasterChannel(Byte, Byte)		Set mode master for the given deviceId and channelId. If channelId 0 all channels set in slave mode.		annelId i	

₫••	Equals(Object)	Determines whether the specified Object is equal to the current Object .
		(Inherited from Object.)
	ErrorResult	Contain the result status of error.
		(Inherited from <u>ATAWGBase</u> .)
⊕	GetDeviceFromSerialId(String)	Return the Device object from serial id.
≓ ♦	GetHashCode()	Serves as a hash function for a particular type.
		(Inherited from Object.)
≡ ₩	GetType()	Gets the <u>Type</u> of the current instance (Inherited from <u>Object</u> .)
≓	ReorderDeviceList(Byte[])	Reorder device list upon new order lis
∃	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> .
		(Inherited from Object.)

■ Inheritance Hierarchy

Object

ATAWGBase

DeviceSet

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DeviceSet

Constructor

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► DeviceSet

▶ DeviceSet()

C#	-
<i>\C 11</i>	4

[This is preliminary documentation and is subject to change.]

Members

Icon	Member	Description
∄	DeviceSet()	Initialize class and generate connected Device list
= 	DeviceSet(Boolean)	Refresh connected devices list

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation DeviceSet

Constructor

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► DeviceSet

▶ DeviceSet()

C#

[This is preliminary documentation and is subject to change.]

Initialize class and generate connected Device list

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public DeviceSet()</pre>			
Public Sub New			
<pre>public: DeviceSet()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DeviceSet

Constructor (refreshDevices)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► DeviceSet

▶ DeviceSet(Boolean)

C#	_

[This is preliminary documentation and is subject to change.]

Refresh connected devices list

Declaration Syntax

Parameters

refreshDevices (Boolean)

false to reset device list, true for renew handles

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library

Documentation ATXSS_SetMasterChannel Method (deviceId, channelId)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>DeviceSet</u>

► ATXSS_SetMasterChannel(Byte, Byte)

C# _

[This is preliminary documentation and is subject to change.]

Set mode master for the given deviceId and channelId. If channelId is 0 all channels set in slave mode.

Declaration Syntax

```
C#
                    Visual Basic
                                         Visual C++
public ATError ATXSS SetMasterChannel(
        byte deviceId,
        byte channelId
)
Public Function ATXSS SetMasterChannel (
        deviceId As Byte, _
        channelId As <u>Byte</u>
) As <u>ATError</u>
public:
ATError^ ATXSS SetMasterChannel(
        unsigned char deviceId,
        unsigned char channelId
)
```

Parameters

deviceId (Byte)

0-based index identification of device

channelId (Byte)

Index channel of master, 0 for all slave

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation DeviceList

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>DeviceSet</u>

▶ DeviceList

C#

[This is preliminary documentation and is subject to change.]

Get the list of usb connected devices

■ Declaration Syntax

```
C# Visual Basic Visual C++
public List<Device> DeviceList { get; }

Public ReadOnly Property DeviceList As List(Of Device)

public:
property List<Device^>>^ DeviceList {
    List<Device^>>^ get ();
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Get Device From Serial Id Method

(serialtofind)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► DeviceSet

▶ GetDeviceFromSerialId(String)

C# _

[This is preliminary documentation and is subject to change.]

Return the Device object from serial id.

Declaration Syntax

Parameters

serialtofind (String)

Serial id of device to be found.

Return Value

Device object. Object is null if not found.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Reorder DeviceList

Method (devices)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>DeviceSet</u>

▶ ReorderDeviceList(Byte[])

C#	_

[This is preliminary documentation and is subject to change.]

Reorder device list upon new order list

Declaration Syntax

■ Parameters

devices (Byte [])

Array of new desire list

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation ErrorCodes Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ErrorCodes

C#	-

[This is preliminary documentation and is subject to change.]

Provide a class with AWG4000Control errod codes.

Declaration Syntax

C# Visual Basic Visual C++

public static class ErrorCodes

Public NotInheritable Class ErrorCodes

public ref class ErrorCodes abstract sealed

Members

All Members	Methods	Fields	
Public			□ Declared
		Static ■ Static	Inherited

Icon Member

E_CHANNEL_AMPLITUDE_PROFILE_NULL

E CHANNEL AMPLITUDE PROFILE SAMPLES ERROR

E CHANNEL BASE FREQUENCY OUT OF RANGE

♦ \$	E_CHANNEL_BULK_TRANSFER_STATUS_KO
⋄ S	E_CHANNEL_BULK_TRANSFER_STATUS_PENDING
⋄ \$	E_CHANNEL_CLOCK_TABLE_NULL
♦ \$	E_CHANNEL_DDSWAVEFORM_MARKERS_OUT_OF_BOUND
⋄ \$	E_CHANNEL_DDSWAVEFORM_MEMORY_DIVISION_NOT_SET
⋄ S	E_CHANNEL_DDSWAVEFORM_MEMORY_DIVISION_OUT_OF_BOUND
♦ S	E_CHANNEL_DDSWAVEFORM_MEMORY_NOT_FILLED
♦ 🕏	E_CHANNEL_FREQUENCY_PRESCALER_OUT_OF_RANGE

S E_CHANNEL_GENERATION_SEQUENCE_ID_OUT_OF_RANGE S E_CHANNEL_GENERATION_SEQUENCE_OUT_OF_RANGE \$\right\{\sigma}\right\} E CHANNEL GENERATION SEQUENCE REPET OUT OF RANGE \$\right\{\sigma}\right\} E CHANNEL INITIAL CONFIG ERROR S **E_CHANNEL_IS_MASTER** S E CHANNEL MODULATION ARB AMPLITUDE VALUE OUT OF RANGE \$\right\{\sigma}\right\} E CHANNEL MODULATION COUNT OUT OF RANGE S E CHANNEL MODULATION DEFINITION NULL

♦ S	E_CHANNEL_MODULATION_PHASE_OUT_OF_RANGE
⋄ \$	E_CHANNEL_MODULATION_REPETITION_OUT_OF_RANGE
⋄ S	E_CHANNEL_PLL_STATUS_NOT_LOCKED
⋄ S	E_CHANNEL_STATUS_BUSY
⋄ S	E_CHANNEL_TRIGGER_IN_SOURCE_NOT_VALID
⋄ S	E_CHANNEL_TRIGGER_MODE_NOT_VALID
§ S	E_CHANNEL_TRIGGER_OUT_OPERATION_NOT_VALID
♦ \$	E_CHANNEL_TRIGGER_OUT_SOURCE_NOT_VALID

♦ S	E_CHANNEL_UPLOAD_FIRMWARE_ERROR
♦ S	E_CHANNEL_VOLTAGE_OUT_OF_BOUNDS
⋄ S	E_CHANNEL_WAVEFORM_DEFINITION_NULL
⋄ S	E_CHANNEL_WAVEFORM_OUT_OF_MEMORY
⋄ S	E_DAC_AGC_GAIN_OUT_OF_RANGE
♦ S	E_DAC_AUX1_DATA_OUT_OF_RANGE
⋄ S	E_DAC_AUX2_DATA_OUT_OF_RANGE
⋄ \$	E_DAC_CLOCK_STATE_OUT_OF_RANGE



\$ \$	E_DAC_PLL_VCO_DIVIDE_RATIO_OUT_OF_RANGE
⋄ S	E_DAC_Q_GAIN_ADJUSTMNENT_OUT_OF_RANGE
•\$	E_DAC_SYNC_I_DELAY_OUT_OF_RANGE
♦ S	E_DAC_SYNC_I_RATIO_OUT_OF_RANGE
⋄ \$	E DAC SYNC I TIME MARGIN OUT OF RANGE
\$ S	E_DEVICE_ATXSS_CABLE_NOT_PRESENT
⋄ \$	E_DEVICE_CHANNEL_COUNT_ERROR
⋄ \$	E_DEVICE_CHANNEL_ID_ERROR

⋄ S	E_DEVICE_DESKEW_DELAY_ERROR
⋄ \$	E_DEVICE_FUNCTIONALITY_NOT_AVAILABLE
⋄ \$	E_DEVICE_ID_ERROR
⋄ S	E_DEVICE_INSTRUMENT_DATA_UNKNOWN
♦ \$	E_DEVICE_PATTERN_NOT_ALLOWED
⋄ S	E_DEVICE_PATTERN_NOT_AVAILABLE
⋄ \$	E_DEVICE_POD_PROBE_NOT_CONNECTED
♦ \$	E_DEVICE_RUNNING_STATE
⋄ S	E_DEVICE_STOP_STATE

⋄ S	E_DRIVER_BOOT_FIRMWARE_ERROR
⋄ \$	E_DRIVER_CLOCK_ERROR
♦ \$	E_GENERIC_APPLICATION_ERROR
♦ \$	E_GENERIC_EXCEPTION_ERROR
⋄ \$	E_PAIR_CHANNEL_NOT_ALIGNED
⋄ S	E PAIR CHANNEL PRESCALER OUT OF RANGE
⋄ S	E_PARAMETER_LIST_COUNT_ZERO
⋄ \$	E_PARAMETER_NULL
8	E_PARSER_BAD_FILE

S E PATTERN INVALID PARAMETER \$\right\{\sigma}\right\} E PATTERN MASTER IDCHANNEL ERROR \$\right\{\sigma}\right\} E SYSTEM DRIVER ERROR ErrorDescription(Int32) \$\right\{\sigma}\right\} **RES APPLICATION** S **RES SUCCESS** S **RES SYSTEM** \$\right\{\sigma}\right\} **RES WARNING**

■ Inheritance Hierarchy

Object

ErrorCodes

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E_CHANNEL_AMPLITUDE_PROFILE_

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control ErrorCodes ► E_CHANNEL_AMPLITUDE_PROFILE_NULL

C#

[This is preliminary documentation and is subject to change.]

Channel amplitude profile is null.

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public const int E_CHANNEL_AMPLITUDE_PROFILE_NULL</pre>			
Public Const E_CHANNEL_AMPLITUDE_PROFILE_NULL As Integer			
<pre>public: literal <u>int</u> E_CHAN</pre>	INEL_AMPLITUDE_PROFI	LE_NULL	

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL AMPLITUDE PROFILE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E CHANNEL AMPLITUDE PROFILE SAMPLES ERROR

[This is preliminary documentation and is subject to change.]

Number of samples for amplitude profile intermediate segment error

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL AMPLITUDE PROFILE SAMPLES ERROR Public Const E CHANNEL AMPLITUDE PROFILE SAMPLES ERROR As Integer

public: literal int E_CHANNEL_AMPLITUDE_PROFILE_SAMPLES_ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL BASE FREQUENCY OL

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E CHANNEL BASE FREQUENCY OUT OF RANGE

[This is preliminary documentation and is subject to change.]

Carrier base frequency is out of bounds (125/2^36...125/2).

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL BASE FREQUENCY OUT OF RANGE Public Const E CHANNEL BASE FREQUENCY OUT OF RANGE As Integer

public: literal int E CHANNEL BASE FREQUENCY OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_BULK_TRANSFER_STA

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control <u>ErrorCodes</u> ► E_CHANNEL_BULK_TRANSFER_STATUS_KO

C

[This is preliminary documentation and is subject to change.]

Bulk transfer to FPGA has failed.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_CHANNEL_BULK_TRANSFER_STATUS_KO

Public Const E_CHANNEL_BULK_TRANSFER_STATUS_KO As Integer

public:
literal int E_CHANNEL_BULK_TRANSFER_STATUS_KO

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E CHANNEL BULK TRANSFER STA

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E CHANNEL BULK TRANSFER STATUS PENDING

[This is preliminary documentation and is subject to change.]

Bulk transfer to FPGA still pending.

Declaration Syntax

C# Visual Basic Visual C++

public const int E CHANNEL BULK TRANSFER STATUS PENDING

Public Const E CHANNEL BULK TRANSFER STATUS PENDING As Integer

public:

literal int E_CHANNEL_BULK_TRANSFER_STATUS_PENDING

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_CLOCK_TABLE_NULL

Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_CHANNEL_CLOCK_TABLE_NULL

C# _

[This is preliminary documentation and is subject to change.]

Clock table for FPGA is null.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_CHANNEL_CLOCK_TABLE_NULL

Public Const E_CHANNEL_CLOCK_TABLE_NULL As Integer

public:

literal int E_CHANNEL_CLOCK_TABLE_NULL

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



Documentation E CHANNEL DDSWAVEFORM MARI

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_DDSWAVEFORM_MARKERS_OUT_OF_BOUND

[This is preliminary documentation and is subject to change.]

Waveform markers in DDS mode must be max 1.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL DDSWAVEFORM MARKERS OUT OF BOUND

Public Const E CHANNEL DDSWAVEFORM MARKERS OUT OF BOUND As Integer

public:

literal int E CHANNEL DDSWAVEFORM MARKERS OUT OF BOUND

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL DDSWAVEFORM MEMO

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

E_CHANNEL_DDSWAVEFORM_MEMORY_DIVISION_NOT_SE

[This is preliminary documentation and is subject to change.]

Waveform DDS memory division mode is not set.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL DDSWAVEFORM MEMORY DIVISION NOT SET Public Const E CHANNEL DDSWAVEFORM MEMORY DIVISION NOT SET As Integer public: literal int E CHANNEL DDSWAVEFORM MEMORY DIVISION NOT SET

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL DDSWAVEFORM MEMO

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

E_CHANNEL_DDSWAVEFORM_MEMORY_DIVISION_OUT_OF

[This is preliminary documentation and is subject to change.]

Waveform DDS memory division mode is out of bounds (0...7).

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL DDSWAVEFORM MEMORY DIVISION OUT OF BOUND Public Const E CHANNEL DDSWAVEFORM MEMORY DIVISION OUT OF BOUND As Inte public: literal int E CHANNEL DDSWAVEFORM MEMORY DIVISION OUT OF BOUND

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL DDSWAVEFORM MEMO

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_DDSWAVEFORM_MEMORY_NOT_FILLED

[This is preliminary documentation and is subject to change.]

DDS waveform must be fill all memory available

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL DDSWAVEFORM MEMORY NOT FILLED

Public Const E CHANNEL DDSWAVEFORM MEMORY NOT FILLED As Integer

public:

literal int E CHANNEL DDSWAVEFORM MEMORY NOT FILLED

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_FREQUENCY_PRESCAL

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_FREQUENCY_PRESCALER_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Channel frequency prescaler is out of bounds (1...16383)

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL FREQUENCY PRESCALER OUT OF RANGE

Public Const E CHANNEL FREQUENCY PRESCALER OUT OF RANGE As Integer

public: literal int E_CHANNEL_FREQUENCY_PRESCALER_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_GENERATION_SEQUEN

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_GENERATION_SEQUENCE_ID_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Index of waveform is out of bounds (1...511)

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL GENERATION SEQUENCE ID OUT OF RANGE Public Const E CHANNEL GENERATION SEQUENCE ID OUT OF RANGE As Integer public: literal int E_CHANNEL_GENERATION_SEQUENCE_ID_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_GENERATION_SEQUEN

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_GENERATION_SEQUENCE_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Channel generation sequence is null or empty.

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL GENERATION SEQUENCE OUT OF RANGE

Public Const E CHANNEL GENERATION SEQUENCE OUT OF RANGE As Integer

public:

literal int E_CHANNEL_GENERATION_SEQUENCE_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_GENERATION_SEQUEN

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_GENERATION_SEQUENCE_REPET_OUT_OF_RA

[This is preliminary documentation and is subject to change.]

Number of repetitions for waveform is out of bounds $(1...2^33)$.

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL GENERATION SEQUENCE REPET OUT OF RANGE Public Const E CHANNEL GENERATION SEQUENCE REPET OUT OF RANGE As Intege public: literal <u>int</u> E CHANNEL GENERATION SEQUENCE REPET OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_INITIAL_CONFIG_ERR

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control ErrorCodes ► E_CHANNEL_INITIAL_CONFIG_ERROR

C#

[This is preliminary documentation and is subject to change.]

Unable to set FPGA for initial configuration.

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public const int E_CHANNEL_INITIAL_CONFIG_ERROR</pre>			
Public Const E_CHANNEL_INITIAL_CONFIG_ERROR As Integer			
<pre>public: literal <u>int</u> E_CHAN</pre>	NEL_INITIAL_CONFIG_	ERROR	

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_IS_MASTER Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_CHANNEL_IS_MASTER

C#

[This is preliminary documentation and is subject to change.]

Channel is master. Can't receive AT-XSS bus signals.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_CHANNEL_IS_MASTER

Public Const E_CHANNEL_IS_MASTER As Integer

public:
literal int E_CHANNEL_IS_MASTER

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E_CHANNEL_MODULATION_ARB_A

Field

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

E_CHANNEL_MODULATION_ARB_AMPLITUDE_VALUE_OUT_

[This is preliminary documentation and is subject to change.]

Arbitrary Amplitude modulation definition is out of range (0..1).

Declaration Syntax

C# Visual Basic Visual C++
public const int E_CHANNEL_MODULATION_ARB_AMPLITUDE_VALUE_OUT_OF_RANGE
Public Const E_CHANNEL_MODULATION_ARB_AMPLITUDE_VALUE_OUT_OF_RANGE As]
public:
literal int E_CHANNEL_MODULATION_ARB_AMPLITUDE_VALUE_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E CHANNEL MODULATION COUNT

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_MODULATION_COUNT_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Number of repetitions for sample are out of bounds (1...4095).

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL MODULATION COUNT OUT OF RANGE

Public Const E CHANNEL MODULATION COUNT OUT OF RANGE As Integer

public: literal int E CHANNEL MODULATION COUNT OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL MODULATION DEFINI

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_MODULATION_DEFINITION_NULL

[This is preliminary documentation and is subject to change.]

Amplitude modulation definition is null or empty.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL MODULATION DEFINITION NULL

Public Const E CHANNEL MODULATION DEFINITION NULL As Integer

public:

literal int E CHANNEL MODULATION DEFINITION NULL

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL MODULATION PHASE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_MODULATION_PHASE_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Phase modulation is out of bounds (0...360).

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL MODULATION PHASE OUT OF RANGE

Public Const E CHANNEL MODULATION PHASE OUT OF RANGE As Integer

public: literal int E CHANNEL MODULATION PHASE OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL MODULATION REPETI

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_MODULATION_REPETITION_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Number of repetitions is out of bounds $(0...2^20)$.

Declaration Syntax

Visual Basic C# Visual C++

public const int E CHANNEL MODULATION REPETITION OUT OF RANGE

Public Const E CHANNEL MODULATION REPETITION OUT OF RANGE As Integer

public:

literal int E CHANNEL MODULATION REPETITION OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_PLL_STATUS_NOT_LO

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E_CHANNEL_PLL_STATUS_NOT_LOCKED

C#

[This is preliminary documentation and is subject to change.]

PLL status is not locked.

Declaration Syntax

C#	Visual Basic	Visual C++		
public const <u>int</u> E	_CHANNEL_PLL_STATUS_	_NOT_LOCKED		
Public Const E_CHANNEL_PLL_STATUS_NOT_LOCKED As Integer				
<pre>public: literal <u>int</u> E_CHANNEL_PLL_STATUS_NOT_LOCKED</pre>				

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_STATUS_BUSY Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_CHANNEL_STATUS_BUSY

C# _

[This is preliminary documentation and is subject to change.]

Channel in busy status. Operation not allowed.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_CHANNEL_STATUS_BUSY

Public Const E_CHANNEL_STATUS_BUSY As Integer

public:

literal <u>int</u> E CHANNEL STATUS BUSY

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E CHANNEL TRIGGER IN SOURCE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_TRIGGER_IN_SOURCE_NOT_VALID

[This is preliminary documentation and is subject to change.]

Channel trigger in source are not allowed.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL TRIGGER IN SOURCE NOT VALID Public Const E CHANNEL TRIGGER IN SOURCE NOT VALID As Integer public: literal int E_CHANNEL_TRIGGER_IN_SOURCE_NOT_VALID

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_TRIGGER_MODE_NOT_

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control **ErrorCodes** ► **E_CHANNEL_TRIGGER_MODE_NOT_VALID**

C#

[This is preliminary documentation and is subject to change.]

Trigger mode not allowed for this channel.

Declaration Syntax

C#	Visual Basic	Visual C++			
public const <u>int</u> E	<pre>public const int E_CHANNEL_TRIGGER_MODE_NOT_VALID</pre>				
Public Const E_CHANNEL_TRIGGER_MODE_NOT_VALID As Integer					
<pre>public: literal int E_CHANNEL_TRIGGER_MODE_NOT_VALID</pre>					

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL TRIGGER OUT OPERA

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_TRIGGER_OUT_OPERATION_NOT_VALID

[This is preliminary documentation and is subject to change.]

Trigger out operation type not allowed

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL TRIGGER OUT OPERATION NOT VALID Public Const E CHANNEL TRIGGER OUT OPERATION NOT VALID As Integer public: literal int E CHANNEL TRIGGER OUT OPERATION NOT VALID

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL TRIGGER OUT SOURCE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_CHANNEL_TRIGGER_OUT_SOURCE_NOT_VALID

[This is preliminary documentation and is subject to change.]

Channel trigger out source are not allowed.

Declaration Syntax

Visual Basic C# Visual C++ public const int E CHANNEL TRIGGER OUT SOURCE NOT VALID Public Const E CHANNEL TRIGGER OUT SOURCE NOT VALID As Integer public: literal int E_CHANNEL_TRIGGER_OUT_SOURCE_NOT_VALID

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_UPLOAD_FIRMWARE_

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_CHANNEL_UPLOAD_FIRMWARE_ERROR**

C#

[This is preliminary documentation and is subject to change.]

Unable to upload firmware to FPGA.

Declaration Syntax

C#	Visual Basic	Visual C++		
<pre>public const int E_CHANNEL_UPLOAD_FIRMWARE_ERROR</pre>				
Public Const E_CHANNEL_UPLOAD_FIRMWARE_ERROR As <u>Integer</u>				
<pre>public: literal int E_CHANNEL_UPLOAD_FIRMWARE_ERROR</pre>				

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL VOLTAGE OUT OF BC

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control ErrorCodes ► E_CHANNEL_VOLTAGE_OUT_OF_BOUNDS

C#

[This is preliminary documentation and is subject to change.]

Channel output voltage is out of bounds (-6...+6).

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL VOLTAGE OUT OF BOUNDS Public Const E CHANNEL VOLTAGE OUT OF BOUNDS As Integer public: literal int E CHANNEL VOLTAGE OUT OF BOUNDS

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_CHANNEL_WAVEFORM_DEFINITI

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► E_CHANNEL_WAVEFORM_DEFINITION_NULL

[This is preliminary documentation and is subject to change.]

Channel waveform definition is null or empty.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL WAVEFORM DEFINITION NULL

Public Const E CHANNEL WAVEFORM DEFINITION NULL As Integer

public:

literal int E CHANNEL WAVEFORM DEFINITION NULL

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E CHANNEL WAVEFORM OUT OF

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► E_CHANNEL_WAVEFORM_OUT_OF_MEMORY

[This is preliminary documentation and is subject to change.]

Not enough memory to load waveform on channel.

Declaration Syntax

C# Visual Basic Visual C++ public const int E CHANNEL WAVEFORM OUT OF MEMORY

Public Const E CHANNEL WAVEFORM OUT OF MEMORY As Integer

public: literal int E CHANNEL WAVEFORM OUT OF MEMORY

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DAC_AGC_GAIN_OUT_OF_RANGI

Field

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

► <u>ErrorCodes</u> ► **E_DAC_AGC_GAIN_OUT_OF_RANGE**

C#

[This is preliminary documentation and is subject to change.]

AGC gain is out of range (0...3).

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DAC_AGC_GAIN_OUT_OF_RANGE

Public Const E_DAC_AGC_GAIN_OUT_OF_RANGE As Integer

public:
literal int E_DAC_AGC_GAIN_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E DAC AUX1 DATA OUT OF RANGE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control ErrorCodes ► E DAC AUX1 DATA OUT OF RANGE

Visual C++

C#

[This is preliminary documentation and is subject to change.]

Auxiliary DAC1 data out of range (0...1023).

Declaration Syntax

C# Visual Basic public const int E DAC AUX1 DATA OUT OF RANGE Public Const E DAC AUX1 DATA OUT OF RANGE As Integer

public: literal int E DAC AUX1 DATA OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC AUX2 DATA OUT OF RANGE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E DAC AUX2 DATA OUT OF RANGE

C#

[This is preliminary documentation and is subject to change.]

Auxiliary DAC2 data out of range (0...1023).

Declaration Syntax

C# Visual Basic Visual C++ public const int E DAC AUX2 DATA OUT OF RANGE

Public Const E DAC AUX2 DATA OUT OF RANGE As Integer

public: literal int E DAC AUX2 DATA OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC CLOCK STATE OUT OF RA

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_DAC_CLOCK_STATE_OUT_OF_RANGE**

C#

[This is preliminary documentation and is subject to change.]

CLOCK_STATE is out of range (0...31).

Declaration Syntax

C# Visual Basic Visual C++

public const int E DAC CLOCK STATE OUT OF RANGE

Public Const E DAC CLOCK STATE OUT OF RANGE As Integer

public:

literal int E DAC CLOCK STATE OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC DATA FORMAT OUT OF RA

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_DAC_DATA_FORMAT_OUT_OF_RANGE**

C#

[This is preliminary documentation and is subject to change.]

Data format is out of range (0...1).

Declaration Syntax

C# Visual Basic Visual C++ public const int E DAC DATA FORMAT OUT OF RANGE

Public Const E DAC DATA FORMAT OUT OF RANGE As Integer

public: literal int E DAC DATA FORMAT OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC DATA TIME MARGIN OUT

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E DAC DATA TIME MARGIN OUT OF RANGE

[This is preliminary documentation and is subject to change.]

Data time margin is out of range (0...15).

Declaration Syntax

Visual Basic C# Visual C++ public const int E DAC DATA TIME MARGIN OUT OF RANGE

Public Const E DAC DATA TIME MARGIN OUT OF RANGE As Integer

public: literal int E DAC DATA TIME MARGIN OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC I GAIN ADJUSTMNENT OU

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E DAC I GAIN ADJUSTMNENT OUT OF RANGE

[This is preliminary documentation and is subject to change.]

I DAC gain adjustment out of range (0...1023).

Declaration Syntax

Visual Basic C# Visual C++ public const int E DAC I GAIN ADJUSTMNENT OUT OF RANGE

Public Const E DAC I GAIN ADJUSTMNENT OUT OF RANGE As Integer

public: literal int E DAC I GAIN ADJUSTMNENT OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DAC_INTERPOLATION_FACTOR

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_DAC_INTERPOLATION_FACTOR_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Interpolation factor is out of range (0...3).

Declaration Syntax

C# Visual Basic Visual C++ public const int E DAC INTERPOLATION FACTOR OUT OF RANGE

Public Const E DAC INTERPOLATION FACTOR OUT OF RANGE As Integer

public:

literal int E_DAC_INTERPOLATION_FACTOR_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DAC_INTERPOLATION MODE OL

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_DAC_INTERPOLATION_MODE_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Interpolation mode is out of range (0...15).

Declaration Syntax

Visual Basic C# Visual C++ public const int E DAC INTERPOLATION MODE OUT OF RANGE

Public Const E DAC INTERPOLATION MODE OUT OF RANGE As Integer

public: literal int E DAC INTERPOLATION MODE OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DAC_PLL_BAND_SELECT_OUT_O

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► E_DAC_PLL_BAND_SELECT_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

PLL band select out of range (0...63).

Declaration Syntax

C# Visual Basic Visual C++
public const int E_DAC_PLL_BAND_SELECT_OUT_OF_RANGE

Public Const E_DAC_PLL_BAND_SELECT_OUT_OF_RANGE As Integer

public: literal int E_DAC_PLL_BAND_SELECT_OUT_OF_RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation E_DAC_PLL_BIAS_OUT_OF_RANGE

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► <u>ErrorCodes</u> ► **E_DAC_PLL_BIAS_OUT_OF_RANGE**

C#

[This is preliminary documentation and is subject to change.]

PLL bias out of range (0...7).

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DAC_PLL_BIAS_OUT_OF_RANGE

Public Const E_DAC_PLL_BIAS_OUT_OF_RANGE As Integer

public:
literal int E DAC PLL BIAS OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



C#

ArbStudio Control Classes Library

Documentation E DAC PLL LOOP BANDWIDTH O

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

Visual C++

E DAC PLL LOOP BANDWIDTH OUT OF RANGE

[This is preliminary documentation and is subject to change.]

PLL loop bandwidth out of range (0...31).

Declaration Syntax

public const int E DAC PLL LOOP BANDWIDTH OUT OF RANGE Public Const E DAC PLL LOOP BANDWIDTH OUT OF RANGE As Integer

public: literal int E DAC PLL LOOP BANDWIDTH OUT OF RANGE

Visual Basic

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC PLL LOOP DIVIDE RATIO

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_DAC_PLL_LOOP_DIVIDE_RATIO_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

PLL loop divide ratio out of range (0...3).

Declaration Syntax

Visual Basic C# Visual C++ public const int E DAC PLL LOOP DIVIDE RATIO OUT OF RANGE

Public Const E DAC PLL LOOP DIVIDE RATIO OUT OF RANGE As Integer

public: literal int E DAC PLL LOOP DIVIDE RATIO OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DAC_PLL_VCO_DIVIDE_RATIO_C

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_DAC_PLL_VCO_DIVIDE_RATIO_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

PLL VCO divide ratio out of range (0...3).

Declaration Syntax

Visual Basic C# Visual C++ public const int E DAC PLL VCO DIVIDE RATIO OUT OF RANGE Public Const E DAC PLL VCO DIVIDE RATIO OUT OF RANGE As Integer public:

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved

literal int E DAC PLL VCO DIVIDE RATIO OUT OF RANGE



Documentation E DAC Q GAIN ADJUSTMNENT OI

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E DAC Q GAIN ADJUSTMNENT OUT OF RANGE

[This is preliminary documentation and is subject to change.]

Q DAC gain adjustment out of range (0...1023).

Declaration Syntax

Visual Basic C# Visual C++

public const int E DAC Q GAIN ADJUSTMNENT OUT OF RANGE

Public Const E DAC Q GAIN ADJUSTMNENT OUT_OF_RANGE As Integer

public:

literal int E DAC Q GAIN ADJUSTMNENT OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DAC SYNC I DELAY OUT OF R.

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E_DAC_SYNC_I_DELAY_OUT_OF_RANGE

C#

[This is preliminary documentation and is subject to change.]

SYNC_I delay is out of range (0...31).

Declaration Syntax

C# Visual Basic Visual C++ public const int E DAC SYNC I DELAY OUT OF RANGE Public Const E DAC SYNC I DELAY OUT OF RANGE As Integer public: literal int E DAC SYNC I DELAY OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



C#

ArbStudio Control Classes Library

Documentation E DAC SYNC I RATIO OUT OF RA

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

Visual C++

ErrorCodes ► E_DAC_SYNC_I_RATIO_OUT_OF_RANGE

C#

[This is preliminary documentation and is subject to change.]

SYNC I ratio is out of range (0...7).

Declaration Syntax

Visual Basic public const int E DAC SYNC I RATIO OUT OF RANGE Public Const E DAC SYNC I RATIO OUT OF RANGE As Integer public:

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved

literal int E DAC SYNC I RATIO OUT OF RANGE



C#

ArbStudio Control Classes Library

Documentation E DAC SYNC I TIME MARGIN OU

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

Visual C++

E_DAC_SYNC_I_TIME_MARGIN_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

SYNC_I time margin is out of range (0...15).

Declaration Syntax

Visual Basic public const int E DAC SYNC I TIME MARGIN OUT OF RANGE Public Const E DAC SYNC I TIME MARGIN OUT OF RANGE As Integer public:

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved

literal int E DAC SYNC I TIME MARGIN OUT OF RANGE



Documentation E_DEVICE_ATXSS_CABLE_NOT_PRE

Field

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

ErrorCodes ► E_DEVICE_ATXSS_CABLE_NOT_PRESENT

C#

[This is preliminary documentation and is subject to change.]

ATXSS cable not connected

Declaration Syntax

C#	Visual Basic	Visual C++			
<pre>public const int E_DEVICE_ATXSS_CABLE_NOT_PRESENT</pre>					
Public Const E_DEVICE_ATXSS_CABLE_NOT_PRESENT As Integer					
<pre>public: literal int E</pre>	DEVICE ATXSS CABLE NOT	PRESENT			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E DEVICE CHANNEL COUNT ERRO

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E DEVICE CHANNEL COUNT ERROR

C#

[This is preliminary documentation and is subject to change.]

Number of channels for device is not valid.

Declaration Syntax

C# Visual Basic Visual C++ public const int E DEVICE CHANNEL COUNT ERROR Public Const E DEVICE CHANNEL COUNT ERROR As Integer public: literal int E DEVICE CHANNEL COUNT ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DEVICE_CHANNEL_ID_ERROR

Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_DEVICE_CHANNEL_ID_ERROR

C# -

[This is preliminary documentation and is subject to change.]

Id of channel is out of range 1...4 (1...2 for AWG 2000).

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DEVICE_CHANNEL_ID_ERROR

Public Const E_DEVICE_CHANNEL_ID_ERROR As Integer

public:

literal <u>int</u> E_DEVICE_CHANNEL_ID_ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



Documentation E_DEVICE_DESKEW_DELAY_ERROR

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► <u>ErrorCodes</u> ► <u>E_DEVICE_DESKEW_DELAY_ERROR</u>

C# __

[This is preliminary documentation and is subject to change.]

DeSkew delay not allowed or out of range

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DEVICE_DESKEW_DELAY_ERROR

Public Const E_DEVICE_DESKEW_DELAY_ERROR As Integer

public:
literal int E_DEVICE_DESKEW_DELAY_ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E_DEVICE_FUNCTIONALITY_NOT_A

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E DEVICE FUNCTIONALITY NOT AVAILABLE

[This is preliminary documentation and is subject to change.]

Channel functionality not available

Declaration Syntax

C# Visual Basic Visual C++

public const int E DEVICE FUNCTIONALITY NOT AVAILABLE

Public Const E DEVICE FUNCTIONALITY NOT AVAILABLE As Integer

public:

literal <u>int</u> E DEVICE FUNCTIONALITY NOT AVAILABLE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DEVICE ID ERROR Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_DEVICE_ID_ERROR

C# _

[This is preliminary documentation and is subject to change.]

Device not found for given id.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DEVICE_ID_ERROR

Public Const E_DEVICE_ID_ERROR As Integer

public:
literal int E DEVICE ID ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E DEVICE INSTRUMENT DATA UN

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control ErrorCodes ► E DEVICE INSTRUMENT DATA UNKNOWN

C:

[This is preliminary documentation and is subject to change.]

Instrument not recognized or data missing

Declaration Syntax

Visual Basic C# Visual C++ public const int E DEVICE INSTRUMENT DATA UNKNOWN Public Const E DEVICE INSTRUMENT DATA UNKNOWN As Integer public: literal int E DEVICE INSTRUMENT DATA UNKNOWN

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DEVICE PATTERN NOT ALLOWE

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_DEVICE_PATTERN_NOT_ALLOWED**

C#

[This is preliminary documentation and is subject to change.]

Pattern functionality not allowed in channel position

Declaration Syntax

C# Visual Basic Visual C++ public const int E DEVICE PATTERN NOT ALLOWED

Public Const E DEVICE PATTERN NOT ALLOWED As Integer

public: literal int E DEVICE PATTERN NOT ALLOWED

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DEVICE PATTERN NOT AVAILAB

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E DEVICE PATTERN NOT AVAILABLE

C#

[This is preliminary documentation and is subject to change.]

Pattern functionality not available

Declaration Syntax

C# Visual Basic Visual C++ public const int E DEVICE PATTERN NOT AVAILABLE

Public Const E DEVICE PATTERN NOT AVAILABLE As Integer

public: literal int E DEVICE PATTERN NOT AVAILABLE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DEVICE_POD_PROBE_NOT_CONN

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_DEVICE_POD_PROBE_NOT_CONNECTED**

C#

[This is preliminary documentation and is subject to change.]

Pod probe not connected

Declaration Syntax

C#	Visual Basic	Visual C++			
<pre>public const int E_DEVICE_POD_PROBE_NOT_CONNECTED</pre>					
Public Const E_DEVICE_POD_PROBE_NOT_CONNECTED As Integer					
<pre>public: literal int E_DEVICE_POD_PROBE_NOT_CONNECTED</pre>					

Send comments on this topic to LeCroy Corporation Support Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_DEVICE_RUNNING_STATE Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_DEVICE_RUNNING_STATE

C#

[This is preliminary documentation and is subject to change.]

Device in running state. Functionality not available

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DEVICE_RUNNING_STATE

Public Const E_DEVICE_RUNNING_STATE As Integer

public:

literal <u>int</u> E DEVICE RUNNING STATE

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E DEVICE STOP STATE Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_DEVICE_STOP_STATE

C# -

[This is preliminary documentation and is subject to change.]

Device in stop state. Functionality not available

Declaration Syntax

C# Visual Basic Visual C++
public const int E_DEVICE_STOP_STATE

Public Const E_DEVICE_STOP_STATE As Integer

public: literal int E_DEVICE_STOP_STATE

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E DRIVER BOOT FIRMWARE ERRO

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► E DRIVER BOOT FIRMWARE ERROR

C#

[This is preliminary documentation and is subject to change.]

Boot firmware upload has failed.

Declaration Syntax

Visual Basic C# Visual C++ public const int E DRIVER BOOT FIRMWARE ERROR Public Const E DRIVER BOOT FIRMWARE ERROR As Integer public: literal int E DRIVER BOOT FIRMWARE ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E DRIVER CLOCK ERROR Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_DRIVER_CLOCK_ERROR

C#

[This is preliminary documentation and is subject to change.]

System driver clock error.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_DRIVER_CLOCK_ERROR

Public Const E_DRIVER_CLOCK_ERROR As Integer

public:
literal int E_DRIVER_CLOCK_ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E_GENERIC_APPLICATION_ERROR

Field

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

► <u>ErrorCodes</u> ► <u>E_GENERIC_APPLICATION_ERROR</u>

C#

[This is preliminary documentation and is subject to change.]

Generic application error.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_GENERIC_APPLICATION_ERROR

Public Const E_GENERIC_APPLICATION_ERROR As Integer

public:
literal int E GENERIC APPLICATION ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation E GENERIC EXCEPTION ERROR

Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_GENERIC_EXCEPTION_ERROR

C#

[This is preliminary documentation and is subject to change.]

Generic exception error.

■ Declaration Syntax

C# Visual Basic Visual C++

public const int E_GENERIC_EXCEPTION_ERROR

Public Const E_GENERIC_EXCEPTION_ERROR As Integer

public:

literal <u>int</u> E_GENERIC_EXCEPTION_ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E PAIR CHANNEL NOT ALIGNED

Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_PAIR_CHANNEL_NOT_ALIGNED

C# _

[This is preliminary documentation and is subject to change.]

Pair channel communication not aligned

Declaration Syntax

C# Visual Basic Visual C++

public const int E_PAIR_CHANNEL_NOT_ALIGNED

Public Const E_PAIR_CHANNEL_NOT_ALIGNED As Integer

public:

literal <u>int</u> E_PAIR_CHANNEL_NOT_ALIGNED

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



Documentation E PAIR CHANNEL PRESCALER OU

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

E_PAIR_CHANNEL_PRESCALER_OUT_OF_RANGE

[This is preliminary documentation and is subject to change.]

Pair channel prescaler out of range (0...2^23)

Declaration Syntax

Visual Basic C# Visual C++ public const int E PAIR CHANNEL PRESCALER OUT OF RANGE

Public Const E PAIR CHANNEL PRESCALER OUT OF RANGE As Integer

public: literal int E PAIR CHANNEL PRESCALER OUT OF RANGE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_PARAMETER_LIST_COUNT_ZERO

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► <u>ErrorCodes</u> ► <u>E_PARAMETER_LIST_COUNT_ZERO</u>

C#

[This is preliminary documentation and is subject to change.]

Parameter list has no elements.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_PARAMETER_LIST_COUNT_ZERO

Public Const E_PARAMETER_LIST_COUNT_ZERO As Integer

public:
literal int E_PARAMETER_LIST_COUNT_ZERO

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation E_PARAMETER_NULL Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_PARAMETER_NULL

C# _

[This is preliminary documentation and is subject to change.]

Parameter null

■ Declaration Syntax

C# Visual Basic Visual C++

public const int E_PARAMETER_NULL

Public Const E_PARAMETER_NULL As Integer

public:

literal int E_PARAMETER_NULL

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation E_PARSER_BAD_FILE Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_PARSER_BAD_FILE

C#

[This is preliminary documentation and is subject to change.]

Bad File

■ Declaration Syntax

C# Visual Basic Visual C++

public const int E_PARSER_BAD_FILE

Public Const E_PARSER_BAD_FILE As Integer

public:

literal int E_PARSER_BAD_FILE

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation E_PATTERN_INVALID_PARAMETER

Field

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

► <u>ErrorCodes</u> ► <u>E_PATTERN_INVALID_PARAMETER</u>

C# <u></u>▼

[This is preliminary documentation and is subject to change.]

Invalid Parameter

Declaration Syntax

C# Visual Basic Visual C++

public const int E_PATTERN_INVALID_PARAMETER

Public Const E_PATTERN_INVALID_PARAMETER As Integer

public:
literal int E_PATTERN_INVALID_PARAMETER

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation E PATTERN MASTER IDCHANNEL

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control

ErrorCodes ► **E_PATTERN_MASTER_IDCHANNEL_ERROR**

C#

[This is preliminary documentation and is subject to change.]

Master channel id error

Declaration Syntax

C#	Visual Basic	Visual C++			
public const <u>int</u> E	<pre>public const int E_PATTERN_MASTER_IDCHANNEL_ERROR</pre>				
Public Const E_PATTERN_MASTER_IDCHANNEL_ERROR As Integer					
<pre>public: literal <u>int</u> E_PATTERN_MASTER_IDCHANNEL_ERROR</pre>					

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation E_SYSTEM_DRIVER_ERROR Field

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► E_SYSTEM_DRIVER_ERROR

C#

[This is preliminary documentation and is subject to change.]

Generic system error.

Declaration Syntax

C# Visual Basic Visual C++

public const int E_SYSTEM_DRIVER_ERROR

Public Const E_SYSTEM_DRIVER_ERROR As Integer

public:

literal <u>int</u> E SYSTEM DRIVER ERROR

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Error Description

Method (errorCode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>ErrorCodes</u>

► ErrorDescription(Int32)

^#	•
U 11	4

[This is preliminary documentation and is subject to change.]

Get error description on a given error code.

Declaration Syntax

```
C# Visual Basic Visual C++
public static string ErrorDescription(
        int errorCode
)

Public Shared Function ErrorDescription ( __
        errorCode As Integer __
) As String

public:
static String^ ErrorDescription(
        int errorCode
)
```

■ Parameters

errorCode (Int32)

Error code to be explained.

■ Return Value

String with error description.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RES_APPLICATION

Field

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► ErrorCodes

► RES_APPLICATION

C#	_
l .	4

[This is preliminary documentation and is subject to change.]

Client error recoverable.

Declaration Syntax

C# Visual Basic Visual C++

public const int RES_APPLICATION

Public Const RES_APPLICATION As Integer

public:

literal <u>int</u> RES APPLICATION

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RES_SUCCESS

Field

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► ErrorCodes

► RES_SUCCESS

C# _

[This is preliminary documentation and is subject to change.]

No error is present.

Declaration Syntax

C# Visual Basic Visual C++

public const int RES_SUCCESS

Public Const RES_SUCCESS As Integer

public:

literal int RES_SUCCESS

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RES_SYSTEM Field

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► ErrorCodes

► RES_SYSTEM

C#

[This is preliminary documentation and is subject to change.]

Sistem error not recoverable.

Declaration Syntax

C# Visual Basic Visual C++

public const int RES_SYSTEM

Public Const RES_SYSTEM As Integer

public:

literal int RES SYSTEM

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RES_WARNING

Field

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► ErrorCodes

► RES_WARNING

C# -

[This is preliminary documentation and is subject to change.]

Client warning

Declaration Syntax

C# Visual Basic Visual C++

public const int RES_WARNING

Public Const RES_WARNING As Integer

public:

literal int RES WARNING

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Frequency Interpolation

Enumeration

Namespaces ►

ActiveTechnologies.Instruments.AWG4000.Control ►

FrequencyInterpolation

C#

[This is preliminary documentation and is subject to change.]

Frequency interpolation for DAC.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum FrequencyInterpolation			
Public Enumeration FrequencyInterpolation			

public enum class FrequencyInterpolation

Members

Member	Description
Frequency1X	Interpolation at 1X.
Frequency2X	Interpolation at 2X.
Frequency4X	Interpolation at 4X.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Frequency Modulation Law Struct

Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FrequencyModulationLawStruct

C#

[This is preliminary documentation and is subject to change.]

Struct of frequency modulation law.

Declaration Syntax

C#	Visual Basic	Visual C++	
public struct Fr	equencyModulation	LawStruct	

Public Structure FrequencyModulationLawStruct

public value class FrequencyModulationLawStruct

GetType()

■ Me	mbers					
All Me	All Members Methods			Properties		
Public Protection Protection Protection Protection Protection Public Pu						□ Declared □ Inherited □
Icon	Member		Desc	ription		
≝∳	Equals(Object)		Indicates whether this instance and a specified object are equal. (Inherited from ValueType .)			
	<u>FrequencyValue</u>		Value of frequency.			
≡	GetHashCode(()		rns the hash coderited from Valu		

Gets the <u>Type</u> of the current instance.

		(Inherited from Object.)
	Repetitions	Number of repetitions to be mantained.
∃	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Frequency Value

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>FrequencyModulationLawStruct</u> ► **FrequencyValue**

C# -

[This is preliminary documentation and is subject to change.]

Value of frequency.

Declaration Syntax

```
C# Visual Basic Visual C++

public double FrequencyValue { get; set; }

Public Property FrequencyValue As Double

public:
property double FrequencyValue {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Repetitions

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>FrequencyModulationLawStruct</u> ► **Repetitions**

C# -

[This is preliminary documentation and is subject to change.]

Number of repetitions to be mantained.

Declaration Syntax

```
C# Visual Basic Visual C++

public ulong Repetitions { get; set; }

Public Property Repetitions As ULong

public:
property unsigned long long Repetitions {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation FrequencyRange

Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FrequencyRange

[This is preliminary documentation and is subject to change.]

Frequency range for flatness

Declaration Syntax

C#	Visual Basic	Visual C++	
public class	FrequencyRange		
Public Class FrequencyRange			

public ref class FrequencyRange

■ Members

All Me	embers	Constructors		Methods	
Pub Prot Prot	lic ected				□ Declared □ Inherited
Icon	Member		Des	cription	
≘♦	FrequencyRan	ige()	Con	structor	
≅ ∲	Equals(Object)	is e	ermines whether th qual to the current s nerited from Object.	Object.
≅ ∲	GetHashCode(par	ves as a hash functi ticular type. nerited from <u>Object</u> .	
≓	GetType()		Get	s the <u>Type</u> of the cu	rrent instance.

		(Inherited from Object.)
≡©	Initialize(Double, Double)	Initialize frequency range
≡	Initialize(FrequencyRange)	Initialize frequency range
≅	<u>IsInRange(Double)</u>	Check if frequency is in range
≅∳	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

FrequencyRange

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation FrequencyRange

Constructor

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

FrequencyRange ► FrequencyRange()

C#	- 1
C#	

[This is preliminary documentation and is subject to change.]

Constructor

Declaration Syntax

C#	Visual Basic	Visual C++	
public FrequencyRa	nge()		
Public Sub New			
<pre>public:</pre>			
<pre>FrequencyRange()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation Initialize Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FrequencyRange ► Initialize()

C#		~
	La company of the Com	

[This is preliminary documentation and is subject to change.]

Members

Icon	Member	Description
=	Initialize(Double, Double)	Initialize frequency range
=	Initialize(FrequencyRange)	Initialize frequency range

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation Initialize Method

(FreqRange)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>FrequencyRange</u> ► **Initialize(FrequencyRange)**

C# -

[This is preliminary documentation and is subject to change.]

Initialize frequency range

Declaration Syntax

■ Parameters

FreqRange (FrequencyRange)

Range

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>

LeCroy

ArbStudio Control Classes Library Documentation Initialize Method

(MinFreq, MaxFreq)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FrequencyRange ► Initialize(Double, Double)

C# -

[This is preliminary documentation and is subject to change.]

Initialize frequency range

Declaration Syntax

```
Visual Basic
                                       Visual C++
C#
public void Initialize(
        double MinFreq,
        double MaxFreq
)
Public Sub Initialize (
        MinFreg As Double,
        MaxFreq As Double
)
public:
void Initialize(
        double MinFreq,
        double MaxFreq
)
```

Parameters

MinFreq (Double)
Min frequency

MaxFreq (Double)

Max frequency

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation IsInRange Method

(Frequency)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FrequencyRange ► IsInRange(Double)

C#

[This is preliminary documentation and is subject to change.]

Check if frequency is in range

Declaration Syntax

■ Parameters

Frequency (Double)

[Missing <param name="Frequency"/> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.FrequencyF

■ Return Value

[Missing <returns> documentation for "M:ActiveTechnologies.Instruments.AWG4000.Control.FrequencyRang

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation FSKChannel Class

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

FSKChannel

C#	•

[This is preliminary documentation and is subject to change.]

Class to manage an phase frequency modulated DDS channel on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class FSKChannel : DDSChannel

Public Class FSKChannel _ Inherits DDSChannel

public ref class FSKChannel: public DDSChannel

Members

All Me	embers	Constructors	Methods	Properties		
Publ Prot						Declare Inherite
Icon	Member				Des	scriptio
∄	<u>FSKChann</u>	<u>el()</u>				ializes a (Channe
	Amplitude	CorrectionFactor				amplitu nerited f
≡	ATXSSSIav	veDisableStartCon	dition()		chai	able star nnel in s
					(1111	ieritea ii

≅	ATXSSSlaveDisableStopCondition()	Disable stor channel in s
		(Inherited f
∃ ₩	ATXSSSlaveEnableStartCondition(ATXSSEvent)	Enable start channel in s line for chec
		(Inherited f
≟	ATXSSSlaveEnableStopCondition(ATXSSEvent)	Enable stop channel in s line for chea
		(Inherited f
	ChannelFunctionality	Get channel Pattern)
		(Inherited f
≡	ClearBuffer(DDSBufferType)	Reset the FI
		(Inherited f
≅©	EnableDisable(Boolean)	Enable or di output on o
		(Inherited f
	EnableFlatnessCompensation	Enables/disl flatness con user defined automatical internal con the user defined applied as is
		(Inherited f
=	Equals(Object)	Determines is equal to t
		(Inherited f
	ErrorResult	Contain the

		(Inherited f
=	GetHashCode()	Serves as a type.
		(Inherited f
≅	<pre>GetType()</pre>	Gets the Ty
		(Inherited f
≡©	GetVersion(String)	Read from F loaded.
		(Inherited f
=•	LoadCarrier(WaveformStruct)	Load waveformemory.
		(Inherited f
≟	LoadFSKModulationLaw(FrequencyModulationLawStruct[], TransferMode, Boolean)	Load FSK m
	<u>MemorySamples</u>	Memory sar
		(Inherited f
≡	ReadMode(ModulationType)	Get the Mod
		(Overrides DDSChanne
≡	ReadMode(Functionality)	Get the Fun
		(Inherited f
≡	ReadModulatingFrequency(Double)	Get the run
∉	ReadStatus(ChannelStatus)	Read status
		(Inherited f
≡	ResetSoftware()	Make a rese
		(Inherited f
		(2111101110011

	<u>SampligRatePrescaler</u>	Set samplin Available or
		(Inherited f
=	SetAmplitudeProfile(AmplitudeProfileStruct)	Set amplitu signal as a f
		(Inherited f
≡	<u>SetExternalTrigger(TriggerSource, SensitivityEdge, TriggerAction)</u>	Set Trigger
A		(Inherited f
≟	SetInternalTrigger()	Set Trigger
		(Inherited f
≡	<u>SetOutputImpedance(OutputImpedance)</u>	Set output i
		(Inherited f
=	SetOutputVoltage(Single)	Set output v channel mu
		(Inherited f
≡	SetTriggerDelay(UInt32)	Set a start (
		(Inherited f
=	SetTriggerMode(TriggerMode)	Set trigger Single, Con
		(Inherited f
≡	SetTriggerOut(TriggerSource[], SensitivityEdge)	Set Trigger
		(Inherited f
≡	SetTriggerOutDelay(UInt32)	Set a trigge
		(Inherited f
=•	ToString()	Returns a <u>S</u> current <u>Obj</u>
		(Inherited f

■ Inheritance Hierarchy

Object

<u>ATAWGBase</u>

<u>Channel</u>

DDSChannel

FSKChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation FSKChannel

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FSKChannel ► **FSKChannel()**

C#	-

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the FSKChannel class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public FSKChannel(</pre>)		
Public Sub New			
<pre>public: FSKChannel()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>

ArbStudio Control Classes Library LeCroy Namespaces ►

Documentation Load FSK Modulation Law Method (samples, mode, clearMemory)

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

FSKChannel ▶

LoadFSKModulationLaw(FrequencyModulationLawStruct[], TransferMode, Boolean)

C#

[This is preliminary documentation and is subject to change.]

Load FSK modulation law in the FIFO.

Declaration Syntax

```
C#
                     Visual Basic
                                           Visual C++
public ATError LoadFSKModulationLaw(
         FrequencyModulationLawStruct[] samples,
         TransferMode mode,
         bool clearMemory
)
Public Function LoadFSKModulationLaw (
         samples As FrequencyModulationLawStruct(),
         mode As <a href="mailto:TransferMode">TransferMode</a>,
         clearMemory As Boolean
) As ATError
public:
ATError^ LoadFSKModulationLaw(
         array<<u>FrequencyModulationLawStruct</u>>^ samples,
         <u>TransferMode</u> mode,
         bool clearMemory
)
```

Parameters

samples (FrequencyModulationLawStruct [])

Array of FrequencyModulationLawStruct with sample value and number o time interval to be maintained.

mode (TransferMode)

Transfer mode, ReEntrant for dynamic load of modulation law.

clearMemory (Boolean)

True to clear memory before load sequence.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadMode Method

(modulation)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

FSKChannel ► **ReadMode(ModulationType)**

C#

[This is preliminary documentation and is subject to change.]

Get the Modulation type of channel DDS.

Declaration Syntax

■ Parameters

) override

modulation (ModulationType)

out ModulationType of channel.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Read Modulating Frequency Method

(frequency)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

FSKChannel ► **ReadModulatingFrequency(Double)**

C#

[This is preliminary documentation and is subject to change.]

Get the running modulating frequency

Declaration Syntax

■ Parameters

frequency (Double)

out modulating frequency

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Functionality

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Functionality

C#

[This is preliminary documentation and is subject to change.]

Channel functionality: Arbitrary, PSK, FSK, PatternGen. Other values are for internal use.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Functi	onality		
Public Enumeration	Functionality		
public enum class	Functionality		

Members

Member	Description
None	Internal use: No definition.
ARB	Arbitrary channel.
DDS	Internal use: DDS channel.
PSK	DDS Phase modulation channel.
FSK	DDS Frequency modulation channel.

PatternGen	Pattern generator channel.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Generation Sequence Struct

Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

GenerationSequenceStruct

C# -

[This is preliminary documentation and is subject to change.]

Struct for waveform generation sequence

Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

public struct GenerationSequenceStruct

Public Structure GenerationSequenceStruct

public value class GenerationSequenceStruct

Members

All Members	Methods	Properties	
Public Public		▽ Instance	Declared
		Static	Inherited

Icon	Member	Description
≡©	Equals(Object)	Indicates whether this instance and a specified object are equal.
		(Inherited from <u>ValueType</u> .)
≡	<u>GetHashCode()</u>	Returns the hash code for this instance.
		(Inherited from ValueType .)
≡©	GetType()	Gets the <u>Type</u> of the current instance.
		(Inherited from Object.)
	Repetitions	Number of times to be generated.

∄	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)
	WaveformIndex	0-base index of waveform in the waveform struct list.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Repetitions

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► GenerationSequenceStruct ► Repetitions

C# -

[This is preliminary documentation and is subject to change.]

Number of times to be generated.

Declaration Syntax

```
C# Visual Basic Visual C++

public ulong Repetitions { get; set; }

Public Property Repetitions As ULong

public:
property unsigned long long Repetitions {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation WaveformIndex

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>GenerationSequenceStruct</u> ► **WaveformIndex**

C# 🔻

[This is preliminary documentation and is subject to change.]

0-base index of waveform in the waveform struct list.

■ Declaration Syntax

```
C# Visual Basic Visual C++
public uint WaveformIndex { get; set; }

Public Property WaveformIndex As UInteger

public:
property unsigned int WaveformIndex {
        unsigned int get ();
        void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Instrument Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Instrument

C#

[This is preliminary documentation and is subject to change.]

Instrument Utilities Class

Declaration Syntax

C# Visual Basic Visual C++

public static class Instrument

Public NotInheritable Class Instrument

public ref class Instrument abstract sealed

Members

All Members	Methods	
Public Public		

Icon	Member	Description
= \S		Get instrument running frequency based on given frequency

■ Inheritance Hierarchy

Object

Instrument

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



C#

ArbStudio Control Classes Library

Documentation GetRunning Frequency Method (frequency Pair, running Frequency Pair)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>Instrument</u>

Visual C++

GetRunningFrequency(Decimal, Decimal)

C# -

[This is preliminary documentation and is subject to change.]

Get instrument running frequency based on given frequency

Visual Basic

Declaration Syntax

Parameters

```
frequencyPair ( Decimal )
  Frequency [Hz] of left pair

runningFrequencyPair ( Decimal )
  out Internal running frequency left pair
```

Return Value

Result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation InstrumentType

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

InstrumentType

C#		

[This is preliminary documentation and is subject to change.]

Type of USB connected device.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Instru	umentType		
Public Enumeration	n InstrumentType		
public enum class	InstrumentType		

Members

Member	Description
Unknown	Device is unknown.
AWG4000	Device is ArbStudio 1104 (4 channels).
AWG2000	Device is ArbStudio 1102 (2 channels).
AWGPower	Obsolete

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation MemorySize

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

MemorySize

C #	~ 1
C 11	

[This is preliminary documentation and is subject to change.]

Amount of installed memory

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Memory	/Size		
Public Enumeration	n MemorySize		
public enum class	MemorySize		

Members

Member	Description
Size2Mega	2 Megabytes
Size256k	256 Kbytes

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



Documentation Modulation Trigger Mode

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ModulationTriggerMode

C#

[This is preliminary documentation and is subject to change.]

Trigger mode for arbitrary waveform modulation.

Visual Basic

Declaration Syntax

$C\pi$	Visual Dasic	Visual CTT	
public enum Modula	tionTriggerMode		
Public Enumeration	ModulationTriggerMo	ode	
public enum class	ModulationTriggerMod	de	

Visual C++

Members

Member	Description
Single	Single.
Continuous	Continuous.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation Modulation Type

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

ModulationType

C#	~ 1
U 11	

[This is preliminary documentation and is subject to change.]

Modulation type of channel.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Modula	ationType		
Public Enumeration	n ModulationType		
public enum class	ModulationType		

■ Members

Member	Description
Frequency	Modulation frequency
Amplitude	Modulation amplitude.
Phase	Modulation phase.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Output Impedance

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

OutputImpedance

C#	~ 1
U 11	

[This is preliminary documentation and is subject to change.]

Output channel impedance.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Output	Impedance		
Public Enumeration	OutputImpedance		
public enum class	OutputImpedance		

Members

Member	Description
Ohm50	50 ohm
LowImpedance	Low impedance.
HighImpedance	High impedance.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



C#

ArbStudio Control Classes Library

Documentation Pair Channel Add Subs Operation

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannelAddSubsOperation

Visual Basic

C#

[This is preliminary documentation and is subject to change.]

Specify sign in add operation between channels.

■ Declaration Syntax

public enum PairChannelAddSubsOperation	
Public Enumeration PairChannelAddSubsOperation	
<pre>public enum class PairChannelAddSubsOperation</pre>	

Visual C++

Members

Member	Description
OperationAdd	Add operation has positive sign.
OperationSubs	Add operation has negative sign.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



Documentation Pair Channel Direction Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannelDirection

C#

[This is preliminary documentation and is subject to change.]

Link type between pair channels.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum PairCh	nannelDirection		
Public Enumeration	n PairChannelDirecti	on	
public enum class	PairChannelDirectio	n	

Members

Member	Description
LinkNone	No link defined
LinkLeft	Waveform samples operation are from channel 2 to channel 1 (and form channel 4 to channel 3 for AWG4000)
LinkRight	Waveform samples operation are from channel 1 to channel 2 (and form channel 3 to channel 4 for AWG4000)

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



C#

ArbStudio Control Classes Library

Documentation Pair Channel Math Operation

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannelMathOperation

Visual Basic

C#

[This is preliminary documentation and is subject to change.]

Math operation type for pair channel.

■ Declaration Syntax

•		7.00.0 20.0.0		
public	enum PairCha	annelMathOperation		
Public	Enumeration	PairChannelMathOper	ation	
public	enum class I	PairChannelMathOpera	ntion	

Visual C++

Members

Member	Description
WithChannel	Operation is performed between channels samples.
WithConstant	Operation is performed between channel and constant.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation PairChannels Class

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

PairChannels

C#	

[This is preliminary documentation and is subject to change.]

Class to manage an pair of channels (left: channels 1, 2 or right: channels 3, 4) on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class PairChannels : ATAWGBase

Public Class PairChannels _ Inherits ATAWGBase

public ref class PairChannels : public ATAWGBase

Members

	Channell oft			Get or set left
Icon	Member			Description
Pub Prot Prot	lic cected			□ Declared □ Inherited
All M	embers	Methods	Properties	

ChannelLeft Get or set left channel of pair.

ChannelRight Get or set right channel of pair.

Disable Digital Port()

Disable digital out

<u>DisableTriggerOutOnDigitalPort()</u>
Disable trigger out

		on digital port.
- 3	EnableDigitalPort()	Enable digital out.
≝₩	EnableTriggerOutOnDigitalPort()	Enable trigger out on digital port.
€Ŵ	Equals(Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> . (Inherited from <u>Object</u> .)
	ErrorResult	Contain the result status of error. (Inherited from ATAWGBase.)
⊒₩	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object.)
≟	GetType()	Gets the Type of the current instance. (Inherited from Object.)
	<u>Pattern</u>	Get or set pattern generator channel of pair.

æ	ReadPLLStatus(PLLStatus)	Read PLL status of assigned channels If one is not locke result is not locked.
∃	SetDigitalPortVoltage(Single)	Set Voh of digital out.
∄	SetFrequencyInterpolation(FrequencyInterpolation)	Set DAC frequency interpolation factor.
₫ ڼ	SetLinkDirection(PairChannelDirection, PairChannelMathOperation, Double, PairChannelMathOperation, Double, PairChannelAddSubsOperation)	Set pair channels FPGA communication direction.
₫◊	SetReferenceVoltage(Single)	Set digital connector reference voltage.
≓ڼ	ToString()	Returns a <u>String</u> that represents th current <u>Object</u> . (Inherited from <u>Object</u> .)

■ Inheritance Hierarchy

Object

<u>ATAWGBase</u>

PairChannels

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Channel Left

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ► ChannelLeft

C#

[This is preliminary documentation and is subject to change.]

Get or set left channel of pair.

Declaration Syntax

```
C# Visual Basic Visual C++

public Channel ChannelLeft { get; set; }

Public Property ChannelLeft As Channel

public:
property Channel^ ChannelLeft {
        Channel^ get ();
        void set (Channel^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Channel Right

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ► ChannelRight

C# -

[This is preliminary documentation and is subject to change.]

Get or set right channel of pair.

Declaration Syntax

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Disable Digital Port

Method

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannels ► DisableDigitalPort()

C# _

[This is preliminary documentation and is subject to change.]

Disable digital out.

Declaration Syntax

C# Visual Basic Visual C++

public ATError DisableDigitalPort()

Public Function DisableDigitalPort As ATError

public:

ATError^ DisableDigitalPort()

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation DisableTriggerOutOnDigitalPort

Method

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>PairChannels</u> ► **DisableTriggerOutOnDigitalPort()**

C#

[This is preliminary documentation and is subject to change.]

Disable trigger out on digital port.

■ Declaration Syntax

C# Visual Basic Visual C++

public ATError DisableTriggerOutOnDigitalPort()

Public Function DisableTriggerOutOnDigitalPort As ATError

public:

ATError^ DisableTriggerOutOnDigitalPort()

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Enable Digital Port

Method

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannels ► EnableDigitalPort()

C# 🔽

[This is preliminary documentation and is subject to change.]

Enable digital out.

Declaration Syntax

C# Visual Basic Visual C++

public ATError EnableDigitalPort()

Public Function EnableDigitalPort As <u>ATError</u>

public:

ATError^ EnableDigitalPort()

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Enable Trigger Out On Digital Port

Method

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ► EnableTriggerOutOnDigitalPort()

C#

[This is preliminary documentation and is subject to change.]

Enable trigger out on digital port.

■ Declaration Syntax

C# Visual Basic Visual C++

public ATError EnableTriggerOutOnDigitalPort()

Public Function EnableTriggerOutOnDigitalPort As ATError

public:

ATError^ EnableTriggerOutOnDigitalPort()

■ Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation Pattern Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ▶ Pattern

C# _

[This is preliminary documentation and is subject to change.]

Get or set pattern generator channel of pair.

Declaration Syntax

```
C# Visual Basic Visual C++
public PatternGenerator Pattern { get; set; }

Public Property Pattern As PatternGenerator

public:
property PatternGenerator^ Pattern {
          PatternGenerator^ get ();
          void set (PatternGenerator^ value);
}
```

Remarks

A pattern generator channel is composed of two channels of pair.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadPLLStatus

Method (status)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ► ReadPLLStatus(PLLStatus)

C# 💌

[This is preliminary documentation and is subject to change.]

Read PLL status of assigned channels. If one is not locked result is not locked.

Declaration Syntax

■ Parameters

```
status ( PLLStatus )
out Status of PLL.
```

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

<u>LeCroy</u>

ArbStudio Control Classes Library

DocumentationSetDigitalPortVoltage Method

(voltage)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PairChannels</u> ► SetDigitalPortVoltage(Single)

C# _

[This is preliminary documentation and is subject to change.]

Set Voh of digital out.

■ Declaration Syntax

Parameters

voltage (Single)

Voltage to be set: 1.6...3.6

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation SetFrequencyInterpolation Method (interpolation)

ActiveTechnologies.Instruments.AWG4000.Control ►

PairChannels ►

SetFrequencyInterpolation(FrequencyInterpolation)

C#

[This is preliminary documentation and is subject to change.]

Set DAC frequency interpolation factor.

Declaration Syntax

```
C#
                    Visual Basic
                                         Visual C++
public ATError SetFrequencyInterpolation(
        <u>FrequencyInterpolation</u> interpolation
Public Function SetFrequencyInterpolation (
        interpolation As FrequencyInterpolation
) As ATError
public:
ATError^ SetFrequencyInterpolation(
        <u>FrequencyInterpolation</u> interpolation
)
```

Parameters

interpolation (FrequencyInterpolation)

FrequencyInterpolation value

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved

ArbStudio Control Classes Library Documentation SetLinkDirection

Method (linkDirection, multiplyOperationMode, multiplyConstant, plusOperationMode, plusConstant, addsubsMode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PairChannels</u> ► SetLinkDirection(PairChannelDirection,

PairChannelMathOperation, Double, PairChannelMathOperation, Double,

PairChannelAddSubsOperation)

C#

[This is preliminary documentation and is subject to change.]

Set pair channels FPGA communication direction.

Declaration Syntax

LeCroy

```
public:
ATError^ SetLinkDirection(
```

```
PairChannelDirection linkDirection,
PairChannelMathOperation multiplyOperationMode,
double multiplyConstant,
PairChannelMathOperation plusOperationMode,
double plusConstant,
PairChannelAddSubsOperation addsubsMode
)
```

Parameters

linkDirection (PairChannelDirection)

Direction of communication

multiplyOperationMode (PairChannelMathOperation)

Specify if multiply operation is with channel or constant.

multiplyConstant (Double)

Multiply constant for multiply operation.

plusOperationMode (PairChannelMathOperation)

Specify if plus operation is with channel or constant.

plusConstant (Double)

Plus constant for multiply operation.

addsubsMode (PairChannelAddSubsOperation)

Specify if plus operation has positive or negative sign.

Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved

<u>LeCroy</u>

ArbStudio Control Classes Library

Documentation Set Reference Voltage Method

(voltage)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PairChannels ► SetReferenceVoltage(Single)

C#

[This is preliminary documentation and is subject to change.]

Set digital connector reference voltage.

Declaration Syntax

Parameters

voltage (Single)

Voltage to be set: -5...5

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Pattern_Control Register Type

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Pattern_ControlRegisterType

 C#	•
\subset^{π}	J. I

[This is preliminary documentation and is subject to change.]

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Patter	rn_ControlRegisterTy	ре	
Public Enumeration	n Pattern_ControlReg	isterType	
public enum class	Pattern_ControlRegi	sterType	

Members

Member	Description
Latch	
NoLatch	
Clear	
NoClear	
AsyncReset	
NoAsyncReset	
ResetCount	
NoResetCount	

AlignerStopped	
StartAligner	
AcqMasterTrg	
AcqCh17	

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Pattern_SC_Connect_Pin

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Pattern_SC_Connect_Pin

C#

[This is preliminary documentation and is subject to change.]

Expansion Bus Pin Connect

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Patter	n_SC_Connect_Pin		
Public Enumeration	Pattern_SC_Connect	_Pin	
public enum class	Pattern_SC_Connect_	Pin	

Members

Description
Expansion Bus Line 0
Expansion Bus Line 1
Expansion Bus Line 2
Expansion Bus Line 3
Internal

Disabled	Disabled

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Pattern_Selection Register Type

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Pattern_SelectionRegisterType

C#

[This is preliminary documentation and is subject to change.]

Parameter for Selection Register

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Patte	ern_SelectionRegist	terType	
Public Enumeration	on Pattern_Selection	onRegisterType	
nublic enum class	Pattern Selection	nRegisterTyne	

Members

Member	Description
Strobe	The pods of the Pattern Generator are in strobe mode
HiZ	The pods of the Pattern Generator are in HiZ
ClkGenOut	ClkGenOut sets the Pod 15 as Clock output
VectOut	ClkGenOut sets the Pod 15 as Vector output
SDR	Single Data Rate Update Mode

DDR_OUT	Double Data Rate Update Mode Not Available
DDR_IN	Double Data Rate Sample Mode Not Available
PatternAligner	Pattern Aligner trasmission Not Available -
PatternVect	Vectors trasmission
StartSeq	Starts the Sequencer module
StopSeq	Stops the Sequencer module
SerBitMode8	Serializer Mode 8 bit (CH0,CH1,CH2,CH3) Not Available
SerBitMode16	Serializer Mode 16 bit (CH0,CH1) Not Available
SerBitMode32	Serializer Mode 32 bit (CH0) Not Available
Cycle	Cyclic memory
OneShot	One shot memory

Parallel	Parallel Vector Mode
Serial	Serial Vector Mode Not Available
ResetStartSeq	Reset the Start Sequencer signal
ResetStopSeq	Reset the Stop Sequencer signal

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PatternGenerator

Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator

C#	
----	--

Bus

[This is preliminary documentation and is subject to change.]

Class to manage a Digital Pattern Generator-Sampler device

Declaration Syntax

C# Visual Basic Visual C++

public class PatternGenerator : ATAWGBase

Public Class PatternGenerator _ Inherits ATAWGBase

public ref class PatternGenerator : public ATAWGBase

Members

	ilibel 5			
All Members Methods		Methods	Properties	
PublicProtected✓ Instance✓ Static		□ Declared □ Inherited		
Icon	Member			Description
∃			Disable Start condition on SI devices connec by the Expansion Bus	
≟	ATXSSSlaveDi	<u>isableStopConditionF</u>	<u>'attern()</u>	Disable Stop condition on SI devices connec by the Expansion

≅	ATXSSSlaveEnableStartConditionPattern(ATXSSEvent)	Enable Start condition on Sladevices connect by the Expansion Bus.
∃	ATXSSSlaveEnableStopConditionPattern(ATXSSEvent)	Enable Stop condition on SI devices connec by the Expansion Bus.
≓ ₩	ATXSSTriggerPattern(Boolean)	Set the Trigger Event as comin from Expansior Bus LINE 7.
≓	ClockGenEnable(Boolean)	It enables or disables the Clo Generator outp
	ClockGenSetFrequency(Single, Single)	It sets the outp frequency of th Clock Generato module. Precautions: th clock generator output frequen depends on the device Sample set on the instrument.
≡	ClockGenSetStrobeOperation(ClockStrobeOpType)	It sets the strol

		operation of the Clock Generato Module.
	ClockGenSetSyncDelay(Byte)	It sets the sync delay. The Cloc Generator mod generates a synsignal for the Sequencer modes of vectors can fully syncronize with the general clock. This function apply a delato the sync sign
	CreateVectorControlsSegmentation(UInt32[], UInt16)	It creates the segmentation f the vectors and controls. The u can segmentate the vector/cont memory in bloc that the sequer will recall durin operations.
∃	Equals(Object)	Determines whether the specified Objec equal to the current Object. (Inherited from Object.)
	ErrorResult	Contain the res

		status of error.
		(Inherited from ATAWGBase.)
ΞΦ	GetFirmwareVersion(Byte[])	Gets the firmway
≅	GetHashCode()	Serves as a had function for a particular type.
		(Inherited from Object.)
∃•	GetMemorySamplesAddress(Int32, Boolean)	Gets the address the last data st into the SRAM it signals in Cycle mode, and it signals in Cycle mode if the cyc memory has reached or not last address.
∄	GetPatternCode(Byte[])	Gets the FPGA Pattern Code
≅₩	GetPIILocked(Boolean)	Checks if the Pl the external clo has locked
Ξ₩	GetPodStatus(Byte[])	Gets the chann status of the Pattern Genera The caller has t instantiate an

		array of 40 byt (Status[]) when Status[0],Statu are the CH0 va sampled in DDF (Double Data Rate); Status[2],Statu are the CH1 va sampled in DDF and so on. Befor calling GetPodStatus, user has to cha the buffers direction in HiZ
	GetServiceRegs(Byte[])	Gets the servic register value. ((TrgStart[0] A 0X2)== '0')) the instrument is in idle state, if it's equal '1', it is running.
∃	GetType()	Gets the Type of the current instance. (Inherited from
∃	LoadBlock(SEQInstructionCodeType[], UInt32, UInt16)	Object.) Loads the sequencer instructions int the Pattern Generator sequencer men

The maximum

		number of instructions is 1024.
	LoadVectorControls(UInt32[], Byte[], UInt32, UInt16)	It loads the vec and the control into the Patterr Generator vector/control memory. The memory depth 1048576x36 bi
□	ReadMemoryData(Int32, String)	Gets the data stored inside the Pattern General memory and withem into a file. The data are acquired in DDI (250 MHz - one sample every 4 and the file for is: 1 sample every 4 and the file for is: 1 sample every 5 bytes (last 2 bytes of the salare always 0) Bytes of the salare always 0) Bytes (70) = CH(70) Samp 0; Bytes (70) = CH(158) Sam 0; Bytes (10) = CH(1716) Sar 0; Bytes (10
	<u>SamplingRatePrescaler</u>	Set sampling ra

		prescaler for pattern channe
	SelStartHiZ(Pattern_SelectionRegisterType)	It sets the selection register. The selection regist configures the instrument parameters for Strobe/Hiz bidirectional pin Pin(15) function Clock Generate Output or Vector Output and Memory Update Mode (One sho Cycle)
	SetAcqTrg(Boolean)	On a slave devicted Ch17 can be us as an input to evaluate the trigger conditio (Master Trigger Event disabled) as a Master Triger Event. The deviset as the mast transmits the Trigger Event be the Expansion to all slave devises as the mast transmits the the Expansion to all slave devises as the mast transmits the the Expansion to all slave devises as the mast transmits the the Expansion to all slave devises as the mast transmits the the Expansion to all slave devises as the mast transmits the the Expansion to all slave devises as the mast transmits the trigger Event be the Expansion to all slave devises as the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion to all slave devises the mast transmits the trigger Event be the Expansion tri
≡	SetMasterPattern(Boolean)	Set the Master channel.

∄	<u>SetSuperCableRegs(Boolean,</u> <u>Pattern_SC_Connect_Pin, Pattern_SC_Connect_Pin,</u> <u>Byte, Boolean)</u>	Set ExpansionE parameters
	SetTriggerCondition(TriggerConditionType[], Setup_TriggerCondType, Setup_LevelCondType, ActionType, Boolean, Byte, UInt32)	Sets the trigger condition on Ch[170]; the can set a trigger condition also on the Vector(16), the maximum number of available changer for triggering is 31 Trigger Lever are available.
€	SetTrigINConfig(TriggerSource, TriggerAction, SensitivityEdge)	Sets the Trigge configuration for the Digital Patt Generator.
€	SetTrigOUTConfig(TriggerSource[], SensitivityEdge)	Sets the Trigge OUT configurat for the Digital Pattern Genera
€	StrobeVectors(UInt64, Byte)	Strobe of the vectors.Before performing the strobe of the vectors, the usemust change the direction of the bidirectional pin

ToString()
Returns a Strin
that represents
current Object.

(Inherited from
Object.)

■ Inheritance Hierarchy

Object

ATAWGBase

PatternGenerator

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation ATXSSS lave Disable Start Condition P

Method

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

<u>PatternGenerator</u> ► ATXSSSlaveDisableStartConditionPattern

C#

[This is preliminary documentation and is subject to change.]

Disable Start condition on Slave devices connected by the Expansion Bus

Declaration Syntax

C# Visual Basic Visual C++

public ATError ATXSSSlaveDisableStartConditionPattern()

Public Function ATXSSSlaveDisableStartConditionPattern As <u>ATError</u>

public:

ATError^ ATXSSSlaveDisableStartConditionPattern()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation ATXSSS lave Disable Stop Condition Pa

Method

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

<u>PatternGenerator</u> ► ATXSSSlaveDisableStopConditionPattern(

C#

[This is preliminary documentation and is subject to change.]

Disable Stop condition on Slave devices connected by the Expansion Bus

Declaration Syntax

C# Visual Basic Visual C++

public ATError ATXSSSlaveDisableStopConditionPattern()

Public Function ATXSSSlaveDisableStopConditionPattern As <u>ATError</u>

public:

ATError^ ATXSSSlaveDisableStopConditionPattern()

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



DocumentationATXSSSlaveEnableStartConditionPa

Method (xssEvent)

Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control PatternGenerator ►

ATXSSSlaveEnableStartConditionPattern(ATXSSEvent)

C#

[This is preliminary documentation and is subject to change.]

Enable Start condition on Slave devices connected by the Expansion Bus.

Declaration Syntax

Parameters

xssEvent (ATXSSEvent)

None means internal Start, Stop means that Slave starts on a Stop event Start means that Slave starts on a Start event, PatEvent means that Slav starts on trigger event.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation ATXSSS lave Enable Stop Condition Particular State of Condition

Method (xssEvent)

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u> PatternGenerator ►

ATXSSSlaveEnableStopConditionPattern(ATXSSEvent)

C#

[This is preliminary documentation and is subject to change.]

Enable Stop condition on Slave devices connected by the Expansion Bus.

Declaration Syntax

Parameters

xssEvent (ATXSSEvent)

None means internal Stop, Stop means that Slave stops on a Stop event, Start means that Slave stops on a Start event, PatEvent means that Slave stops on trigger event.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation ATXSSTrigger Pattern Method

(fromATXSS)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ► ATXSSTriggerPattern(Boolean)

C#

[This is preliminary documentation and is subject to change.]

Set the Trigger Event as coming from Expansion Bus LINE 7.

Declaration Syntax

Parameters

fromATXSS (Boolean)

If false the Trigger Event is internal, if true it is from the Line7 of Expansion Bus

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved

copyright © 2010, Lecroy Corporation, An rights reserved



ArbStudio Control Classes Library Documentation ClockGenEnable

Method (On)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

<u>PatternGenerator</u> ► ClockGenEnable(Boolean)

C# -

[This is preliminary documentation and is subject to change.]

It enables or disables the Clock Generator output.

Declaration Syntax

■ Parameters

On (Boolean)

If On is true, it enables the output. If false, it disables the Clock Generato output.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Clock Gen Set Frequency Method (Frequency, Duty Cycle)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► ClockGenSetFrequency(Single, Single)

C#

[This is preliminary documentation and is subject to change.]

It sets the output frequency of the Clock Generator module. Precautions: the clock generator output frequency depends on the device Sample Rate set on the instrument.

Declaration Syntax

```
C#
                     Visual Basic
                                          Visual C++
public ATError ClockGenSetFrequency(
        float Frequency,
         float DutyCycle
)
Public Function ClockGenSetFrequency (
        Frequency As Single, _
        DutyCycle As <a href="Single">Single</a>
) As ATError
public:
ATError^ ClockGenSetFrequency(
         float Frequency,
         float DutyCycle
)
```

Parameters

Frequency (Single)

sets the frequency (in MHz) of the Clock Generator. Warning: entries coul be resampled to allowed values.

DutyCycle (Single)

sets the clock generator duty cycle. Warning: entries could be resampled to allowed values.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Clock GenSetStrobeOperation

Method (StrobeOperation)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ►

ClockGenSetStrobeOperation(ClockStrobeOpType)

C#

[This is preliminary documentation and is subject to change.]

It sets the strobe operation of the Clock Generator Module.

Declaration Syntax

Parameters

StrobeOperation (ClockStrobeOpType)

Strobe Operation parameter

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved

<u>LeCroy</u>

ArbStudio Control Classes Library

Documentation Clock Gen Set Sync Delay Method

(Delay)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ► ClockGenSetSyncDelay(Byte)

C#

[This is preliminary documentation and is subject to change.]

It sets the sync delay. The Clock Generator module generates a sync signal for the Sequencer module, so vectors can be fully syncronized with the generate clock. This function can apply a delay to the sync signal.

Declaration Syntax

Parameters

Delay (Byte)

Sync delay. Allowed values are 0..255

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



Documentation Create Vector Controls Segmentation

Method (pSegmentDim, Count)

<u>Namespaces</u> ► <u>ActiveTechnologies.Instruments.AWG4000.Control</u>

► <u>PatternGenerator</u> ►

CreateVectorControlsSegmentation(UInt32[], UInt16)

C#	<u>F</u>

[This is preliminary documentation and is subject to change.]

It creates the segmentation for the vectors and the controls. The user can segmentate the vector/control memory in blocks that the sequencer will recal during its operations.

Declaration Syntax

```
C#
                    Visual Basic
                                        Visual C++
public ATError CreateVectorControlsSegmentation(
        uint[] pSegmentDim,
        ushort Count
)
Public Function CreateVectorControlsSegmentation ( _
        pSegmentDim As UInteger(), _
        Count As UShort
) As <u>ATError</u>
public:
ATError^ CreateVectorControlsSegmentation(
        array<unsigned int>^ pSegmentDim,
        unsigned short Count
)
```

Parameters

pSegmentDim (UInt32 [])

array that contains the segments dimensions. Total segments dimension is 1048576

Count (UInt16)

pSegmentDim array size

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

<u>LeCroy</u>

ArbStudio Control Classes Library

Documentation GetFirmware Version Method

(FirmVers)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► GetFirmwareVersion(Byte[])

C#

[This is preliminary documentation and is subject to change.]

Gets the firmware version

Declaration Syntax

Parameters

FirmVers (Byte [])

Firmware Version

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



C#

ArbStudio Control Classes Library

Documentation Get Memory Samples Address

Method (MemAddress, Turnaround)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ► **GetMemorySamplesAddress(Int32,**

Boolean)

C# _

[This is preliminary documentation and is subject to change.]

Visual Basic

Gets the address of the last data stored into the SRAM and it signals in Cycle mode, and it signals in Cycle mode if the cyclic memory has reached or not it last address.

Visual C++

Declaration Syntax

Parameters

MemAddress (Int32)

is the memory address of the last data stored

Turnaround (**Boolean**)

if Turnaround = true the cyclic memory has reached its last address.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation GetPatternCode

Method (PattCode)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► GetPatternCode(Byte[])

C# _

[This is preliminary documentation and is subject to change.]

Gets the FPGA Pattern Code

Declaration Syntax

```
C# Visual Basic Visual C++

public ATError GetPatternCode(
    ref byte[] PattCode
)

Public Function GetPatternCode (
        ByRef PattCode As Byte()
) As ATError

public:
ATError^ GetPatternCode(
    array<unsigned char>^% PattCode
)
```

■ Parameters

```
PattCode ( Byte [] )
Pattern Code is 0x10
```

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation **GetPIlLocked**

Method (Locked)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

<u>PatternGenerator</u> ► **GetPIILocked(Boolean)**

C# 🔽

[This is preliminary documentation and is subject to change.]

Checks if the PLL of the external clock has locked

Declaration Syntax

■ Parameters

```
Locked ( Boolean )
If true, the PLL has locked.
```

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation **GetPodStatus**

Method (Status)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► GetPodStatus(Byte[])

C#

[This is preliminary documentation and is subject to change.]

Gets the channels status of the Pattern Generator. The caller has to instantiat an array of 40 bytes (Status[]) where Status[0], Status[1] are the CHO values sampled in DDR (Double Data Rate); Status[2], Status[3] are the CHO values sampled in DDR and so on. Before calling GetPodStatus, the user has to change the buffers direction in HiZ.

Declaration Syntax

Parameters

```
Status ( Byte []) 40 bytes array
```

Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation GetServiceRegs

Method (TrgStart)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► GetServiceRegs(Byte[])

C# -

[This is preliminary documentation and is subject to change.]

Gets the service register value. If ((TrgStart[0] AND 0X2)== '0')) the instrument is in idle state, if it's equal '1', it is running.

Declaration Syntax

Parameters

TrgStart (Byte [])

TrgStart[0] gives the Service Register Value.

■ Return Value

ATError class result of operation

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation Load Block Method

(pInstructions, Count, Block)

ActiveTechnologies.Instruments.AWG4000.Control ►

PatternGenerator ► LoadBlock(SEQInstructionCodeType[],

UInt32, UInt16)

C#

[This is preliminary documentation and is subject to change.]

Loads the sequencer instructions into the Pattern Generator sequencer memory. The maximum number of instructions is 1024.

Declaration Syntax

```
Visual C++
C#
                    Visual Basic
public ATError LoadBlock(
        SEQInstructionCodeType[] pInstructions,
        uint Count,
        ushort Block
)
Public Function LoadBlock ( _
        pInstructions As SEQInstructionCodeType(),
        Count As <u>UInteger</u>, _
        Block As <u>UShort</u>
) As ATError
public:
ATError^ LoadBlock(
        array<<u>SEQInstructionCodeType</u>^>^ pInstructions,
        unsigned int Count,
        unsigned short Block
)
```

Parameters

```
pInstructions ( <a href="SEQInstructionCodeType">SEQInstructionCodeType</a> [])
   sequencer instructions array
Count (UInt32)
   array size of pInstructions
```

Block (<u>UInt16</u>) block address

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

ArbStudio Control Classes Library Documentation Load Vector Controls Method (pVectors, pControls, Count, Vector Control) Namespaces ► ActiveTechnologies.Instruments.AWG4000.Control PatternGenerator ► Load Vector Controls (UInt32[], Byte[], UInt32, UInt16)

[This is preliminary documentation and is subject to change.]

It loads the vectors and the controls into the Pattern Generator vector/contro memory. The memory depth is 1048576x36 bits.

C#

Declaration Syntax

```
Visual C++
C#
                    Visual Basic
public ATError LoadVectorControls(
        uint[] pVectors,
        byte[] pControls,
        uint Count,
        ushort VectorControl
)
Public Function LoadVectorControls (
        pVectors As UInteger(), _
        pControls As Byte(), _
        Count As <u>UInteger</u>,
        VectorControl As UShort
) As ATError
public:
ATError^ LoadVectorControls(
        array<<u>unsigned int</u>>^ pVectors,
        array<unsigned char>^ pControls,
        unsigned int Count,
        unsigned short VectorControl
```

Parameters

pVectors (UInt32 [])

Vectors array. pVectors[i](17 downto 0) represent the value of the output pins, pVectors[i](31 downto 18) represent the direction of the pins 13..0. If the direction is '1', the pin works as Output, if the direction is '0', it works as Input.

pControls (Byte [])

pControls[i](3 downto 0) represent the direction of the pins 17..14. If the direction is '1', the pin works as Output, if the direction is '0', it works as Input.

Count (UInt32)

vector/control array size

VectorControl (UInt16)

vector/control block number

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation Read Memory Data

Method (length, FilePath)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► ReadMemoryData(Int32, String)

C#

[This is preliminary documentation and is subject to change.]

Gets the data stored inside the Pattern Generator memory and write them into a file. The data are acquired in DDR (250 MHz - one sample every 4 ns) and the file format is: 1 sample every 5 bytes (last 2 bytes of the sample are always 0) Byte0(7..0) = CH(7..0) Sample 0;Byte1(7..0) = CH(15..8) Sample 0;Byte2(1..0) = CH(17..16) Sample 0;Byte 3 = 0x0 Sample 0; Byte 4 = 0x0 Sample 0

Declaration Syntax

```
Visual C++
C#
                     Visual Basic
public ATError ReadMemoryData(
        int length,
        string FilePath
)
Public Function ReadMemoryData (
        length As <u>Integer</u>, _
        FilePath As String
) As ATError
public:
ATError^ ReadMemoryData(
        int length,
        String<sup>^</sup> FilePath
)
```

Parameters

```
length (Int32)
length = 5*memoryLength (see example on SDK application)
FilePath (String)
File Path
```

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Sampling Rate Prescaler Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ► **SamplingRatePrescaler**

C# __

[This is preliminary documentation and is subject to change.]

Set sampling rate prescaler for pattern channel

Declaration Syntax

```
C# Visual Basic Visual C++
public uint SamplingRatePrescaler { get; set; }

Public Property SamplingRatePrescaler As UInteger

public:
property unsigned int SamplingRatePrescaler {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SelStartHiZ

Method (SelectionRegister)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ►

SelStartHiZ(Pattern_SelectionRegisterType)

C#

[This is preliminary documentation and is subject to change.]

It sets the selection register. The selection register configures the instrument parameters for Strobe/Hiz bidirectional pins, Pin(15) function as Clock Generator Output or Vector Output and Memory Update Mode (One shot or Cycle)

Declaration Syntax

Parameters

SelectionRegister (<u>Pattern_SelectionRegisterType</u>)

Selection Register parameter

Return Value

ATError class result of operation.

Send comments on this topic to LeCroy Corporation Support

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetAcqTrg Method

(MasterTrg)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► SetAcqTrg(Boolean)

C# -

[This is preliminary documentation and is subject to change.]

On a slave device, Ch17 can be used as an input to evaluate the trigger condition (Master Trigger Event disabled) or as a Master Trigger Event. The device set as the master transmits the Trigger Event by the Expansion Bus to all slave devices.

Declaration Syntax

Parameters

MasterTrg (Boolean)

If True enables the master trigger event

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetMasterPattern

Method (master)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

<u>PatternGenerator</u> ► **SetMasterPattern(Boolean)**

C#

[This is preliminary documentation and is subject to change.]

Set the Master channel.

Declaration Syntax

■ Parameters

master (Boolean)

If true the channel is the Master, otherwise it is Slave

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Set Super Cable Regs Method (MasterSlave, StartToPin, StopToPin, AsyncCfg, TrigFromSC7)

<u>Namespaces</u> ►

ArbStudio Control Classes Library

ActiveTechnologies.Instruments.AWG4000.Control ►

PatternGenerator ► SetSuperCableRegs(Boolean,

Pattern SC Connect Pin, Pattern SC Connect Pin, Byte,

Boolean)

C#

[This is preliminary documentation and is subject to change.]

Set ExpansionBus parameters

Declaration Syntax

```
Visual C++
C#
                   Visual Basic
public ATError SetSuperCableRegs(
        bool MasterSlave,
        Pattern SC Connect Pin StartToPin,
        Pattern SC Connect Pin StopToPin,
        byte AsyncCfq,
        bool TrigFromSC7
)
Public Function SetSuperCableRegs (
       MasterSlave As Boolean,
        StartToPin As Pattern SC Connect Pin,
        StopToPin As Pattern SC Connect Pin,
       AsyncCfq As Byte,
        TrigFromSC7 As Boolean
) As ATError
```

```
public:
ATError^ SetSuperCableRegs(
        bool MasterSlave,
        Pattern SC Connect Pin StartToPin,
        Pattern SC Connect Pin StopToPin,
        unsigned char AsyncCfq,
        bool TrigFromSC7
```

Parameters

MasterSlave (Boolean)

True for master

StartToPin (Pattern_SC_Connect_Pin)

Start pin from the Expansion Bus bus. If Master Start is trasmitted on ExpansionBus[0].

StopToPin (Pattern_SC_Connect_Pin)

Stop pin from ExpansionBus bus. If Master Start Stop is trasmitted on ExpansionBus[2]

AsyncCfg (Byte)

if '0' the trigger signal comes from the internal logic, if '1' from the ExpansionBus[4], if '2' or '3' the trigger comes from one of the ExpansionBus lines 0,1,2,3,4

TrigFromSC7 (Boolean)

Get trigger from ExpansionBus channel 7 or internal

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

ArbStudio Control Classes Library

(TriggerCondition Method (TriggerConditionArray, TriggerCond, LevelCond, Action, CondA_B, LevelNum, Pretrigger)

LeCroy

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PatternGenerator</u> ►

SetTriggerCondition(TriggerConditionType[], Setup_TriggerCondType, Setup_LevelCondType, ActionType, Boolean, Byte, UInt32)

C#

[This is preliminary documentation and is subject to change.]

Sets the trigger condition on Ch[17..0]; the user can set a trigger condition also on the Vector(16), so the maximum number of available channels for triggering is 18. 31 Trigger Levels are available.

Declaration Syntax

Parameters

TriggerConditionArray (TriggerConditionType [])

array of 18 elements that contains the trigger condition on each channel.

TriggerCond (Setup_TriggerCondType)

sets the conditions on edges

LevelCond (Setup_LevelCondType)

sets the level condition

Action (ActionType)

Action for trigger level

CondA_B (Boolean)

Condition A if true, Condition B if false

LevelNum (Byte)

Trigger Level Number

Pretrigger (UInt32)

N.C.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



ArbStudio Control Classes Library Documentation SetTrigINConfig

Method (TrgSource, Action, Edge)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

<u>PatternGenerator</u> ► **SetTrigINConfig(TriggerSource,**

TriggerAction, SensitivityEdge)

C#

[This is preliminary documentation and is subject to change.]

Sets the Trigger IN configuration for the Digital Pattern Generator.

Declaration Syntax

```
Visual C++
C#
                   Visual Basic
public ATError SetTrigINConfig(
        TriggerSource, TrqSource,
        TriggerAction Action,
        SensitivityEdge Edge
)
Public Function SetTrigINConfig ( _
        TrgSource As TriggerSource,
        Action As TriggerAction,
        Edge As SensitivityEdge
) As ATError
public:
ATError^ SetTrigINConfig(
        TriggerSource, TrqSource,
        TriggerAction Action,
        SensitivityEdge Edge
)
```

Parameters

TrgSource (TriggerSource)

FrontPanel means the Trigger IN comes from the front panel BNC connector, DigitalConnector from the front panel Digital Connector

Action (TriggerAction)

It is the action associated to the trigger signal.

Edge (SensitivityEdge)

It is the sensitivity edge of the trigger signal.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SetTrigOUTConfig

Method (TrgOutSource, Polarity)

ActiveTechnologies.Instruments.AWG4000.Control ►

PatternGenerator ► SetTrigOUTConfig(TriggerSource[],

SensitivityEdge)

C#

[This is preliminary documentation and is subject to change.]

Sets the Trigger OUT configuration for the Digital Pattern Generator.

Declaration Syntax

```
Visual C++
C#
                    Visual Basic
public ATError SetTrigOUTConfig(
        TriggerSource[] TrgOutSource,
        SensitivityEdge Polarity
)
Public Function SetTrigOUTConfig (
        TrgOutSource As TriggerSource(),
        Polarity As <u>SensitivityEdge</u>
) As ATError
public:
ATError^ SetTrigOUTConfig(
        array<<u>TriggerSource</u>>^ TrgOutSource,
        SensitivityEdge Polarity
)
```

Parameters

TrgOutSource (TriggerSource [])

TrgOutSource is the line source for the Trigger Out signal

Polarity (SensitivityEdge)

It is the polarity for the Trigger Out signal. Rising means positive polarity, Falling means negative polarity.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Strobe Vectors

Method (pVectors, pControls)

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PatternGenerator ► StrobeVectors(UInt64, Byte)

C# _

[This is preliminary documentation and is subject to change.]

Strobe of the vectors. Before performing the strobe of the vectors, the user must change the direction of the bidirectional pins.

Declaration Syntax

```
Visual C++
C#
                   Visual Basic
public ATError StrobeVectors(
        ulong pVectors,
        byte pControls
)
Public Function StrobeVectors (
        pVectors As ULong, _
        pControls As Byte
) As ATError
public:
ATError^ StrobeVectors(
        unsigned long long pVectors,
        unsigned char pControls
)
```

Parameters

pVectors (<u>UInt64</u>)

Vectors. pVectors(17 downto 0) represent the value of the output pins, pVectors(31 downto 18) represent the direction of the pins 13..0. If the direction is '1', the pin works as Output, if the direction is '0', it works as Input.

pControls (Byte)

pControls(3 downto 0) represent the direction of the pins 17..14. If the direction is '1', the pin works as Output, if the direction is '0', it works as

Input.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library

Documentation Phase Modulation Law Struct

Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PhaseModulationLawStruct

C#

[This is preliminary documentation and is subject to change.]

Struct of phase modulation law.

■ Declaration Syntax

PhaseValue

C#	Visual Basic	Visual C++	
public struct Phase	eModulationLawS	truct	

Public Structure PhaseModulationLawStruct

public value class PhaseModulationLawStruct

Members

All Me	embers	Methods		Properties	
Public Protected ■ ■ Protected ■ ■ Protected ■ Protected				□ Declared □ Inherited	
Icon Member De		Desc	escription		
≘∳	<u>=qs.a.e(==g==</u>		Indicates whether this instance and a specified object are equal. (Inherited from ValueType .)		
≓	GetHashCode()		Returns the hash code for this instance. (Inherited from ValueType .)		
≅∳	GetType()			the <u>Type</u> of the curre	ent instance.

Value of phase.

	Repetitions	Number of repetitions to be mantained.
=₩	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)
		(Innericed from <u>valuetype</u> r)

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PhaseValue

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PhaseModulationLawStruct ▶ PhaseValue

C#

[This is preliminary documentation and is subject to change.]

Value of phase.

■ Declaration Syntax

```
C# Visual Basic Visual C++

public double PhaseValue { get; set; }

Public Property PhaseValue As Double

public:
property double PhaseValue {
         double get ();
         void set (double value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Repetitions

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PhaseModulationLawStruct</u> ► **Repetitions**

C# -

[This is preliminary documentation and is subject to change.]

Number of repetitions to be mantained.

Declaration Syntax

```
C# Visual Basic Visual C++

public ulong Repetitions { get; set; }

Public Property Repetitions As ULong

public:
property unsigned long long Repetitions {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PLLStatus

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► **PLLStatus**

C#

[This is preliminary documentation and is subject to change.]

Status of PLL.

Declaration Syntax

Visual Basic	Visual C++		
public enum PLLStatus			
Public Enumeration PLLStatus			
public enum class PLLStatus			
	us PLLStatus	us PLLStatus	

Members

Member	Description
Free	PLL are free.
Locked	PLL are locked.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation PSKChannel Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PSKChannel

C#

[This is preliminary documentation and is subject to change.]

Class to manage an phase modulated DDS channel on Arbitrary Waveform Generator device.

Declaration Syntax

C# Visual Basic Visual C++

public class PSKChannel : DDSChannel

Public Class PSKChannel _ Inherits DDSChannel

public ref class PSKChannel: public DDSChannel

Members

All Me	embers	Constructors	Methods	Properti	es	
Pub Prot	lic ected		☑ Instance ☑ Static			Declare Inherite
Icon	Member				Descrip	tion
Ξ ∲	PSKChann	iel()			Initialize PSKChai	es a new nnel clas
	Amplitude	CorrectionFactor			Set amp	
∃	ATXSSSlav	veDisableStartCon	dition()		Disable schannel (Inherite	

=	ATXSSSlaveDisableStopCondition()	Disable stop fror channel in slave
		(Inherited from
∃	ATXSSSlaveEnableStartCondition(ATXSSEvent)	Enable start fror channel in slave line for check sta
		(Inherited from
Ξ ψ	ATXSSSlaveEnableStopCondition(ATXSSEvent)	Enable stop from channel in slave line for check stop
		(Inherited from
	ChannelFunctionality	Get channel fun Pattern)
		(Inherited from
≡	ClearBuffer(DDSBufferType)	Reset the FPGA
		(Inherited from
=	EnableDisable(Boolean)	Enable or disable output on or off.
		(Inherited from
	<u>EnableFlatnessCompensation</u>	Enables/disbles flatness compenuser defined am automatically cointernal compenthe user defined applied as is.
		(Inherited from
=0	Equals(Object)	Determines whe is equal to the c
		(Inherited from
	ErrorResult	Contain the resu

		(Inherited from
≡	GetHashCode()	Serves as a hash type.
		(Inherited from
≡	GetType()	Gets the <u>Type</u> of
		(Inherited from
€	GetVersion(String)	Read from FPGA loaded.
		(Inherited from
=0	<u>LoadCarrier(WaveformStruct)</u>	Load waveform a memory.
		(Inherited from
€₩	<u>LoadPSKModulationLaw(PhaseModulationLawStruct[], TransferMode, Boolean)</u>	Load PSK modul
	MemorySamples	Memory samples
		(Inharitad from
≡	DoodModo(ModuleticaTues)	(Inherited from
	ReadMode(ModulationType)	Get the Modulat
		(Overrides DDSChannel.Rea
≡	ReadMode(Functionality)	Get the Functior
		(Inherited from
≡	ReadStatus(ChannelStatus)	Read status of cl
		(Inherited from
≡⊚		
=•	ResetSoftware()	Make a reset sof
	ResetSoftware()	Make a reset sof (Inherited from
	ResetSoftware() SampligRatePrescaler	

=•	SetAmplitudeProfile(AmplitudeProfileStruct)	Set amplitude pr signal as a funct
		(Inherited from
=	SetCarrierBaseFrequency(Single)	Set frequency ba
=0	<u>SetExternalTrigger(TriggerSource, SensitivityEdge, TriggerAction)</u>	Set Trigger to Ex
≡	SetInternalTrigger()	Set Trigger to In (Inherited from
≡	SetOutputImpedance(OutputImpedance)	Set output impe
=0	SetOutputVoltage(Single)	Set output volta channel must be
		(Inherited from
=0	SetTriggerDelay(UInt32)	Set a start delay (Inherited from
∃	SetTriggerMode(TriggerMode)	Set trigger mode Single, Continuc
≡	CotTriggorOut(TriggorCourgo[] ConsitivityEdgo)	(Inherited from
	SetTriggerOut(TriggerSource[], SensitivityEdge)	Set Trigger out (Inherited from
=0	SetTriggerOutDelay(UInt32)	Set a trigger out
=		(Inherited from
	ToString()	Returns a <u>String</u> current <u>Object</u> .
		(Inherited from

■ Inheritance Hierarchy

Object
ATAWGBase
Channel
DDSChannel
PSKChannel

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

Assembly: ArbStudioSDK (Module: ArbStudioSDK) Version: 3.0.1.0 (3.0.1.0)

4



ArbStudio Control Classes Library Documentation PSKChannel

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>PSKChannel</u> ► **PSKChannel()**

C#	~
U 11	4

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the PSKChannel class.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public PSKChannel	()		
Public Sub New			
<pre>public: PSKChannel()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>

ArbStudio Control Classes Library LeCroy Namespaces ►

Documentation Load PSK Modulation Law Method (samples, mode, clearMemory)

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

PSKChannel ▶

LoadPSKModulationLaw(PhaseModulationLawStruct[], TransferMode, Boolean)

C#

[This is preliminary documentation and is subject to change.]

Load PSK modulation law in the FIFO.

Declaration Syntax

```
C#
                      Visual Basic
                                            Visual C++
public ATError LoadPSKModulationLaw(
         PhaseModulationLawStruct[] samples,
         <u>TransferMode</u> mode,
         bool clearMemory
)
Public Function LoadPSKModulationLaw (
         samples As PhaseModulationLawStruct(), _
         mode As <a href="mailto:TransferMode">TransferMode</a>,
         clearMemory As Boolean
) As ATError
public:
ATError^ LoadPSKModulationLaw(
         array<<u>PhaseModulationLawStruct</u>>^ samples,
         <u>TransferMode</u> mode,
         bool clearMemory
)
```

Parameters

samples (PhaseModulationLawStruct [])

Array of PhaseModulationLawStruct with sample value and number of time interval to be maintained.

mode (TransferMode)

Transfer mode, ReEntrant for dynamic load of modulation law.

clearMemory (Boolean)

True to clear memory before load sequence.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation ReadMode Method

(modulation)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

PSKChannel ► ReadMode(ModulationType)

C#

[This is preliminary documentation and is subject to change.]

Get the Modulation type of channel DDS.

Declaration Syntax

■ Parameters

) override

modulation (ModulationType)

out ModulationType of channel.

■ Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation Set Carrier Base Frequency Method

(frequency)

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>PSKChannel</u> ► **SetCarrierBaseFrequency(Single)**

C#

[This is preliminary documentation and is subject to change.]

Set frequency base of carrier.

Declaration Syntax

Parameters

frequency (Single)

Frequency to be set.

Return Value

ATError class result of operation.

Send comments on this topic to <u>LeCroy Corporation Support</u>

Copyright © 2010, LeCroy Corporation, All rights reserved



Documentation ReEntrantQueueStruct Structure

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ReEntrantQueueStruct

C# __

[This is preliminary documentation and is subject to change.]

Struct for ReEntrant bulk transfers.

Declaration Syntax

C# Visual Basic Visual C++

public struct ReEntrantQueueStruct

Public Structure ReEntrantQueueStruct

public value class ReEntrantQueueStruct

Members

All Members	Methods	Properties	
Public Public		Instance	□ Declared
		Static ■ Static	Inherited

Icon	Member	Description
	<u>Buffer</u>	Array of byte to be transferred.
∄	Equals(Object)	Indicates whether this instance and a specified object are equal. (Inherited from ValueType .)
∃	GetHashCode()	Returns the hash code for this instance. (Inherited from ValueType .)
≡♦	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)

	<u>OperationType</u>	Transfer type: read or write.
	<u>Status</u>	Status of transfer.
∃	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)
	<u>TransferId</u>	Id of transfer.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Buffer Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ReEntrantQueueStruct ► Buffer

C#

[This is preliminary documentation and is subject to change.]

Array of byte to be transferred.

Declaration Syntax

```
C# Visual Basic Visual C++

public byte[] Buffer { get; set; }

Public Property Buffer As Byte()

public:
property array<unsigned char>^ Buffer {
        array<unsigned char>^ get ();
        void set (array<unsigned char>^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation OperationType

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ReEntrantQueueStruct ▶ **OperationType**

C# -

[This is preliminary documentation and is subject to change.]

Transfer type: read or write.

Declaration Syntax

```
C# Visual Basic Visual C++
public TransferTypeDef OperationType { get; set; }

Public Property OperationType As TransferTypeDef

public:
property TransferTypeDef OperationType {
         TransferTypeDef get ();
         void set (TransferTypeDef value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Status Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ReEntrantQueueStruct ► Status

C#

[This is preliminary documentation and is subject to change.]

Status of transfer.

Declaration Syntax

```
C# Visual Basic Visual C++
public TransferStatus Status { get; set; }

Public Property Status As TransferStatus

public:
property TransferStatus Status {
         TransferStatus get ();
         void set (TransferStatus value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TransferId

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

ReEntrantQueueStruct ► **TransferId**

C# -

[This is preliminary documentation and is subject to change.]

Id of transfer.

■ Declaration Syntax

```
C# Visual Basic Visual C++

public int TransferId { get; set; }

Public Property TransferId As Integer

public:
property int TransferId {
    int get ();
    void set (int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation RunningState

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

RunningState

 C#	•
U 11	4

[This is preliminary documentation and is subject to change.]

Run state of instrument

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Runnin	gState		
Public Enumeration	RunningState		
public enum class	RunningState		

Members

Member	Description
Running	Instrument is running
Stopped	Instrument is stopped
ErrorState	Instrument is in error

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation SensitivityEdge

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

SensitivityEdge

 C#	▼ 1
U 11	4

[This is preliminary documentation and is subject to change.]

Sensitive edge for the IN/OUT trigger.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Sensit	ivityEdge		
Public Enumeration	SensitivityEdge		
public enum class	SensitivityEdge		

Members

Member	Description
RisingEdge	Rising edge.
FallingEdge	Falling edge.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation SEQInstruction Code Type Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

SEQInstructionCodeType

C#

[This is preliminary documentation and is subject to change.]

Pattern generator struct for Sequencer.

Declaration Syntax

C#	Visual Basic	Visual C++	
public class SEQIn	structionCodeType		
Public Class SEQIn	structionCodeType		
<pre>public ref class SEQInstructionCodeType</pre>			

Members

All Me	embers	Constructors	Methods	Properties	
Pub Prot Prot	lic ected		☑ Instance ☑ Static		Declare ✓ Inherite
Icon	Member		Description		
≘	SEQInstru	ictionCodeType()		ew instance of the onCodeType class	
	BlockAddr	<u>ess</u>	Block address		
≅ ∳	Equals(Ob	ject)		nether the specific current <u>Object</u> . n <u>Object</u> .)	ed <u>Object</u>
	EventMasl	Κ	Event Mask		

≡	GetHashCode()	Serves as a hash function for a particula type.
		(Inherited from Object.)
≟∳	GetType()	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)
	InputEvent	Input Event. Input Event[140] are DPIO[140], Input Event[15] is the trigger event. The Input Event bits are masked logically in AND with Mask value The expression to evaluate will be: Input Bank AND Mask = Input Event
	LoopMode	Loop Mode
	LoopRepetitions	Number of loop repetitions
	<u>Opcode</u>	Instruction OpCode
	Repetitions	Number of instruction repetitions
∃	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)
	VectorNumber	Vector number

■ Inheritance Hierarchy

Object

SEQInstructionCodeType

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



DocumentationSEQInstructionCodeType

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SEQInstructionCodeType</u> ► **SEQInstructionCodeType()**

C#

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the SEQInstructionCodeType class

Declaration Syntax

C#	Visual Basic	Visual C++	
public SEQInstruct	:ionCodeType()		
Public Sub New			
<pre>public: SEQInstructionCode</pre>	eType()		

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation **BlockAddress**

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>SEQInstructionCodeType</u> ► **BlockAddress**

C#

[This is preliminary documentation and is subject to change.]

Block address

Declaration Syntax

```
C# Visual Basic Visual C++

public ushort BlockAddress { get; set; }

Public Property BlockAddress As UShort

public:
property unsigned short BlockAddress {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation **EventMask**

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>SEQInstructionCodeType</u> ► **EventMask**

C# -

[This is preliminary documentation and is subject to change.]

Event Mask

Declaration Syntax

```
C# Visual Basic Visual C++

public ushort EventMask { get; set; }

Public Property EventMask As UShort

public:
property unsigned short EventMask {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation InputEvent

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>SEQInstructionCodeType</u> ► **InputEvent**

C#

[This is preliminary documentation and is subject to change.]

Input Event. Input Event[14..0] are DPIO[14..0], Input Event[15] is the trigger event. The Input Event bits are masked logically in AND with Mask value. The expression to evaluate will be: Input Bank AND Mask = Input Event

Declaration Syntax

```
C# Visual Basic Visual C++
public ushort InputEvent { get; set; }

Public Property InputEvent As UShort

public:
property unsigned short InputEvent {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, <u>All rights reserved</u>



ArbStudio Control Classes Library Documentation Loop Mode

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>SEQInstructionCodeType</u> ► **LoopMode**

C# -

[This is preliminary documentation and is subject to change.]

Loop Mode

Declaration Syntax

```
C# Visual Basic Visual C++
public SequencerLoopType LoopMode { get; set; }

Public Property LoopMode As SequencerLoopType

public:
property SequencerLoopType LoopMode {
        SequencerLoopType get ();
        void set (SequencerLoopType value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Loop Repetitions

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>SEQInstructionCodeType</u> ► **LoopRepetitions**

C# -

[This is preliminary documentation and is subject to change.]

Number of loop repetitions

■ Declaration Syntax

```
C# Visual Basic Visual C++

public ushort LoopRepetitions { get; set; }

Public Property LoopRepetitions As UShort

public:
property unsigned short LoopRepetitions {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Opcode Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SEQInstructionCodeType</u> ► **Opcode**

C#

[This is preliminary documentation and is subject to change.]

Instruction OpCode

Declaration Syntax

```
C# Visual Basic Visual C++
public SequencerOpcodeType Opcode { get; set; }

Public Property Opcode As SequencerOpcodeType

public:
property SequencerOpcodeType Opcode {
    SequencerOpcodeType get ();
    void set (SequencerOpcodeType value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Repetitions

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ► <u>SEQInstructionCodeType</u> ► **Repetitions**

C#

[This is preliminary documentation and is subject to change.]

Number of instruction repetitions

Declaration Syntax

```
C# Visual Basic Visual C++

public ushort Repetitions { get; set; }

Public Property Repetitions As UShort

public:
property unsigned short Repetitions {
    unsigned short get ();
    void set (unsigned short value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation VectorNumber

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>SEQInstructionCodeType</u> ► **VectorNumber**

C#

[This is preliminary documentation and is subject to change.]

Vector number

Declaration Syntax

```
C# Visual Basic Visual C++

public uint VectorNumber { get; set; }

Public Property VectorNumber As UInteger

public:
property unsigned int VectorNumber {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Sequencer Loop Type Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

SequencerLoopType

C#	•
C II	_

[This is preliminary documentation and is subject to change.]

Pattern generator sequencer loop type

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Sequen	cerLoopType		
Public Enumeration	SequencerLoopType		
public enum class	SequencerLoopType		

Members

Member	Description
NO_LOOP	No loop.
LOOP_END	Loop to end.
LOOP_BEGIN	Loop to begin.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



DocumentationSequencerOpcodeType

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

SequencerOpcodeType

C# -

[This is preliminary documentation and is subject to change.]

Pattern generator sequencer instruction.

■ Declaration Syntax

C#	Visual Basic	Visual C++
public enum Seq	uencerOpcodeType	

Public Enumeration SequencerOpcodeType

public enum class SequencerOpcodeType

Members

Member	Description
SEQ_NOP	No operation.
SEQ_NOP_RESET	No operation and reset event.
SEQ_SYNC_REPEAT_VECTORS	Repeat vectors synchronized with clock generator frequency.
SEQ_WAIT_EVENT	Wait for user event.
SEQ_JUMP_IF_EVENT	Jump if user event occurs.

SEQ_JUMP_TO	Unconditional jump.
SEQ_LOAD_MASK	Load event mask.
SEQ_RETURN	Return.
SEQ_REPEAT_VECTORS	Repeat vectors not synchronized with clock generator frequency.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Setup_LevelCondType

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Setup_LevelCondType

C# -

[This is preliminary documentation and is subject to change.]

Pattern generator indicate whether the trigger condition must be detected on high or a low logic level.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public enum Setup_LevelCondType</pre>			
Public Enumeration Setup_LevelCondType			
<pre>public enum class Setup_LevelCondType</pre>			

Members

Member	Description
TriggerWhenEqual	The trigger event occurs when the state of the evaluated signals is equal to the specified pattern.
TriggerWhenNotEqual	The trigger event occurs when the state of the evaluated signals is different from the specified pattern.
TriggerWhenChanged	The trigger event occurs when the state of the selected signals is different from the reference pattern.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Setup_TriggerCondType

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

Setup_TriggerCondType

C# -

[This is preliminary documentation and is subject to change.]

Pattern generator relationship between trigger conditions on edges and trigge conditions on logic levels.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public enum Setup_TriggerCondType</pre>			
Public Enumeration Setup_TriggerCondType			
<pre>public enum class Setup_TriggerCondType</pre>			

Members

Member	Description
EdgeORLevel	Between an edge condition or a logic level condition is detected.
EdgeANDLevel	Edge condition and a logic level condition are satisfied simultaneously.
EdgeBeforeLevel	Edge condition has been detected at least once before that on logic levels.
EdgeAfterLevel	Edge condition has been detected at least once after that on logic levels.

Always	A trigger condition occurs automatically wher data acquisition starts.
Never	Any setting on edges or on logic levels will be ignored.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SuperCableStruct

Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

SuperCableStruct

 C#	-
•	4

[This is preliminary documentation and is subject to change.]

Pattern generator class for Expansion Bus

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public class Super	·CableStruct		
Public Class Super	CableStruct		
public ref class S	SuperCableStruct		

Members

All Me	embers	Constructors	Methods	Properties	
Publ Prot					Declare Inherite
Icon	Member		Description		
≡ ₩	SuperCabl	eStruct()	Initializes a new SuperCableSti	w instance of the r uct class	
	AsyncCfg				
≟	Equals(Ob	ject)	Determines who is equal to the (Inherited from		d <u>Object</u>
≡	GetHashC	ode()	Serves as a has type.	sh function for a p	particulai

		(Inherited from Object.)
=	GetType()	Gets the <u>Type</u> of the current instance.
		(Inherited from Object.)
	<u>Master</u>	
	<u>StartToPin</u>	
	<u>StopToPin</u>	
∃	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)
	TrigFromSC7	

■ Inheritance Hierarchy

Object

SuperCableStruct

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation SuperCableStruct

Constructor

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SuperCableStruct</u> ➤ **SuperCableStruct()**

C#	- 1
C#	

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the <a>SuperCableStruct class

Declaration Syntax

C#	Visual Basic	Visual C++	
<pre>public SuperCableStruct()</pre>			
Public Sub New			
<pre>public: SuperCableStruct()</pre>			

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation AsyncCfg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SuperCableStruct</u> ► **AsyncCfg**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public byte AsyncCfg { get; set; }

Public Property AsyncCfg As Byte

public:
property unsigned char AsyncCfg {
    unsigned char get ();
    void set (unsigned char value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> Copyright © 2010, <u>LeCroy Corporation</u>, All rights reserved



ArbStudio Control Classes Library Documentation Master Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SuperCableStruct</u> ► **Master**

C# 🔽

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public bool Master { get; set; }

Public Property Master As Boolean

public:
    property bool Master {
        bool get ();
        void set (bool value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation StartToPin

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

<u>SuperCableStruct</u> ► **StartToPin**

C# -

[This is preliminary documentation and is subject to change.]

■ Declaration Syntax

```
C# Visual Basic Visual C++
public Pattern_SC_Connect_Pin StartToPin { get; set; }

Public Property StartToPin As Pattern_SC_Connect_Pin

public:
property Pattern_SC_Connect_Pin StartToPin {
         Pattern_SC_Connect_Pin get ();
         void set (Pattern_SC_Connect_Pin value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation StopToPin

Property

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ► SuperCableStruct ► StopToPin

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public Pattern_SC_Connect_Pin StopToPin { get; set; }

Public Property StopToPin As Pattern_SC_Connect_Pin

public:
property Pattern_SC_Connect_Pin StopToPin {
         Pattern_SC_Connect_Pin get ();
         void set (Pattern_SC_Connect_Pin value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TrigFromSC7

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>SuperCableStruct</u> ► **TrigFromSC7**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public bool TrigFromSC7 { get; set; }

Public Property TrigFromSC7 As Boolean

public:
    property bool TrigFromSC7 {
        bool get ();
        void set (bool value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TransferMode

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TransferMode

C#	

[This is preliminary documentation and is subject to change.]

Bulk transfer mode to USB.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum TransferMode			
Public Enumeration TransferMode			
public enum class	TransferMode		

Members

Member	Description
ReEntrant	After transfer command function return to caller.
NonReEntrant	The function return to caller after the completion of all transfers.

Remarks

For ReEntrant mode the close of pending transfers must be managed by the caller.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TransferStatus

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

TransferStatus

C#	▼
C 11	

[This is preliminary documentation and is subject to change.]

Status of bulk tranfers to the channel.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Transf	erStatus		
Public Enumeration	TransferStatus		
public enum class	TransferStatus		

Members

Member	Description
TransferOK	Transfer is successfully.
TransferPending	Transfer is still pending.
TransferKO	Transfer is failed.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TransferTypeDef

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ►

TransferTypeDef

C#	~
U 11	

[This is preliminary documentation and is subject to change.]

Specify type of bulk transfer.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Transf	^F erTypeDef		
Public Enumeration	n TransferTypeDef		
public enum class	TransferTypeDef		

■ Members

Member	Description
READ_TRANSFER	Transfer type read.
WRITE_TRANSFER	Transfer type write.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerAction

Enumeration

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

TriggerAction

C#	~
C 11	

[This is preliminary documentation and is subject to change.]

Channel action for trigger operation.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Trigge	rAction		
Public Enumeration	TriggerAction		
public enum class	TriggerAction		

■ Members

Member	Description
TriggerIgnore	Ignore. No operation.
TriggerStop	Stop channel.
TriggerStart	Start channel.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Condition Type Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerConditionType

C#	-
	$\overline{}$

[This is preliminary documentation and is subject to change.]

Pattern generator trigger condition type.

Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Trigge	rConditionType		
Public Enumeration	TriggerConditionTy	pe	
public enum class	TriggerConditionTyp	e	

Members

- Manager	
Member	Description
DC	DC level.
HighLevel	High level.
LowLevel	Low level.
RisingEdge	Rising edge.
FallingEdge	Falling edge.
AllEdge	Edge transitions.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Trigger Mode

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerMode

C#		▼
	4	_

[This is preliminary documentation and is subject to change.]

Trigger mode in waveform sequence.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Trigge	rMode		
Public Enumeration	TriggerMode		
public enum class	TriggerMode		

Members

Member	Description
Single	Single.
Continuous	Continuous.
Stepped	Stepped.
Burst	Burst.
Dynamic	Dynamic.

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerSource

Enumeration

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerSource

 C#	_

[This is preliminary documentation and is subject to change.]

Line source for TriggerOUT channel trigger and handled event signal.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public enum Trigge	rSource		
Public Enumeration	TriggerSource		
public enum class	TriggerSource		

Members

Member	Description		
None	None.		
Stop	Stop.		
Start	Start.		
Event_Marker	Marker or PatternEvent.		
DCTriggerIN	Digital Connector Pod		
FPTriggerIN	Front Panel TriggerIN		

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerStruct Class

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct

C#

[This is preliminary documentation and is subject to change.]

Pattern generator struct for trigger

Declaration Syntax

C#	Visual Basic	Visual C++	
public class Trigg	erStruct		

Public Class TriggerStruct

GetType()

public ref class TriggerStruct

■ Me	mbers				
All Me	embers	Constructors	Methods	Properties	
Pub Prot Prot	lic cected		☑ Instance ☑ Static		Declare Inherite
Icon	Member		Description		
∄	TriggerStr	uct()	Initializes a ne TriggerStruct	w instance of the class	
Equals(Object)		Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> . (Inherited from <u>Object</u> .)			
≡	GetHashC	ode()	Serves as a ha type.	sh function for a	particula

(Inherited from Object.)

Gets the <u>Type</u> of the current instance.

		(Inherited from Object.)
	<u>Pretrigger</u>	
a ©	ToString()	Returns a <u>String</u> that represents the current <u>Object</u> . (Inherited from <u>Object</u> .)
	TriggerEdgeType1Reg	
	TriggerEdgeType2Reg	
	TriggerWord1Reg	
	TriggerWord2Reg	
	TriggerWordEnable1Reg	
	TriggerWordEnable2Reg	
	TriggerWordEvent12Reg	

■ Inheritance Hierarchy

Object

TriggerStruct

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerStruct

Constructor

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

TriggerStruct ► TriggerStruct()

C#	- 1
C#	

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the <u>TriggerStruct</u> class

Declaration Syntax

C#	Visual Basic	Visual C++		
<pre>public TriggerStruct()</pre>				
Public Sub New				
<pre>public: TriggerStruct()</pre>				

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Pretrigger

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

<u>TriggerStruct</u> ► **Pretrigger**

C# 🔽

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public uint Pretrigger { get; set; }

Public Property Pretrigger As UInteger

public:
property unsigned int Pretrigger {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Edge Type 1 Reg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerEdgeType1Reg**

C# __

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public ulong TriggerEdgeType1Reg { get; set; }

Public Property TriggerEdgeType1Reg As ULong

public:
    property unsigned long long TriggerEdgeType1Reg {
        unsigned long long get ();
        void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Edge Type 2 Reg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerEdgeType2Reg**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public ulong TriggerEdgeType2Reg { get; set; }

Public Property TriggerEdgeType2Reg As ULong

public:
property unsigned long long TriggerEdgeType2Reg {
    unsigned long long get ();
    void set (unsigned long long value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerWord 1Reg

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerWord1Reg**

C# -

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public uint TriggerWord1Reg { get; set; }

Public Property TriggerWord1Reg As UInteger

public:
property unsigned int TriggerWord1Reg {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation TriggerWord2Reg

Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerWord2Reg**

C# _

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public uint TriggerWord2Reg { get; set; }

Public Property TriggerWord2Reg As UInteger

public:
property unsigned int TriggerWord2Reg {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Word Enable 1 Reg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerWordEnable1Reg**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public uint TriggerWordEnable1Reg { get; set; }

Public Property TriggerWordEnable1Reg As UInteger

public:
property unsigned int TriggerWordEnable1Reg {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Word Enable 2 Reg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerWordEnable2Reg**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++
public uint TriggerWordEnable2Reg { get; set; }

Public Property TriggerWordEnable2Reg As UInteger

public:
property unsigned int TriggerWordEnable2Reg {
    unsigned int get ();
    void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



Documentation Trigger Word Event 12 Reg Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

TriggerStruct ► **TriggerWordEvent12Reg**

C#

[This is preliminary documentation and is subject to change.]

Declaration Syntax

```
C# Visual Basic Visual C++

public uint TriggerWordEvent12Reg { get; set; }

Public Property TriggerWordEvent12Reg As UInteger

public:
    property unsigned int TriggerWordEvent12Reg {
        unsigned int get ();
        void set (unsigned int value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Waveform Struct

Structure

<u>Namespaces</u> ►

ActiveTechnologies.Instruments.AWG4000.Control ▶

WaveformStruct

C#	_
----	---

[This is preliminary documentation and is subject to change.]

Struct of Waveform reprepresentation.

■ Declaration Syntax

C#	Visual Basic	Visual C++	
public struc	t WaveformStruct		
Public Struc	ture WaveformStruct		

public value class WaveformStruct

Marker

■ Me	mbers					
All Members Methods			Properties			
Public Protected Protected						□ Declared □ Inherited
Icon	n Member		Description			
€	Equals(Object)		Indicates whether this instance and a specified object are equal. (Inherited from ValueType .)			
≡∳	GetHashCode()		Returns the hash code for this instance. (Inherited from ValueType .)			
-= (GetType()			Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)		

Position of marker in sample array. Positions must be in ascending order.

	Sample	Sample is array of samples.
€₩	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Marker Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

WaveformStruct ► Marker

C# _

[This is preliminary documentation and is subject to change.]

Position of marker in sample array. Positions must be in ascending order.

Declaration Syntax

```
C# Visual Basic Visual C++

public double[] Marker { get; set; }

Public Property Marker As Double()

public:
property array<double>^ Marker {
        array<double>^ get ();
        void set (array<double>^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



ArbStudio Control Classes Library Documentation Sample Property

<u>Namespaces</u> ►

<u>ActiveTechnologies.Instruments.AWG4000.Control</u> ►

WaveformStruct ► Sample

C# -

[This is preliminary documentation and is subject to change.]

Sample is array of samples.

Declaration Syntax

```
C# Visual Basic Visual C++

public double[] Sample { get; set; }

Public Property Sample As Double()

public:
property array<double>^ Sample {
        array<double>^ get ();
        void set (array<double>^ value);
}
```

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>



[This is preliminary documentation and is subject to change.]

ArbStudio Control Classes Library Documentation

■ Namespaces

Namespace	Description
ActiveTechnologies.Instruments.AWG4000.Control	ArbStudio Control Classes Library contains all object t manage LeCroy Arbitrary Waveform Generator Series ArbStudio 1104 and 1102

Send comments on this topic to <u>LeCroy Corporation Support</u> <u>Copyright © 2010, LeCroy Corporation, All rights reserved</u>