Gamry Electrochemistry Toolkit™

Programmer's Reference Manual

Revision 1.2
February 8, 2008

Gamry Electrochemistry Toolkit Programming Reference

Product Description

Introduction

The Gamry Electrochemical Toolkit provides a powerful way to control your Gamry Instruments potentiostat. Interfaces are made available using a Component Object Model (COM). These interfaces allow for control of both the hardware, as well as signals and data acquisition objects (Dtaqs). The library that exposes these COM interfaces is known as GamryCOM.

Documentation Overview

Gamry offers several products that utilize the Gamry Electrochemistry Toolkit. Each product makes different interfaces available for your use. In this document you will find a description of the interfaces available, as well as descriptions of the methods and events they expose. While this documentation describes all of the currently available interfaces, it does not imply that you have a license to use every interface. Consult your packing slip for your package description as well as any pertinent authorization codes.

In addition to this written document, a number of code samples are included to assist you in developing your own software. This document provides descriptions of these samples.

Sample Programs

DC Demo - CpivRamp

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryDtaqCpiv
- IGamrySignalRamp

The objective of this sample is to show how to create a complete application using the Gamry Electrochemistry Toolkit. The example shows the end-to-end process of loading the Device List, initializing a Pstat, and taking data using the Ramp Signal combined with the CPIV Dtaq.

Please note that as this is a fairly straightforward example, it does not contain much of the error handling or extra features that would benefit a fully functional application.

Installation Location

DC Demo directory of the Gamry Electrochemistry Toolkit installation

Main File

DC Demo – CpivRamp.vi

DC Demo – IVArray

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryDtaqIv
- IGamrySignalArray

The objective of this sample is to show how to create a complete application using the Gamry Electrochemistry Toolkit. The example shows the end-to-end process of loading the Device List, initializing a Pstat, and taking data using the Array Signal combined with the IV Dtaq.

In this example, the Array Signal is only used to create a few constant voltage points; however, this Signal may be used to create any user-defined set of points, as seen in the *VFP600 Source* example.

Please note that as this is a fairly straightforward example, it does not contain much of the error handling or extra features that would benefit a fully functional application.

Installation Location

DC Demo directory of the Gamry Electrochemistry Toolkit installation

Main File

DC Demo – IVArray.vi

DC Demo – Univ

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryDtaqUniv
- IGamrySignalUniv

The objective of this sample is to show how to create a complete application using the Gamry Electrochemistry Toolkit. The example shows the end-to-end process of loading the Device List, initializing a Pstat, and taking data using the Univ Signal combined with the Univ Dtaq.

Much like the *DC Demo – IvArray* example, the Univ Signal is only used to create a few constant voltage points; however, this Signal may be used to create any user-defined set of points. Please see the appropriate section in this document for details on using the Univ Signal.

Please note that as this is a fairly straightforward example, it does not contain much of the error handling or extra features that would benefit a fully functional application.

Installation Location

DC Demo directory of the Gamry Electrochemistry Toolkit installation

Main File

DC Demo – Univ.vi

AC Demo

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryReadZ

The objective of this sample is to show how to create a complete application using the Gamry Electrochemistry Toolkit. The example shows the end-to-end process of loading the Device List, initializing a Pstat, and taking a single AC Impedance measurement with the ReadZ class.

Please note that as this is a fairly straightforward example, not only does it not contain much of the error handling or extra features that would benefit a fully functional application, it also does not utilize many of the features of the ReadZ class. For a detailed example of a complete application, please see the *LVEIS* example.

Installation Location

ACDemo directory of the Gamry Electrochemistry Toolkit installation

Main File

AC Demo.vi

VFP600 Source

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryDtaqIV
- IGamrySignalArray

The main object of this sample is to show how to build a more advance application using the Gamry Electrochemistry Toolkit. This sample is the original source code for the Virtual Front Panel software package. The sample makes use of simple IGamryDtaqIV Dtaq as well as the IGamrySignalArray. These two interfaces allow a wide variety of data acquisition modes and signal waveforms. This sample also shows how to handle dynamic device changes.

Installation Location

VFP600 directory of the Gamry Electrochemistry Toolkit installation

Main File

VFP.vi

LVEIS

Language

LabVIEW 7.1

Description

This sample uses the following interfaces via LabVIEW:

- IGamryDeviceList
- IGamryPstat
- IGamryReadZ

The main object of this sample is to show how to run AC Impedance spectra using the IGamryReadZ interface. This sample is somewhat complex and one should first become familiar with the ACDemo.

Installation Location

LVEIS directory of the Gamry Electrochemistry Toolkit installation

Main File

LVEIS.vi

VCClient

Language

Visual C++ 6.0

Description

This sample uses the following interfaces via C++:

- IGamryDeviceList
- IGamryPstat
- IGamrySignalConst
- IGamryDtaqIvt

The object of this sample is to first, create a device list and check the number of instruments available. If there is at least one instrument available, it then creates a GamryPstat object and uses the first instrument. A constant signal is then created along with an IVT dtaq. The dtaq is run and the output is then shown in a console window.

Installation Location

C++ directory of the Gamry Electrochemistry Toolkit installation

Main File

VCClient.dsp

Excel IVTUniv

Language

Excel 2003/VBA

Description

This sample uses the following interfaces via VBA:

- IGamryDeviceList
- IGamryPstat
- IGamrySignalUniv
- IGamryDtaqIvt

The object of this sample is to first, create a device list and check the number of instruments available. If there is at least one instrument available, it then enables data acquisition using the IVT dtaq and the Universal Signal. The signal is output from data entered on an Excel spreadsheet. The collected data is placed onto the spreadsheet and then graphed on a chart.

Installation Location

Excel directory of the Gamry Electrochemistry Toolkit installation

Main File

Excel IVTUniv.xls

Typedefs and Enums

Overview

The GamryCOM Interface Library contains the following typedefs and enums.

Member	Description
gcCELLSTATE	State of the cell switch
gcCTRLMODE	Control mode for the potentiostat
gcICONVENTION	Current measurement convention
gcGROUND	Ground Isolation Switch
gcSENSESPEED	I/E Electrometer
gcIRUPTMODE	IR Correction Modes
gcEUCALCMODE	EU Calculation Modes
gcMODELNO	Device Model Number
gcIESTABILITY	I/E Converter Stability
gcDTAQCHRONO_TYPE	DtaqChrono Measurement Type
gcDTAQEISSTATUS	Detailed status of EIS Dtaq
gcREADZSPEED	ReadZ Speed
gcREADZSTATUS	Status of ReadZ Measurement
gcFC350IDIVISOR	FC350 I channel Divisor
gcSIGTWEAKSTYLE	Signal tweak style
gcESAFILTERORDER	DtaqEsa Filter Order

gcCELLSTATE Enum

State of the cell switch

Definition

```
enum
gcCELLSTATE{
  CellOff = 0,
  CellOn
  } gcCELLSTATE;
```

See Also

Cell
SetCell

gcCTRLMODE Enum

Control mode for the potentiostat

Definition

```
enum
gcCTRLMODE{
   GstatMode = 0,
   PstatMode,
   ZRAMode,
   FRAMode
   } gcCTRLMODE;
```

See Also

CtrlMode
SetCtrlMode

gcICONVENTION Enum

Current measurement convention

Definition

```
enum
gcICONVENTION{
   Cathodic = 0,
   Anodic
   } gcICONVENTION;
```

See Also

IConvention
SetIConvention

gcGROUND Enum

Ground isolation switch

Definition

```
enum
gcGROUND{
   Float = 0,
   Earth
   } gcGROUND;
```

See Also

Ground
SetGround

gcSENSESPEED Enum

I/E Electrometer

Definition

```
enum
gcSENSESPEED{
   SenseFast = 0,
   SenseSlow
   } gcSENSESPEED;
```

See Also

SenseSpeed
SetSenseSpeed

gcIRUPTMODE Enum

IR Correction Modes

Definition

```
enum
gcIRUPTMODE{
   IruptOff= 0,
   IruptNorm,
   IruptClfg
   } gcIRUPTMODE;
```

See Also

SetIruptMode

gcEUCALCMODE Enum

EU Calculation Modes

Definition

```
enum
gcEUCALCMODE{
   EuNone = 0,
   EuExtrap,
   EuAverage
   } gcEUCALCMODE;
```

See Also

SetIruptMode

gcMODELNO Enum

Device Model Number

Definition

```
enum
gcMODELNO{
   PC4300   = 32,
   PC4750   = 33,
   PCI4300  = 34,
   PCI4750  = 35,
   FAS1     = 16,
   FC350    = 17,
   FAS2     = 18,
   FCI350   = 19,
   PC5600   = 4,
   PCI4G300 = 36,
   PCI4G750 = 37,
   FCIG350  = 21
   } gcMODELNO;
```

See Also

ModelNo

gcIESTABILITY Enum

I/E Converter Stability

Definition

```
enum
gcIESTABILITY{
   StabilityFast= 0,
   StabilityMedFast,
   StabilityNorm,
   StabilitySlow
   } gcIESTABILITY;
```

See Also

IEStability
SetIEStability

gcDTAQCHRONO_TYPE Enum

DtaqChrono Measurement Type

Definition

```
enum
gcDTAQCHRONO_TYPE{
   ChronoAmp = 0,
   ChronoCoul,
   ChronoPot
   } gcDTAQCHRONO_TYPE;
```

See Also

DtaqChrono Init

gcDTAQEISSTATUS Enum

Detailed status of EIS Dtaq

Definition

```
enum
gcDTAQEISSTATUS{
   DtaqEISStatusInvalid = 0,
   DtaqEISStatusDelay,
   DtaqEISStatusMeasuring,
   DtaqEISStatusMeasOk,
   DtaqEISStatusCommErr,
   DtaqEISStatusTimeout,
   DtaqEISStatusCycleLim,
   DtaqEISStatusControl,
   DtaqEISStatusOverrun,
   DtaqEISStatusOverrange,
   DtaqEISStatusOverrunQ,
   DtaqEISStatusRetry
   } gcDTAQEISSTATUS;
```

See Also

DtaqEis Result

gcREADZSPEED Enum

ReadZ Speed

Definition

```
enum
gcREADZSPEED{
  ReadZSpeedFast= 0,
  ReadZSpeedNorm,
  ReadZSpeedLow
  } gcREADZSPEED;
```

See Also

ReadZ SetSpeed

gcREADZSTATUS Enum

Status of ReadZ Measurement

Definition

```
enum
gcREADZSTATUS{
  ReadZStatusOk= 0,
  ReadZStatusRetry,
  ReadZStatusError
  } gcREADZSTATUS;
```

See Also

IGamryReadZ::OnDataDone

gcFC350IDIVISOR Enum

FC350 I channel divisor

Definition

```
enum
gcFC350IDIVISOR{
  FC350IDivisor1= 0,
  FC350IDivisor10,
  FC350IDivisor100
  } gcFC350IDIVISOR;
```

See Also

FC350IDivisor
SetFC350IDivisor

gcSIGTWEAKSTYLE Enum

Signal tweak style

Definition

```
enum
gcSIGTWEAKSTYLE{
   SigTweakStyleReset= 0,
   SigTweakStyleContinue,
   SigTweakStyleScale
   } gcSIGTWEAKSTYLE;
```

gcESAFILTERORDER Enum

DtaqEsa filter order.

Definition

```
enum
gcESAFILTERORDER{
  EsaFilterOrder17 = 0,
  EsaFilterOrder43,
  EsaFilterOrder68
  } gcESAFILTERORDER;
```

IGamryDeviceList

Overview

The IGamryDeviceList interface is exported by the GAMRYCOM library. It enables applications to determine information about the Gamry devices connected to the system.

Member	Description
Count	Retrieve the number of devices connected
EnumSections	Enumerate the sections of the connected devices
EnumLabels	Enumerate the labels of the connected devices

Count Method

This function returns a count of Gamry devices which are currently installed or connected to the system and controlled by GamryCom.

Definition

```
HRESULT
Count(
   [out,retval] long* Count
   );
```

Parameter

`Count`

A variable that receives the device count.

Comments

The count returned by this function can be used to determine if any devices are currently installed or connected to the system. A count of 0 means that no devices are currently connected or installed, and the application should handle this accordingly.

EnumSections Method

This function returns an array containing the section identifier for each Gamry device currently installed or connected to the system and controlled by GamryCom.

Definition

```
HRESULT
EnumSections(
   [out,retval] SAFEARRAY(BSTR)* Sections
   );
```

Parameter

`Sections`

An array that receives the list of section identifiers.

Comments

The list returned by this function is used to identify which devices are to be used by the application. A device is specified by using its section identifier in the call to Init.

See Also

IGamryPstat.Init

EnumLabels Method

This function returns an array containing the labels for each Gamry device currently installed or connected to the system and controlled by GamryCom.

Definition

```
HRESULT
EnumLabels(
   [out,retval] SAFEARRAY(BSTR)* Labels
   );
```

Parameter

Labels
An array that receives the list of device labels

Comments

The list returned by this function can be used to present the user with friendly names for the available devices rather than the section identifiers. A device cannot be specified using its label, but there is a direct correlation between the labels returned from **EnumLabels** and the section identifiers returned from **EnumSections**.

See Also

EnumSections

_IGamryDeviceListEvents

Overview

The _IGamryDeviceListEvents interface provides the event call-back functions required to handle events issued by GamryDeviceList objects.

The following table summarizes the members of _IGamryDeviceListEvents. The methods are described in detail in this section.

Member	Description
OnDeviceListChanged	A device was either added or removed from the system.

OnDeviceListChanged Method

This function is called when a change has occurred to the device list. This change is usually a device addition or removal.

Definition

```
HRESULT
OnDeviceListChanged();
```

Comments

This event is fired based on notification GamryCom receives from the operating system. Most firings of this event are a result of a device being added or removed from the system. However, on some occasions, this event can be fired multiple times for the same device change. Because of this behavior, the application should check for changes to the list returned by **EnumSections**. If the list has changed, a device has been either added or removed. If the list is the same, the event can be disregarded.

See Also

EnumSections

IGamryPstat

Overview

The IGamryPstat interface is exported by the GAMRYCOM library. It enables applications to control a potentiostat device which is connected to the system.

Member	Description
Init	Initializes the Pstat object
Label	The label name of the Pstat object
ModelNo	The model number of the Pstat object
Open	Open the Pstat
Close	Close the Pstat
SetSignal	Set the active Signal Object
InitSignal	Initialize the active Signal Object
SetVoltage	Set the cell voltage of the Pstat
SetBias	Set the BIAS DAC
SetScan	Set the SCAN DAC
SetScanRange	Set the range of the SCAN DAC
TestScanRange	Determine the appropriate range of the SCAN DAC
GstatRatio	Determine the Current/Voltage ratio of an I/E Range
IruptMode	Specify the current interrupt mode
CalDate	Read the calibration date
SetCalDate	Write the calibration date
Cell	Read the cell state
SetCell	Set the cell state
CtrlMode	Read the control mode setting
SetCtrlMode	Set the control mode
CASpeed	Read the control amp speed setting
SetCASpeed	Set the control amp speed
TestCASpeed	Test the appropriate control amp speed
IEStability	Read the I/E converter stability setting
SetIEStability	Set the I/E converter stability
IConvention	Read the Current (I) convention
SetIConvention	Set the Current (I) convention
Ground	Read the ground relay setting
SetGround	Set the ground relay
SenseSpeed	Read the current I/E electrometer setting
SetSenseSpeed	Set which I/E electrometer to use
SenseSpeedMode	Read the auto-select mode of the SenseSpeed setting
SetSenseSpeedMode	Set the auto-select mode of the SenseSpeed setting
PosFeedEnable	Read the enable state of Positive Feedback mode
SetPosFeedEnable	Set the enable stat of Positive Feedback mode
PosFeedResistance	Read the Positive Feedback Resistance setting
SetPosFeedResistance	Set the Positive Feedback Resistance

AnalogOut	Read the Analog out setting
SetAnalogOut	Set the Analog Out DAC value
DigitalOut	Read the Digital Out bits setting
SetDigitalOut	Set the Digital Out bits
DigitalIn	Read the Digital In bits
DDSEnable	Read the enable state of the DDS synthesizer
SetDDSEnable	Set the enable state of the DDS synthesizer
DDSAmpl	Read the amplitude of the DDS synthesizer
SetDDSAmpl	Set the amplitude of the DDS synthesizer
DDSAmplBits	Read the bit resolution of the DDS synthesizer
SetDDSAmplBits	Set the bit resolution of the DDS synthesizer
DDSFreq	Read the frequency of the DDS synthesizer
SetDDSFreq	Set the frequency of the DDS synthesizer
FreqLimitUpper	Read the upper AC frequency limit of the Pstat
FreqLimitLower	Read the lower AC frequency limit of the Pstat
IERange	Read the I/E converter range setting
SetIERange	Set the I/E converter range
TestIERange	Test the appropriate I/E converter range
TestIERangeAC	Test the appropriate I/E converter range with bandwidth
IERangeMode	Read the auto-range mode of the I/E converter
SetIERangeMode	Set the auto-range mode of the I/E converter
IERangeLowerLimit	Read the lower limit (range) setting for the I/E converter
SetIERangeLowerLimit	Set the lower limit (range) of the I/E converter
IEResistor	Read the effective resistance of an I/E converter Range
FindIERange	Find an appropriate I/E converter Range
VchRange	Read the voltage channel range setting
SetVchRange	Set the voltage channel range setting
TestVchRange	Test the appropriate voltage channel range
TestVchRangeAC	Test the appropriate voltage channel range with bandwidth
VchRangeMode	Read the auto-range mode of the voltage channel
SetVchRangeMode	Set the auto-range mode of the voltage channel
VchOffset	Read the voltage channel A/D offset setting
SetVchOffset	Set the voltage channel A/D offset
VchOffsetEnable	Read the enable state of the voltage channel offset DAC
SetVchOffsetEnable	Set the enable state of the voltage channel offset DAC
VchFilter	Read the voltage channel filter setting
SetVchFilter	Set the voltage channel filter
TestVchFilter	Test the appropriate voltage channel filter
FindVchRange	Find an appropriate voltage channel range
IchRange	Read the current channel range setting
SetIchRange	Set the current channel range setting
TestIchRange	Test the appropriate current channel range
TestIchRangeAC	Test the appropriate current channel range with bandwidth
IchRangeMode	Read the auto-range mode of the current channel
SetIchRangeMode	Set the auto-range mode of the current channel

IchOffset	Read the current channel A/D offset setting
SetIchOffset	Set the current channel A/D offset
IchOffsetEnable	Read the enable state of the current channel offset DAC
SetIchOffsetEnable	Set the enable state of the current channel offset DAC
IchFilter	Read the current channel filter setting
SetIchFilter	Set the current channel filter
TestIchFilter	Test the appropriate current channel filter
FindIchRange	Find an appropriate current channel range
AchRange	Read the auxiliary channel range setting
SetAchRange	Set the auxiliary channel range setting
TestAchRange	Test the appropriate auxiliary channel range
TestAchRangeAC	Test the appropriate auxiliary channel range with bandwidth
AchRangeMode	Read the auto-range mode of the auxiliary channel
SetAchRangeMode	Set the auto-range mode of the auxiliary channel
AchOffset	Read the auxiliary channel A/D offset setting
SetAchOffset	Set the auxiliary channel A/D offset
AchOffsetEnable	Read the enable state of the auxiliary channel offset DAC
SetAchOffsetEnable	Set the enable state of the auxiliary channel offset DAC
AchFilter	Read the auxiliary channel filter setting
SetAchFilter	Set the auxiliary channel filter
TestAchFilter	Test the appropriate auxiliary channel filter
FindAchRange	Find an appropriate auxiliary channel range
MeasureV	Read the voltage at the cell input
MeasureI	Read the current at the cell input
MeasureA	Read the voltage on the potentiostat's Auxiliary A/D input
ScanLimitAC	Report the maximum RMS signal the Pstat can deliver
FC350VRange	Report the current FC350 voltage range
SetFC350VRange	Set the current FC350 voltage range
TestFC350VRange	Test the appropriate FC350 voltage range
FC350VRangeMultiplier	Report the current FC350 voltage range multiplier
FC350IDivisor	Report the current FC350 I channel divisor
SetFC350IDivisor	Set the current FC350 I channel divisor

Init Method

This function initializes a GamryPstat object and specifies the physical device to use.

Definition

```
HRESULT
Init(
   [in] BSTR Section
   );
```

Parameter

Section
A section identifier for the device to use

Comments

The **Init** method is used to specify the physical device associated with the GamryPstat object. This section identifier should be one that is returned from a call to the **EnumSections** method of **IGamryDeviceList**.

See Also

IGamryDeviceList.EnumSections

Label Method

This function returns the label, or friendly name, of the Pstat.

Definition

```
HRESULT
Label(
   [out,retval] BSTR* Label
   );
```

Parameter

Label
A variable that receives the label.

Comments

The **Label** method is used primarily to return the friendly name of the Pstat so the user can easily recognize the device.

ModelNo Method

This function returns the model number of the Pstat.

Definition

```
HRESULT
ModelNo(
   [out,retval] gcMODELNO* ModelNo
   );
```

Parameter

ModelNo
A variable that receives the model number.

Comments

The **ModelNo** method is used to return which type of Pstat is represented by this object.

See Also

gcMODELNO

SerialNo Method

This function returns the serial number of the Pstat.

Definition

```
HRESULT
SerialNo(
   [out,retval] BSTR* SerialNo
   );
```

Parameter

SerialNo
A variable that receives the serial number.

Comments

The **SerialNo** is the unit serial number for external instruments, or the control board number for internal instruments.

Open Method

This function opens the Pstat.

Definition

```
HRESULT
Open();
```

Comments

The **Open** method is used to take control of the Pstat. Any functions which set a Pstat parameter require the Pstat to be open. A Pstat can only be opened by one application at a time. If more that one application tries to **Open** a Pstat, only the first one will succeed.

See Also

Close

Close Method

This function closes an open Pstat.

Definition

```
HRESULT
Close(
   [in,defaultvalue(VARIANT_TRUE)] VARIANT_BOOL* Safe
   );
```

Parameter

Safe
A Boolean value specifying whether or not to verify if the device can be closed.

Comments

The **Close** method is used to release an open Pstat. Once a Pstat is closed, other applications can make use of it. By default this function does not require any parameters. The optional **Safe** parameter should be left in the default state.

See Also

Open

SetSignal Method

Specifies which signal object is to be used by the Pstat.

Definition

```
HRESULT
SetSignal(
   [in] IGamrySignal* Signal
   );
```

Parameter

Signal
A Gamry Signal object.

Comments

Multiple signals can be pre-created and then used one at a time by making a call to **SetSignal**. The Signal can be one of many types of signals exposed by GamryCom, like GamrySignalConst.

See Also

IGamrySignal

InitSignal Method

Initialize the signal which is currently being used by the Pstat.

Definition

```
HRESULT
InitSignal();
```

Comments

This method causes the signal to be reset to the beginning so any data acquisition objects run after the call to **InitSignal** will start with the beginning of the applied signal. This call should be made to make sure the signal being used by the Pstat is in a known state.

SetVoltage Method

Set the cell voltage of the Pstat.

Definition

```
HRESULT
SetVoltage(
   [in] float Voltage
   );
```

Parameter

Voltage
The voltage to apply (working versus reference)

Comments

This function provides a way to easily set the cell voltage when in Potentiostat mode. This function does not use or allow for current interrupt IR compensation. The voltage will not be applied unless the cell switch is on. This function should not be used when the hardware is in Galvanostat mode without prior considerations like knowing the GstatRatio.

See Also

SetCell
SetCtrlMode
GstatRatio
SetBias
SetScan

SetBias Method

Set the applied bias voltage.

Definition

```
HRESULT
SetBias(
   [in] float Voltage,
   [out,retval] float BiasSet
   );
```

Parameter

Voltage
Bias to be applied (Volts)

BiasSet
Bias as set (Volts)

Comments

This signal is first summed with the Scan, DDS signal, and External signal. It is then sent to the potentiostat's signal input. In potentiostat mode this sum is applied to the cell. The voltage will not be applied unless the cell switch is on. In galvanostat mode the applied current depends on the GstatRatio.

See Also

SetCell
GstatRatio
SetVoltage
SetScan

SetScan Method

Set the scan voltage.

Definition

```
HRESULT
SetScan(
   [in] float Voltage,
   [out,retval] float VoltageSet
   );
```

Parameter

Voltage
Bias to be applied (Volts)

VoltageSet
Scan as set (Volts)

Comments

This signal is first summed with the Bias, DDS signal, and External signal. It is then sent to the potentiostat's signal input. In potentiostat mode this sum is applied to the cell. The voltage will not be applied unless the cell switch is on. In galvanostat mode the applied current depends on the GstatRatio.

See Also

SetCell
GstatRatio
SetBias
SetVoltage

SetScanRange Method

Set the voltage range of the scan DAC.

Definition

```
HRESULT
SetScanRange(
   [in] VARIANT ScanRange,
   [out,retval] long* ScanRangeSet
   );
```

Parameter

`ScanRange`

ScanRange is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestScanRange call prior to setting the scan range. This floating point number is the voltage range over which the scan DAC is to operate. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the scan range based on an integer identifier.

0 = ScanRangeCoarse
1 = ScanRangeMed
2 = ScanRangeFine

`ScanRangeSet`

Range as set

0 = ScanRangeCoarse
1 = ScanRangeMed
2 = ScanRangeFine

Comments

When a signal is attached to the potentiostat, the scan range is automatically set. However, if you intend to set the scan voltage manually using **SetScan**, a call to **SetScanRange** should be used beforehand.

See Also

SetScan

TestScanRange Method

Test for a scan DAC voltage range.

Definition

```
HRESULT
TestScanRange(
   [in] float Voltage1,
   [in] float Voltage2,
   [out,retval] long* ScanRange
   );
```

Parameter

Voltage1
First voltage of overall scan range

Voltage2
Second voltage of overall scan range

ScanRange
Range which should be used

0 = ScanRangeCoarse
1 = ScanRangeMed
2 = ScanRangeFine

Comments

The scan range is calculated by taking the difference between Voltage1 and Voltage2. If an appropriate scan range can be determined it will be returned, otherwise, an error will be generated.

See Also

SetScanRange

GstatRatio Method

Return the number of amps/volt when in galvanostat mode.

Definition

```
HRESULT
GstatRatio(
   [in] long IERange,
   [out,retval] float* Ratio
   );
```

Parameter

IERange
A valid I/E converter range number

Ratio
Amps/(Input Volt) in galvanostat mode

Comments

This function is sensitive to sign convention since amps are defined differently by convention. It is used primarily for calibration, but can also be used to configure external or manual signal sources.

See Also

SetIConvention

SetIruptMode Method

Set the current interrupt IR compensation mode.

Definition

```
HRESULT
SetIruptMode(
   [in] gcIRUPTMODE FbMode,
   [in,defaultvalue(EuNone)] gcEUCALCMODE EuMode,
   [in,defaultvalue(0.0)] float Timer,
   [in,defaultvalue(0.0)] float Eoc,
   [in,defaultvalue(1.0)] float Gain
   );
```

Parameter

FbMode

Feedback mode

IruptOff = NoFeedback
IruptNorm = Normal Feedback
IruptClfg = Control Loop Fixed Gain

EuMode

IR error calculation mode

EuNone = No interrupt measurement
EuExtrap = Extrapolation
EuAverage = Average

Timer

Nominal τ value (in seconds). This is the shorted time. The default value is normally around 50 µs.

Eoc

Open circuit voltage, used only in control loop modes.

Gain

Cell gain. Used only in fixed gain control loop.

Comments

The current interrupt feature of the potentiostat uses a 3 sample algorithm to estimate the IR-free voltage. It first samples V at the given current level then it interrupts the current path and waits a time period, Tau. At that point it samples

V again. It waits Tau seconds again and samples V for the third time. It then turns on the current.

The sample time, Tau, is increased as the current decreases. The Time parameter used in the call to **SetIruptMode** is used directly as Tau on the least sensitive current ranges. On the more sensitive current ranges, Tau is longer, but remains proportional to the Time parameter. Check your potentiostat's operating manual for information on valid Tau values.

There are two different ways of calculating the IR voltage error. In calculation Mode 1, the three voltages, V_i, V_{oc1}, and V_{oc2} are used to calculate an IR error voltage V_{IR} via an extrapolation. In Mode 2, an average is used.

$V_{IR} = V_i - 2 \cdot V_{oc1} + V_{oc2}$ (Calculation Mode 1)

$V_{IR} = V_i - \frac{1}{2}(V_{oc1} + V_{oc2})$ (Calculation Mode 2)

A third "calculation mode", Mode 0, is used to turn off the interrupt. In Mode 0, all of the other function parameters are ignored.

Once the IR error has been calculated, it can be used in several different ways, most of which involve feedback of the error signal.

In Galvanostat mode, there is never any IR compensation feedback. The feedback mode setting is therefore ignored in the Galvanostat mode. The measured error is used to correct potential measurements stored in the data curve.

In Potentiostat mode, there are 3 different feedback modes.

a) No feedback.
b) Normal feedback. The measured error voltage is added to the voltage applied for the next point.
c) Control loop with fixed gain.

See Also

SetPosFeedEnable

CalDate Method

Report the current calibration date setting.

Definition

```
HRESULT
CalDate(
   [in] long CalType,
   [out,retval] BSTR* CalDate
   );
```

Parameter

CalType
Type of calibration to return date

0 = DC Calibration Date
1 = AC Calibration Date

CalDate
Date of last calibration

Comments

This function is used to check the calibration date, and potentially warn the user if the calibration is too old, or if the unit is not calibrated.

See Also

SetCalDate

SetCalDate Method

Set the calibration date

Definition

```
HRESULT
SetCalDate(
   [in] long CalType,
   [in] BSTR* CalDate
   );
```

Parameter

CalType
Type of calibration

0 = DC Calibration
1 = AC Calibration

CalDate
Calibration date to set

Comments

This function is used by the calibration, and should not be used under any other conditions.

See Also

CalDate

Cell Method

Return the current state of the cell switch.

Definition

```
HRESULT
Cell(
   [out,retval] gcCELLSTATE* CellState
   );
```

Parameter

CellState
Cell switch state

CellOff
CellOn

Comments

Since the reference electrode is always connected, measurements of the open circuit voltage can be made even if the cell switch is in the off position.

See Also

SetCell

SetCell Method

Set the state of the cell switch.

Definition

```
HRESULT
SetCell(
   [in] gcCELLSTATE CellState,
   [out,retval] gcCELLSTATE* CellStateSet
   );
```

Parameter

CellState
Cell switch state to set

CellOff
CellOn

CellStateSet
Cell switch state as set

Comments

When the cell is in the off position, no cell current can flow. The counter electrode is disconnected.

When the cell is in the on position, cell current may flow. The counter electrode is connected to the cell.

See Also

Cell

CtrlMode Method

Return the current potentiostat control mode.

Definition

```
HRESULT
CtrlMode(
   [out,retval] gcCTRLMODE* CtrlMode
   );
```

Parameter

CtrlMode
Current Control Mode

GstatMode = Galvanostat
PstatMode = Potentiostat
ZRAMode = Zero Resistance Ammeter (ZRA)
FRAMode = Frequency Response Analyzer (FRA)

See Also

SetCtrlMode

SetCtrlMode Method

Set the potentiostat control mode.

Definition

```
HRESULT
SetCtrlMode(
   [in] gcCTRLMODE CtrlMode,
   [out,retval] gcCTRLMODE* CtrlModeSet
   );
```

Parameter

CtrlMode
Control Mode to set

GstatMode = Galvanostat
PstatMode = Potentiostat
ZRAMode = Zero Resistance Ammeter (ZRA)
FRAMode = Frequency Response Analyzer (FRA)

CtrlModeSet
Control Mode as set

Comments

FRAMode should normally not be set using this method.

See Also

CtrlMode

CASpeed Method

Return the control amplifier speed setting.

Definition

```
HRESULT
CASpeed(
   [out,retval] long* CASpeed
   );
```

Parameter

CASpeed
Control Amplifier speed setting. Refer to your potentiostat's operating manual for information on the available speed settings.

See Also

SetCASpeed
TestCASpeed

SetCASpeed Method

Set the roll off filter on the potentiostat control amplifier.

Definition

```
HRESULT
SetCASpeed (
   [in] VARIANT CASpeed,
   [out,retval] long* CASpeedSet
   );
```

Parameter

CASpeed

CASpeed is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestCASpeed call prior to setting the speed. The floating point number is the frequency of interest when the unit is in potentiostat mode. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the speed based on an integer identifier. Consult your potentiostat's operating manual for information on the available CA Speed settings.

CASpeedSet

Control amplifier speed setting as set

Comments

The control amplifier roll-off affects the overall stability of the potentiostat. As with the IEStability disclaimer, we can only offer you a guideline in setting the CASpeed.

If the potentiostat oscillates on all current ranges change the setting.

See Also

CASpeed
TestCASpeed
SetIEStability

TestCASpeed Method

Test the speed setting of the potentiostat control amplifier.

Definition

```
HRESULT
TestCASpeed (
   [in] float Frequency,
   [out,retval] long* CASpeed
   );
```

Parameter

Frequency
The frequency of interest when the unit is in potentiostat mode.

CASpeed
Appropriate control amplifier speed setting.

Comments

The frequency bandwidth for the control amplifier is dependent upon the control mode of the potentiostat. When using this function, it expects the frequencies to be entered as if the unit were in potentiostat mode. Galvanostat mode will have an increased bandwidth.

See Also

CASpeed
SetCASpeed

IEStability Method

Report the potentiostat I/E converter stability setting.

Definition

```
HRESULT
IEStability(
   [out,retval] gcIESTABILITY* Stability
   );
```

Parameter

Stability

Current I/E Stability Setting

StabilityFast
StabilityMedFast
StabilityNorm
StabilitySlow

See Also

SetIEStability

SetIEStability Method

Set the I/E converter stability for potentiostat mode.

Definition

```
HRESULT
SetIEStability (
   [in] gcIESTABILITY Stability,
   [out,retval] gcIESTABILITY* StabilitySet
   );
```

Parameter

Stability

I/E converter stability setting to set.

StabilityFast
StabilityMedFast
StabilityNorm
StabilitySlow

StabilitySet

I/E converter stability setting as set

Comments

The StabilityFast value corresponds to no extra I/E converter filtering. The most stable stability is StabilitySlow. The Slow value corresponds to a large I/E filter which removes 50/60 Hz noise in the lower current ranges.

While it would take an advanced course in amplifier design to completely understand the stability setting, we can offer you a few guidelines:

a) Always leave the stability in Fast for galvanostatic operation.
b) If the potentiostat is showing high speed oscillation that depends on the current range in use, try increasing the Stability setting.
c) If your curves are noisy at low currents, try setting Stability to Slow.
d) If you are acquiring data faster than 0.2 seconds/point and see glitches when the current range changes, try decreasing the stability.

See Also

CASpeed
TestCASpeed
SetIEStability

IConvention Method

Return the setting of the current (I) convention.

Definition

```
HRESULT
Convention(
   [out,retval] gcICONVENTION* Convention
   );
```

Parameter

Convention
Current convention setting

Cathodic
Anodic

Comments

The setting refers to which direction is positive. When the convention is Anodic, anodic currents are considered positive. When the convention is Cathodic, cathodic currents are considered positive.

See Also

SetIConvention

SetIConvention Method

Set the state of the cell switch.

Definition

```
HRESULT
SetIConvention(
   [in] gcICONVENTION Convention,
   [out,retval] gcICONVENTION* ConventionSet
   );
```

Parameter

Convention

Convention to set

Anodic
Cathodic

ConventionSet

Cell switch state as set

Comments

The setting refers to which direction is positive. When the convention is Anodic, anodic currents are considered positive. When the convention is Cathodic, cathodic currents are considered positive.

See Also

IConvention

Ground Method

Read the current state of the ground isolation switch.

Definition

```
HRESULT
Ground(
   [out,retval] gcGROUND* Ground
   );
```

Parameter

Ground

Current state of the ground isolation switch

Float – Potentiostat floating ground is isolated from earth ground
Earth – Potentiostat floating ground is connected to earth ground

Comments

This setting does not pertain to all instruments. Please consult your potentiostat's operating manual for information regarding the ability to connect float and earth grounds.

See Also

SetGround

SetGround Method

Set the state of the ground isolation switch.

Definition

```
HRESULT
SetGround(
   [in] gcGROUND Ground,
   [out,retval] gcGROUND* GroundSet
   );
```

Parameter

Ground

Ground state to set

Float – Potentiostat floating ground is isolated from earth ground
Earth – Potentiostat floating ground is connected to earth ground

GroundSet

Ground switch state as set

Comments

This setting does not pertain to all instruments. Please consult your potentiostat's operating manual for information regarding the ability to connect float and earth grounds.

See Also

Ground

SenseSpeed Method

Return which I/E electrometer (Fast or Slow) is currently being used.

Definition

```
HRESULT
SenseSpeed(
   [out,retval] gcSENSESPEED* SenseSpeed
   );
```

Parameter

SenseSpeed

Current state of the sense speed

SenseSlow – Slow I/E and Electrometer (more accuracy)
SenseFast – Fast I/E and Electrometer (more speed)

Comments

This setting pertains only to FAS1 and FAS2 model instruments.

See Also

SetSenseSpeed
SetSenseSpeedMode

SetSenseSpeed Method

Sets which I/E electrometer (Fast or Slow) is to be used.

Definition

```
HRESULT
SetSenseSpeed(
   [in] gcSENSESPEED SenseSpeed,
   [out,retval] gcSENSESPEED* SenseSpeedSet
   );
```

Parameter

SenseSpeed

Sense speed to set

SenseSlow – Slow I/E and Electrometer (more accuracy)
SenseFast – Fast I/E and Electrometer (more speed)

SenseSpeedSet

Sense speed state as set

Comments

This setting pertains only to FAS1 and FAS2 model instruments.

See Also

SenseSpeed
SetSenseSpeedMode

SenseSpeedMode Method

Returns the status of the Sense Speed auto-set mode.

Definition

```
HRESULT
SenseSpeedMode(
   [out,retval] VARIANT_BOOL* SenseSpeedMode
   );
```

Parameter

SenseSpeedMode
Current state of the sense speed

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

Comments

This setting pertains only to FAS1 and FAS2 model instruments.

See Also

SetSenseSpeed
SetSenseSpeedMode

SetSenseSpeedMode Method

Sets the Sense Speed auto-set mode.

Definition

```
HRESULT
SetSenseSpeed(
   [in] VARIANT_BOOL SenseSpeedMode,
   [out,retval] VARIANT_BOOL* SenseSpeedModeSet
   );
```

Parameter

SenseSpeedMode
Sense speed mode to set

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

SenseSpeedModeSet
Sense speed mode as set

Comments

This setting pertains only to FAS1 and FAS2 model instruments. Most applications should leave this in the enabled state. This allows speed settings to be automatically adjusted based on I/E current range selections. In cases where DC measurements are being performed across wide current range selections, a user may choose to turn off the auto-set mode and manually specify the slow sense speed.

See Also

SetSenseSpeed
SenseSpeedMode

PosFeedEnable Method

Return the enable state of the Positive Feedback DAC

Definition

```
HRESULT
PosFeedEnable(
   [out,retval] VARIANT_BOOL* PosFeedEnable
   );
```

Parameter

PosFeedEnable

Current enable state of the positive feedback DAC

VARIANT_TRUE – Enabled
VARIANT_FALSE – Disabled

Comments

Positive feedback is enabled when performing positive feedback IR compensation.

See Also

SetPosFeedEnable
SetPosFeedResistance

SetPosFeedEnable Method

Sets the enable state of the positive feedback DAC.

Definition

```
HRESULT
SetPosFeedEnable(
   [in] VARIANT_BOOL PosFeedEnable,
   [out,retval] VARIANT_BOOL* PosFeedEnableSet
   );
```

Parameter

PosFeedEnable

Enable state to set

VARIANT_TRUE – Enable
VARIANT_FALSE – Disable

PosFeedEnableSet

Positive feedback enable state as set

Comments

Positive feedback is enabled when performing positive feedback IR compensation. A call should also be made to SetPosFeedResistance to set the uncompensated resistance that will be used by the feedback loop.

See Also

PosFeedEnable
SetPosFeedResistance

PosFeedResistance Method

Return the uncompensated resistance value used for positive feedback.

Definition

```
HRESULT
PosFeedResistance(
   [out,retval] float* PosFeedResistance
   );
```

Parameter

PosFeedResistance
Value of the uncompensated resistance in ohms.

Comments

This value is not used unless the positive feedback mode is enabled.

See Also

SetPosFeedEnable
SetPosFeedResistance

SetPosFeedResistance Method

Sets the uncompensated resistance value used during positive feedback.

Definition

```
HRESULT
SetPosFeedResistance(
   [in] float PosFeedResistance,
   [out,retval] float* PosFeedResistanceSet
   );
```

Parameter

PosFeedResistance
Value of the uncompensated resistance to set in ohms

PosFeedResistanceSet
Value of the uncompensated resistance as set.

Comments

Positive feedback is enabled when performing positive feedback IR compensation. The positive feedback mode should be enabled prior to making this call.

See Also

SetPosFeedEnable
PosFeedResistance

AnalogOut Method

Read the current voltage setting for the auxiliary DAC output.

Definition

```
HRESULT
AnalogOut(
   [out,retval] float* Out
   );
```

Parameter

Out
The current auxiliary voltage in volts.

See Also

SetAnalogOut

SetAnalogOut Method

Sets the voltage of the auxiliary DAC output.

Definition

```
HRESULT
SetAnalogOut(
   [in] float Out,
   [out,retval] float* OutSet
   );
```

Parameter

Out
Auxiliary voltage to set in volts.

OutSet
Auxiliary voltage as set.

Comments

For some potentiostats, the offset and/or full scale range of the auxiliary DAC can be changed using jumpers on the controller card. Consult your potentiostat's operating manual for more information.

See Also

AnalogOut
SetDigitalOut

DigitalOut Method

Read the current digital output setting.

Definition

```
HRESULT
DigitalOut(
   [out,retval] short* Out
   );
```

Parameter

Out
The current digital output setting in the 4 least significant bits.

See Also

SetDigitalOut

SetDigitalOut Method

Sets the digital output setting.

Definition

```
HRESULT
SetDigitalOut(
   [in] short Out,
   [in] short Mask,
   [out,retval] short* OutSet
   );
```

Parameter

Out
New output setting in the lowest 4 bits.

Mask
Identify the bits to be changed.

OutSet
Output setting as set.

Comments

The `Out` parameter is used to show the desired bit pattern and to show the range of bits to be changed.

Each of the lowest 4 bits of the `Out` argument corresponds to one of the output bits. A bit will only be changed if the mask argument has a one in that bit's bit position. A mask argument of 0x0003 would allow changes in Output0 and Output1. With this mask value, Output 2 and Output 3 will not be changed regardless of the bit pattern argument.

See Also

DigitalOut

DigitalIn Method

Read the potentiostat's digital input bits.

Definition

```
HRESULT
DigitalIn(
   [out,retval] short* In
   );
```

Parameter

In
The input bits. In0 is the least significant bit.

Comments

There are 4 uncommitted digital inputs in the Miscellaneous I/O Connector. The bits are labeled In0, In1, In2 and In3. See your potentiostat's Operator's Manual for the pin assignments in the Miscellaneous I/O Connector.

See Also

SetDigitalOut

DDSEnable Method

Return the enable state of the DDS sine wave synthesizer.

Definition

```
HRESULT
DDSEnable(
   [out,retval] VARIANT_BOOL* DDSEnable
   );
```

Parameter

DDSEnable

Current enable state of the DDS sine wave synthesizer

VARIANT_TRUE – Enabled
VARIANT_FALSE – Disabled

See Also

SetDDSEnable
SetDDSAmpl
SetDDSFreq

SetDDSEnable Method

Sets the enable state of the DDS sine wave synthesizer.

Definition

```
HRESULT
SetDDSEnable(
   [in] VARIANT_BOOL DDSEnable,
   [out,retval] VARIANT_BOOL* DDSEnableSet
   );
```

Parameter

DDSEnable

Enable state to set

VARIANT_TRUE – Enable
VARIANT_FALSE – Disable

DDSEnableSet

DDS enable state as set

Comments

The DDS enable state must be set to enable for the DDS sine wave synthesizer to be summed in through the signal chain. In order for the signal to be applied at the cell, the cell state must also be turned on. Prior to setting the DDS to the enable state, calls should be made to SetDDSFreq and SetDDSAmpl.

See Also

DDSEnable
SetDDSFreq
SetDDSAmpl
SetCell

DDSAmpl Method

Report the current RMS voltage of the DDS sine wave synthesizer.

Definition

```
HRESULT
DDSAmpl(
   [out,retval] float* DDSAmpl
   );
```

Parameter

DDSAmpl
The current RMS voltage in volts.

See Also

SetDDSAmpl

SetDDSAmpl Method

Sets the RMS voltage of the DDS sine wave synthesizer.

Definition

```
HRESULT
SetDDSAmpl(
   [in] float DDSAmpl,
   [out,retval] float* DDSAmplSet
   );
```

Parameter

DDSAmpl
Requested RMS amplitude in volts.

DDSAmplSet
Resulting RMS amplitude as set.

Comments

This amplitude will only be applied if the DDS is enabled with a call to SetDDSEnable.

For a sine wave $V_{rms} = \frac{V_{peak}}{\sqrt{2}} = \frac{V_{pk2pk}}{2\sqrt{2}}$.

The peak voltages that a potentiostat can produce vary by instrument. Consult your operator's manual for specifications.

See Also

DDSAmpl
SetDDSEnable
SetDDSFreq

DDSAmplBits Method

Read the bit resolution of the DDS synthesizer.

Definition

```
HRESULT
DDSAmplBits(
   [out,retval] float* DDSAmplBits
   );
```

Parameter

DDSAmplBits
DDS Synthesizer's bit resolution

Comments

Certain Gamry Instrument's devices have different bit resolution for the DDS synthesizer. These are due to changes in the DDS circuitry itself. This function is used during the calibration of the instruments.

See Also

SetDDSAmplBits

SetDDSAmplBits Method

Set the bit resolution of the DDS synthesizer.

Definition

```
HRESULT
SetDDSAmpl(
   [in] float DDSAmplBits,
   [out,retval] float* DDSAmplBitsSet
   );
```

Parameter

DDSAmplBits
Requested DDS bit resolution.

DDSAmplSet
DDS bit resolution as set.

Comments

Certain Gamry Instrument's devices have different bit resolution for the DDS synthesizer. These are due to changes in the DDS circuitry itself. This function is used during the calibration of the instruments.

See Also

DDSAmpl
SetDDSEnable
SetDDSFreq

DDSFreq Method

Report the current frequency setting of the DDS sine wave synthesizer.

Definition

```
HRESULT
DDSFreq(
   [out,retval] float* DDSFreq
   );
```

Parameter

DDSFreq
The current frequency setting in hertz.

See Also

SetDDSFreq

SetDDSFreq Method

Sets the frequency of the DDS sine wave synthesizer.

Definition

```
HRESULT
SetDDSFreq(
   [in] float DDSFreq,
   [out,retval] float* DDSFreqSet
   );
```

Parameter

DDSFreq
Requested frequency in hertz.

DDSFreqSet
Resulting frequency as set.

Comments

This frequency will only be applied if the DDS is enabled with a call to SetDDSEnable. This call should be made in conjunction with a call to SetDDSAmpl. The upper and lower limits vary based on the instrument. Consult your operator's manual for specifications.

See Also

SetDDSAmpl
SetDDSEnable
DDSFreq

FreqLimitUpper Method

Report the potentiostat's maximum allowed frequency.

Definition

```
HRESULT
FreqLimitUpper(
   [out,retval] float* FreqLimitUpper
   );
```

Parameter

FreqLimitUpper
Maximum frequency in hertz.

Comments

Each potentiostat driver reports the limitation imposed by its associated hardware. Consult your potentiostat's operator's manual for specifications.

See Also

FreqLimitLower

FreqLimitLower Method

Report the potentiostat's minimum allowed frequency.

Definition

```
HRESULT
FreqLimitLower(
   [out,retval] float* FreqLimitLower
   );
```

Parameter

FreqLimitLower
Minimum frequency in hertz.

Comments

Each potentiostat driver reports the limitation imposed by its associated hardware. Consult your potentiostat's operator's manual for specifications.

See Also

FreqLimitUpper

IERange Method

Report the current I/E range of the potentiostat.

Definition

```
HRESULT
IERange(
   [out,retval] long* Range
   );
```

Parameter

Range

I/E Range of the potentiostat

Absolute current range (Full Scale Limit)
0 = 3 pA
1 = 30 pA
2 = 300 pA
3 = 3 nA
4 = 30 nA
5 = 300 nA
6 = 3 µA
7 = 30 µA
8 = 300 µA
9 = 3 mA
10 = 30 mA
11 = 300 mA
12 = 3 A
13 = 30 A
14 = 300 A
15 = 3 kA

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat. The ranges listed are for 300 mA models. For 750 mA models, multiply the ranges by 2.5. For 600 mA models, multiply the ranges by 2.0.

See Also

SetIERange

SetIERange Method

Sets the I/E Range of the potentiostat.

Definition

```
HRESULT
SetIERange (
   [in] VARIANT Range,
   [out,retval] long* RangeSet
   );
```

Parameter

Range

`Range` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestIERange call prior to setting the range. The floating point number is the current (in Amps) of interest. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the I/E range based on an integer identifier. Consult your potentiostats operating manual for information on the available I/E Range settings.

RangeSet

Resulting range as set.

0 = 3 pA
1 = 30 pA
2 = 300 pA
3 = 3 nA
4 = 30 nA
5 = 300 nA
6 = 3 µA
7 = 30 µA
8 = 300 µA
9 = 3 mA
10 = 30 mA
11 = 300 mA
12 = 3 A
13 = 30 A
14 = 300 A
15 = 3 kA

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat. The ranges listed are for 300 mA models. For 750 mA models, multiply the ranges by 2.5. For 600 mA models, multiply the ranges by 2.0. If you attempt to set a range which is not available for a specific potentiostat, a Parameter Error will be issued.

Some ranges may be unavailable if the LowerLimit has been set. See the SetIERangeLowerLimit method for more information.

See Also

IERange
TestIERange
SetIERangeLowerLimit

TestIERange Method

Find the optimum I/E range setting of the potentiostat for a given current.

Definition

```
HRESULT
TestIERange (
   [in] float Current,
   [out,retval] long* Range
   );
```

Parameter

Current
The maximum anticipated current.

Range
Appropriate I/E range setting.

Comments

If Abs(Current) is greater than Imax for the selected potentiostat, the maximum possible range for that potentiostat will be returned.

See Also

IERange
SetIERange

TestIERangeAC Method

Find the optimum I/E range setting of the potentiostat for a given current and frequency.

Definition

```
HRESULT
TestIERangeAC (
   [in] float CurrentAC,
   [in] float VoltageAC,
   [in] float CurrentDC,
   [in] float VoltageDC,
   [in] float Frequency,
   [out,retval] long* Range
   );
```

Parameter

CurrentAC
The maximum absolute AC current in amps

VoltageAC
The maximum absolute AC voltage in volts

CurrentDC
The maximum absolute AC current in amps

VoltageDC
The maximum absolute DC voltage in volts

Frequency
Measurement frequency in hertz

Range
Appropriate I/E range setting.

Comments

This function takes into account the measurement frequency since a higher than normal current range is required at high frequencies. It also considers the maximum estimated voltage to avoid very large common mode voltages. The actual current range is not changed.

See Also

SetIERange

IERangeMode Method

Returns the status of auto-ranging mode of the I/E converter.

Definition

```
HRESULT
IERangeMode(
   [out,retval] VARIANT_BOOL* RangeMode
   );
```

Parameter

RangeMode
Current state of the auto-ranging mode

VARIANT_TRUE – Auto-ranging is enabled
VARIANT_FALSE – Auto-ranging is disabled

See Also

SetIERange
SetIERangeMode

SetIERangeMode Method

Enable or disable current measurement auto-ranging.

Definition

```
HRESULT
SetIERange(
   [in] VARIANT_BOOL RangeMode,
   [out,retval] VARIANT_BOOL* RangeModeSet
   );
```

Parameter

RangeMode
Auto-range mode to set

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

RangeModeSet
Sense speed mode as set

Comments

This setting pertains only to FAS1 and FAS2 model instruments. Most applications should leave this in the enabled state. This allows speed settings to be automatically adjusted based on I/E current range selections. In cases where DC measurements are being performed across wide current range selections, a user may choose to turn off the auto-set mode and manually specify the slow sense speed.

See Also

SetIERange
IERangeMode

IERangeLowerLimit Method

Reads the lower limit of the IERange as currently set.

Definition

```
HRESULT
IERangeLowerLimit(
   [out,retval] long* RangeLowerLimit
   );
```

Parameter

RangeLowerLimit

Lower Limit IERange as set

0 = 3 pA
1 = 30 pA
2 = 300 pA
3 = 3 nA
4 = 30 nA
5 = 300 nA
6 = 3 µA
7 = 30 µA
8 = 300 µA
9 = 3 mA
10 = 30 mA
11 = 300 mA
12 = 3 A
13 = 30 A
14 = 300 A
15 = 3 kA

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat. The ranges listed are for 300 mA models. For 750 mA models, multiply the ranges by 2.5. For 600 mA models, multiply the ranges by 2.0. If you attempt to set a range which is not available for a specific potentiostat, a Parameter Error will be issued.

See Also

SetIERange
SetIERangeLowerLimit

SetIERangeLowerLimit Method

Sets the lower limit of the IERange.

Definition

```
HRESULT
SetIERangeLowerLimit(
   [in] long RangeLowerLimit,
   [out,retval] long* RangeLowerLimitSet
   );
```

Parameter

RangeLowerLimit

The IERange to be used as the lower limit by the potentiostat

0 = 3 pA
1 = 30 pA
2 = 300 pA
3 = 3 nA
4 = 30 nA
5 = 300 nA
6 = 3 µA
7 = 30 µA
8 = 300 µA
9 = 3 mA
10 = 30 mA
11 = 300 mA
12 = 3 A
13 = 30 A
14 = 300 A
15 = 3 kA

RangeLowerLimitSet

Resulting range lower limit as set.

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat. The ranges listed are for 300 mA models. For 750 mA models, multiply the ranges by 2.5. For 600 mA models, multiply the ranges by 2.0. If you attempt to set a range which is not available for a specific potentiostat, a Parameter Error will be issued.

This limit is used when the potentiostat is making autorange decisions. By setting this lower limit, one can insure that the potentiostat will never range to a range lower than that specified by the limit. This is helpful when doing measurements at high speed.

See Also

SetIERange
SetIERangeLowerLimit

IEResistor Method

Calculate the effective I/E resistance.

Definition

```
HRESULT
IEResistor(
   [in] long Range,
   [out,retval] float* Resistor
   );
```

Parameter

Range
Current IERange

Resistor
Effective Resistance (Ohms)

Comments

This resistance, when multiplied by the cell current, gives the voltage output of the I/E converter.

See Also

SetIERange

FindIERange Method

Attempts to find optimum current range for use during a scan.

Definition

```
HRESULT
FindIERange(
   [out,retval] long* IERange
   );
```

Parameter

IERange
Range as determined

Comments

Acquires several current points, adjusting the current range after each point is taken. Use this method prior to starting a scan, in order to hone in on the optimum current range to use during the scan.

See Also

IERange
SetIERange

VchRange Method

Report the voltage channel range

Definition

```
HRESULT
VchRange(
   [out,retval] long* Range
   );
```

Parameter

Range

Voltage for 30000 counts on A/D

0	0.03 V range
1	0.30 V range
2	3.00 V range
3	30.00 V (PCI4) 12V (PC5)

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat

See Also

SetVchRange

SetVchRange Method

Set voltage channel range.

Definition

```
HRESULT
SetVchRange(
   [in] VARIANT Range,
   [out,retval] long* RangeSet
   );
```

Parameter

Range

`Range` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestVchRange call prior to setting the range. The floating point number is the maximum anticipated voltage (in Volts). Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage range based on an integer identifier. Consult your potentiostat's operating manual for information on the available I/E Range settings.

RangeSet

VchRange as set

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat

See Also

VchRange
TestVchRange

TestVchRange Method

Finds the optimum voltage range of the voltage channel.

Definition

```
HRESULT
TestVchRange(
   [in] float Voltage,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Range
Optimum voltage range for V channel

Comments

Finds the optimum voltage range of the voltage channel to maximize the resolution of the A/D converter.

See Also

VchRange
SetVchRange

TestVchRangeAC Method

Finds the optimum voltage range of the voltage channel.

Definition

```
HRESULT
TestVchRangeAC(
   [in] float Voltage,
   [in] float Frequency,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Frequency
Measurement Frequency

Range
Optimum voltage range for V channel

Comments

Finds the optimum voltage range of the voltage channel to maximize the resolution of the A/D converter. Takes into account a higher-than-normal measurement frequency.

See Also

VchRange
SetVchRange
TestVchRange

VchRangeMode Method

Returns the status of auto-ranging mode of the V channel.

Definition

```
HRESULT
VchRangeMode(
   [out,retval] VARIANT_BOOL* RangeMode
   );
```

Parameter

RangeMode
Current state of the auto-ranging mode

VARIANT_TRUE – Auto-ranging is enabled
VARIANT_FALSE – Auto-ranging is disabled

See Also

SetVchRange
SetVchRangeMode

SetVchRangeMode Method

Enable or disable current measurement auto-ranging.

Definition

```
HRESULT
SetVchRange(
   [in] VARIANT_BOOL RangeMode,
   [out,retval] VARIANT_BOOL* RangeModeSet
   );
```

Parameter

RangeMode

Auto-range mode to set

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

RangeModeSet

Vch Range mode as set

Comments

Autoranging is performed by the DTAQ in use based upon the last measurement taken. Ranging only occurs if a higher or lower range is available.

See Also

SetVchRange
VchRangeMode

VchOffset Method

Report the offset voltage of the voltage channel

Definition

```
HRESULT
VchOffset(
   [out,retval] float* Offset
   );
```

Parameter

Offset
The offset voltage

Comments

The offset voltage is subtracted using the offset DAC in front of the A/D converter. This offset value is then added back to the measured value after the measurement is taken. By using offsets, the channel can be set to a more sensitive range for off-zero measurements.

See Also

SetVchOffset
VchOffsetEnable
SetVchOffsetEnable

SetVchOffset Method

Set the DC offset voltage in the voltage channel

Definition

```
HRESULT
SetVchOffset(
   [in] float Offset,
   [out,retval] float* OffsetSet
   );
```

Parameter

Offset
Offset voltage level

OffsetSet
The offset voltage as set

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the VchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn't support the DC offset, 0 will also be returned.

See Also

VchOffset
VchOffsetEnable
SetVchOffsetEnable

VchOffsetEnable Method

Reports enabled/disabled state of the voltage channel

Definition

```
HRESULT
VchOffsetEnable(
   [out,retval] VARIANT_BOOL* OffsetEnable
   );
```

Parameter

OffsetEnable

Offset state to set

VARIANT_TRUE – Offset is enabled
VARIANT_FALSE – Offset is disabled

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the VchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn’t support the DC offset, 0 will also be returned.

See Also

VchOffset
VchOffsetEnable
SetVchOffsetEnable

SetVchOffsetEnable Method

Enable or Disable post potentiostat offset correction.

Definition

```
HRESULT
SetVchOffsetEnable(
   [in] VARIANT_BOOL OffsetEnable,
   [out,retval] VARIANT_BOOL* OffsetEnableSet
   );
```

Parameter

OffsetEnable

Enable state to set

VARIANT_TRUE – Enable
VARIANT_FALSE – Disable

OffsetEnableSet

Vch Offset enable state as set

Comments

Enable or Disable post potentiostat offset correction. This allows the DC component of a measured voltage signal to be removed. Removal of this DC component allows multiplication of the signal by a post gain factor (1x, 10x, 100x) while keeping this post gain signal within the +/- 3V limit of the analog to digital converter (ADC).

See Also

VchOffset
SetVchOffset
VchOffsetEnable

VchFilter Method

Report current setting of Voltage Filter

Definition

```
HRESULT
VchFilter(
   [out,retval] long* Filter
   );
```

Parameter

Filter

Filter as set

0	No Filter
1	200 kHz
2	1 kHz
3	5 Hz

Comments

This filter has no effect on potentiostat stability.

See Also

SetVchFilter
TestVchFilter

SetVchFilter Method

Sets the filter on the voltage measuring channel

Definition

```
HRESULT
SetVchFilter(
   [in] VARIANT Filter,
   [out,retval] long* FilterSet
   );
```

Parameter

Filter

`Filter` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestVchFilter call prior to setting the range. The floating point number is the desired frequency. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage filter based on an integer identifier. Consult your potentiostat's operating manual for information on the available V channel filter settings.

Filter

Filter as set

Comments

The No Filter setting should not be used under normal circumstances. It is to be used exclusively for AC impedance measurements beyond 100 kHz..

See Also

VchFilter
TestVchFilter

TestVchFilter Method

Returns the appropriate Vch Filter given a desired frequency

Definition

```
HRESULT
TestVchFilter(
   [in] float Frequency,
   [out,retval] long* Filter
   );
```

Parameter

Frequency
Frequency of interest in Hz

Filter
Best filter setting to use for the specified frequency. See VchFilter.

Comments

This function is automatically called by SetVchFilter is called with a frequency passed in as the first argument.

See Also

VchFilter
SetVchFilter

FindVchRange Method

Attempts to find optimum voltage range for use during a scan.

Definition

```
HRESULT
FindVchRange(
   [out,retval] long* VchRange
   );
```

Parameter

VchRange
Range as determined

Comments

Acquires several current points, adjusting the voltage measurement range after each point is taken. Use this method prior to starting a scan, in order to hone in on the optimum voltage range to use during the scan.

See Also

VchRange
SetVchRange

MeasureV Method

Read the voltage on the cell input

Definition

```
HRESULT
MeasureV(
   [out,retval] float* Voltage
   );
```

Parameter

`Voltage`
Cell voltage in volts.

Comments

The full scale range on will depend on the voltage range currently selected. The MeasureV method will autorange the voltage range if voltage channel autoranging is enabled.

See Also

SetVchRange
SetVchRangeMode

IchRange Method

Report the current channel range

Definition

```
HRESULT
IchRange(
   [out,retval] long* Range
   );
```

Parameter

Range

Voltage for 30000 counts on A/D

0	0.03 V range
1	0.30 V range
2	3.00 V range
3	30.00 V (PCI4) 12V (PC5)

Comments

The measured current is converted into a voltage using the I/E Converter. This function reports the code for the full scale A/D range in use.

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat

See Also

SetIchRange

SetIchRange Method

Set current channel range.

Definition

```
HRESULT
SetIchRange(
   [in] VARIANT Range,
   [out,retval] long* RangeSet
   );
```

Parameter

Range

`Range` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestIchRange call prior to setting the range. The floating point number is the maximum anticipated voltage (in Volts). Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage range based on an integer identifier. Consult your potentiostat's operating manual for information on the available I/E Range settings.

RangeSet

IchRange as set

Comments

Setting the IchRange using a voltage is preferred. The measured current is converted into a voltage on the I channel using the I/E converter.

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat

See Also

IchRange
TestIchRange

TestIchRange Method

Finds the optimum voltage range of the current channel.

Definition

```
HRESULT
TestIchRange(
   [in] float Voltage,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Range
Optimum voltage range for I channel

Comments

Finds the optimum voltage range of the current channel to maximize the resolution of the A/D converter.

See Also

IchRange
SetIchRange

TestIchRangeAC Method

Finds the optimum voltage range of the current channel.

Definition

```
HRESULT
TestIchRangeAC(
   [in] float Voltage,
   [in] float Frequency,
   [out,retval] long* Range
   );
```

Parameter

`Voltage`
Maximum anticipated voltage

`Frequency`
Measurement Frequency

`Range`
Optimum voltage range for I channel

Comments

Finds the optimum voltage range of the current channel to maximize the resolution of the A/D converter. Takes into account a higher-than-normal measurement frequency.

See Also

IchRange
SetIchRange
TestIchRange

IchRangeMode Method

Returns the status of auto-ranging mode of the I channel.

Definition

```
HRESULT
IchRangeMode(
   [out,retval] VARIANT_BOOL* RangeMode
   );
```

Parameter

RangeMode
Current state of the auto-ranging mode

VARIANT_TRUE – Auto-ranging is enabled
VARIANT_FALSE – Auto-ranging is disabled

See Also

SetIchRange
SetIchRangeMode

SetIchRangeMode Method

Enable or disable current measurement auto-ranging.

Definition

```
HRESULT
SetIchRange(
   [in] VARIANT_BOOL RangeMode,
   [out,retval] VARIANT_BOOL* RangeModeSet
   );
```

Parameter

RangeMode
Auto-range mode to set

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

RangeModeSet
Ich Range mode as set

Comments

Autoranging is performed by the DTAQ in use based upon the last measurement taken. Ranging only occurs if a higher or lower range is available.

See Also

SetIchRange
IchRangeMode

IchOffset Method

Report the offset voltage of the current channel

Definition

```
HRESULT
IchOffset(
   [out,retval] float* Offset
   );
```

Parameter

Offset
The offset voltage

Comments

The offset voltage is subtracted using the offset DAC in front of the A/D converter. This offset value is then added back to the measured value after the measurement is taken. By using offsets, the channel can be set to a more sensitive range for off-zero measurements.

See Also

SetIchOffset
IchOffsetEnable
SetIchOffsetEnable

SetIchOffset Method

Set the DC offset voltage in the current channel

Definition

```
HRESULT
SetIchOffset(
   [in] float Offset,
   [out,retval] float* OffsetSet
   );
```

Parameter

Offset
Offset voltage level

OffsetSet
The offset voltage as set

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the IchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn't support the DC offset, 0 will also be returned.

See Also

IchOffset
IchOffsetEnable
SetIchOffsetEnable

IchOffsetEnable Method

Reports enabled/disabled state of the current channel

Definition

```
HRESULT
IchOffsetEnable(
  [out,retval] VARIANT_BOOL* OffsetEnable
  );
```

Parameter

OffsetEnable
Offset state to set

VARIANT_TRUE – Offset is enabled
VARIANT_FALSE – Offset is disabled

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the IchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn't support the DC offset, 0 will also be returned.

See Also

IchOffset
IchOffsetEnable
SetIchOffsetEnable

SetIchOffsetEnable Method

Enable or Disable post potentiostat offset correction.

Definition

```
HRESULT
SetIchOffsetEnable(
   [in] VARIANT_BOOL OffsetEnable,
   [out,retval] VARIANT_BOOL* OffsetEnableSet
   );
```

Parameter

OffsetEnable

Enable state to set

VARIANT_TRUE – Enable
VARIANT_FALSE – Disable

OffsetEnableSet

Ich Offset enable state as set

Comments

Enable or Disable post potentiostat offset correction. This allows the DC component of a measured voltage signal to be removed. Removal of this DC component allows multiplication of the signal by a post gain factor (1x, 10x, 100x) while keeping this post gain signal within the +/- 3V limit of the analog to digital converter (ADC). The effective resistance of the I/E Resistor must be taken into account to determine the voltage seen by the ADC for a corresponding current level.

See Also

IchOffset
SetIchOffset
IchOffsetEnable

IchFilter Method

Report current IchFilter setting

Definition

```
HRESULT
IchFilter(
   [out,retval] long* Filter
   );
```

Parameter

Filter

Filter as set

0	No Filter
1	200 kHz
2	1 kHz
3	5 Hz

Comments

This is a different filter than the I/E Filter described in the SetStability section. It has no effect on potentiostat stability, only on the current measuring circuit's frequency response.

See Also

SetIchFilter
TestIchFilter

SetIchFilter Method

Sets the 2 pole Butterworth low pass filter on the current measuring channel

Definition

```
HRESULT
SetIchFilter(
   [in] VARIANT Filter,
   [out,retval] long* FilterSet
   );
```

Parameter

Filter

`Filter` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestIchFilter call prior to setting the range. The floating point number is the desired frequency. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the filter based on an integer identifier. Consult your potentiostat's operating manual for information on the available I channel filter settings.

Filter

Filter as set

Comments

This is a different filter than the I/E Filter described in the SetStability section. It has no effect on potentiostat stability, only on the current measuring circuit's frequency response.

The No Filter setting should not be used under normal circumstances. It is to be used exclusively for AC impedance measurements beyond 100 kHz..

See Also

IchFilter
TestIchFilter

TestIchFilter Method

Returns the appropriate Ich Filter given a desired frequency

Definition

```
HRESULT
TestIchFilter(
   [in] float Frequency,
   [out,retval] long* Filter
   );
```

Parameter

Frequency
Frequency of interest in Hz

Filter
Best filter setting to use for the specified frequency. See IchFilter.

Comments

This function is automatically called by SetIchFilter is called with a frequency passed in as the first argument.

See Also

IchFilter
SetIchFilter

FindIchRange Method

Attempts to find optimum current range for use during a scan.

Definition

```
HRESULT
FindIchRange(
   [out,retval] long* IchRange
   );
```

Parameter

IchRange
Range as determined

Comments

Acquires several current points, adjusting the current range after each point is taken. Use this method prior to starting a scan, in order to hone in on the optimum current range to use during the scan.

See Also

IchRange
SetIchRange

MeasureI Method

Read the current at the cell input.

Definition

```
HRESULT
MeasureI(
   [out,retval] float* Current
   );
```

Parameter

Current
Current at the cell input in amps.

Comments

The full scale range on the current input will depend on the IERange selected and the current channel range. The MeasureI method will autorange the I/E range selection if I/E autoranging is enabled.

See Also

SetIchRange
SetIERange
SetIERangeMode

AchRange Method

Report the range of the Auxiliary A/D input.

Definition

```
HRESULT
AchRange(
   [out,retval] long* Range
   );
```

Parameter

Range

Voltage for 30000 counts on A/D

0	0.03 V range
1	0.30 V range
2	3.00 V range

Comments

Currently, this function performs no valid use on the PCI4, as this potentiostat has only one Ach Range. Ranges listed are for PC5 Family potentiostats only.

See Also

SetAchRange

SetAchRange Method

Sets the range of the Auxiliary A/D input.

Definition

```
HRESULT
SetAchRange(
   [in] VARIANT Range,
   [out,retval] long* RangeSet
   );
```

Parameter

Range

`Range` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestAchRange call prior to setting the range. The floating point number is the maximum anticipated voltage (in Volts). Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage range based on an integer identifier. Consult your potentiostat's operating manual for information on the available I/E Range settings.

RangeSet

AchRange as set

Comments

Not all ranges are available on specific potentiostats. Please consult your potentiostat's operator's manual for information specific to your potentiostat

See Also

AchRange
TestAchRange

TestAchRange Method

Finds the optimum voltage range of the voltage channel.

Definition

```
HRESULT
TestAchRange(
   [in] float Voltage,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Range
Optimum voltage range for V channel

Comments

Finds the optimum voltage range of the voltage channel to maximize the resolution of the A/D converter.

See Also

AchRange
SetAchRange

TestAchRangeAC Method

Finds the optimum voltage range of the voltage channel.

Definition

```
HRESULT
TestAchRangeAC(
   [in] float Voltage,
   [in] float Frequency,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Frequency
Measurement Frequency

Range
Optimum voltage range for V channel

Comments

Finds the optimum voltage range of the voltage channel to maximize the resolution of the A/D converter. Takes into account a higher-than-normal measurement frequency.

See Also

AchRange
SetAchRange
TestAchRange

AchRangeMode Method

Returns the status of auto-ranging mode of the V channel.

Definition

```
HRESULT
AchRangeMode(
   [out,retval] VARIANT_BOOL* RangeMode
   );
```

Parameter

RangeMode
Current state of the auto-ranging mode

VARIANT_TRUE – Auto-ranging is enabled
VARIANT_FALSE – Auto-ranging is disabled

See Also

SetAchRange
SetAchRangeMode

SetAchRangeMode Method

Enable or disable current measurement auto-ranging.

Definition

```
HRESULT
SetAchRange(
   [in] VARIANT_BOOL RangeMode,
   [out,retval] VARIANT_BOOL* RangeModeSet
   );
```

Parameter

RangeMode
Auto-range mode to set

VARIANT_TRUE – Auto-set Mode is enabled
VARIANT_FALSE – Auto-set Mode is disabled

RangeModeSet
Ach Range mode as set

Comments

Autoranging is performed by the DTAQ in use based upon the last measurement taken. Ranging only occurs if a higher or lower range is available.

See Also

SetAchRange
AchRangeMode

AchOffset Method

Report the offset voltage of the voltage channel

Definition

```
HRESULT
AchOffset(
   [out,retval] float* Offset
   );
```

Parameter

Offset
The offset voltage

Comments

The offset voltage is subtracted using the offset DAC in front of the A/D converter. This offset value is then added back to the measured value after the measurement is taken. By using offsets, the channel can be set to a more sensitive range for off-zero measurements.

See Also

SetAchOffset
AchOffsetEnable
SetAchOffsetEnable

SetAchOffset Method

Set the DC offset voltage in the voltage channel

Definition

```
HRESULT
SetAchOffset(
   [in] float Offset,
   [out,retval] float* OffsetSet
   );
```

Parameter

Offset
Offset voltage level

OffsetSet
The offset voltage as set

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the AchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn't support the DC offset, 0 will also be returned.

See Also

AchOffset
AchOffsetEnable
SetAchOffsetEnable

AchOffsetEnable Method

Reports enabled/disabled state of the voltage channel

Definition

```
HRESULT
AchOffsetEnable(
   [out,retval] VARIANT_BOOL* OffsetEnable
   );
```

Parameter

OffsetEnable
Offset state to set

VARIANT_TRUE – Offset is enabled
VARIANT_FALSE – Offset is disabled

Comments

The actual offset circuitry may be implemented in the potentiostat or in the FRA. Not all hardware configurations have this feature. The actual DC offset voltage is returned. If the AchOffsetEnable flag is set to FALSE, 0 will be returned. If the hardware doesn't support the DC offset, 0 will also be returned.

See Also

AchOffset
AchOffsetEnable
SetAchOffsetEnable

SetAchOffsetEnable Method

Enable or Disable post potentiostat offset correction.

Definition

```
HRESULT
SetAchOffsetEnable(
   [in] VARIANT_BOOL OffsetEnable,
   [out,retval] VARIANT_BOOL* OffsetEnableSet
   );
```

Parameter

OffsetEnable
Enable state to set

VARIANT_TRUE – Enable
VARIANT_FALSE – Disable

OffsetEnableSet
Ach Offset enable state as set

Comments

Enable or Disable post potentiostat offset correction. This allows the DC component of a measured voltage signal to be removed. Removal of this DC component allows multiplication of the signal by a post gain factor (1x, 10x, 100x) while keeping this post gain signal within the +/- 3V limit of the analog to digital converter (ADC).

See Also

AchOffset
SetAchOffset
AchOffsetEnable

AchFilter Method

Report current setting of Voltage Filter

Definition

```
HRESULT
AchFilter(
   [out,retval] long* Filter
   );
```

Parameter

Filter

Filter as set

4	No Filter
5	200 kHz
6	1 kHz
7	5 Hz

Comments

This filter has no effect on potentiostat stability.

See Also

SetAchFilter
TestAchFilter

SetAchFilter Method

Sets the filter on the voltage measuring channel

Definition

```
HRESULT
SetAchFilter(
   [in] VARIANT Filter,
   [out,retval] long* FilterSet
   );
```

Parameter

Filter

`Filter` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestAchFilter call prior to setting the range. The floating point number is the desired frequency. Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage filter based on an integer identifier. Consult your potentiostat's operating manual for information on the available V channel filter settings.

Filter

Filter as set

Comments

The No Filter setting should not be used under normal circumstances. It is to be used exclusively for AC impedance measurements beyond 100 kHz..

See Also

AchFilter
TestAchFilter

TestAchFilter Method

Returns the appropriate Ach Filter given a desired frequency

Definition

```
HRESULT
TestAchFilter(
   [in] float Frequency,
   [out,retval] long* Filter
   );
```

Parameter

Frequency
Frequency of interest in Hz

Filter
Best filter setting to use for the specified frequency. See AchFilter.

Comments

This function is automatically called by SetAchFilter is called with a frequency passed in as the first argument.

See Also

AchFilter
SetAchFilter

FindAchRange Method

Attempts to find optimum voltage range for use during a scan.

Definition

```
HRESULT
FindAchRange(
   [out,retval] long* AchRange
   );
```

Parameter

AchRange
Range as determined

Comments

Acquires several current points, adjusting the voltage measurement range after each point is taken. Use this method prior to starting a scan, in order to hone in on the optimum voltage range to use during the scan.

See Also

AchRange
SetAchRange

MeasureA Method

Read the voltage on the potentiostats Auxiliary A/D input.

Definition

```
HRESULT
MeasureA(
   [out,retval] float* Voltage
   );
```

Parameter

Voltage
Auxiliary A/D input voltage

Comments

The full scale range on the Auxiliary A/D input is ± 3 volts. The bit resolution is 0.1 mV. You can expect about 1 bit rms noise in the reading. This noise will average out if you take multiple readings. If you need to read a higher voltage, a resistive voltage divider can be placed in front of the high impedance differential input.

ScanLimitAC Method

Report the maximum RMS signal the Pstat can deliver

Definition

```
HRESULT
ScanLimitAC(
   [out,retval] float* Limit
   );
```

Parameter

Limit
Maximum RMS signal that can be applied

Comments

The various instruments offered by Gamry Instruments may have different AC scan capabilities. Because of these possible differences, this function is made available to determine what AC scan capabilities a specific instrument may have. It is normally used when performing an impedance measurement. A ReadZ object makes use of this function when performing an impedance measurements.

See Also

IGamryReadZ

FC350VRange Method

Report the voltage range of an FC350 or FCI350.

Definition

```
HRESULT
FC350VRange(
   [out,retval] long* Range
   );
```

Parameter

Range

Voltage Range

0 Low Range, Default (10 V)

1 High Range, Default (50V)

Comments

This function is only available on an FC350/FCI350 device.

See Also

SetFC350VRange
TestFC350VRange

SetFC350VRange Method

Set the voltage range of an FC350 or FCI350.

Definition

```
HRESULT
SetFC350VRange(
   [in] VARIANT Range,
   [out,retval] long* RangeSet
   );
```

Parameter

Parameter

`Range` is a variant that can take either a floating point number or an integer. Floating point numbers (VT_R4, VT_R8) first perform a TestFC350VRange call prior to setting the range. The floating point number is the maximum anticipated voltage (in Volts). Integer numbers (VT_I4, VT_UI4, VT_I2, VT_UI2) set the voltage range based on an integer identifier.

0	Low Range, Default (10 V)
1	High Range, Default (50V)

Comments

This function is only available on an FC350/FCI350 device. There are two voltage ranges on FC350 devices. The measurement scale of these ranges can vary based upon the requirements when the system was purchased. Please consult your hardware documentation for more information.

See Also

FC350VRange
TestFC350VRange

TestFC350VRange Method

Finds the optimum voltage range of an FC350 or FCI350.

Definition

```
HRESULT
TestFC350VRange(
   [in] float Voltage,
   [out,retval] long* Range
   );
```

Parameter

Voltage
Maximum anticipated voltage

Range
Optimum voltage range for V channel
0 Low Range, Default (10 V)
1 High Range, Default (50V)

Comments

Finds the optimum voltage range to maximize the resolution of the A/D converter.

This function is only available on an FC350/FCI350 device.

See Also

FC350VRange
SetFC350VRange

FC350VRangeMultiplier Method

Return the current Voltage Range Multiplier (FC/FCI350 Only).

Definition

```
HRESULT
FC350VRangeMultiplier(
   [out,retval] float* Multiplier
   );
```

Parameter

Multiplier
Voltage Multiplier

Comments

This function is only available on an FC350/FCI350 device. The Voltage multiplier is used to scale a value measured on the voltage channel through a FC350 device. The multiplier is used to convert from a 10V scale (Default) to a user entered scale. The default scales of 10V and 50V result in multipliers of -1.0 and -5.0, respectively. The negative values take into account that there is an inversion in the voltage channel chain.

FC350IDivisor Method

Return the current FC350 I channel divisor (FC/FCI350 Only)

Definition

```
HRESULT
FC350IDivisor(
   [out,retval] gcFC350IDIVISOR* Divisor
   );
```

Parameter

Divisor
Current FC350 I channel divisor

Comments

This function is only available on an FC350/FCI350 device. This function is included for completeness, but currently, there is no practical reason for having an I divisor other than **FC350IDivisor1**.

See Also

SetFC350IDivisor

SetFC350IDivisor Method

Set the current FC350 I channel divisor (FC/FCI350 Only).

Definition

```
HRESULT
SetFC350IDIVISOR(
   [in] gcFC350IDIVISOR Divisor
   [out,retval] gcFC350IDIVISOR* DivisorSet
   );
```

Parameter

Divisor
FC350 I channel divisor to set

DivisorSet
Divisor as set

Comments

This function is only available on an FC350/FCI350 device. This function is included for completeness, but currently, there is no practical reason for having an I divisor other than **FC350IDivisor1**.

See Also

FC350IDivisor

_IGamryPstatEvents

Overview

The _IGamryPstatEvents interface provides the event call-back functions required to handle events issued by GamryPstat objects.

The following table summarizes the members of _IGamryPstatEvents. The methods are described in detail in this section.

Member	Description
OnPstatUnavailable	The Potentiostat is no longer available for use
OnPstatClosed	The Potentiostat has been closed

OnPstatUnavailable Method

This function is called when the Potentiostat is no longer available. This change is usually caused by a plug and play event such as unplugging a USB instrument.

Definition

```
void
OnPstatUnavailable();
```

Comments

This event should be monitored to insure that your application does not try to use a device that is no longer available. When a device is no longer available, your application should close any references to the device and notify the end user that the device has become unavailable.

See Also

OnPstatClosed

OnPstatClosed Method

This function is called when the Potentiostat is closed. This change event is caused by the application making a call to Pstat.Close.

Definition

```
void
OnPstatClosed();
```

Comments

This event can be monitored when an application likes to present the current status of a potentiostat. This status can represent whether or not the device is open or closed.

See Also

Pstat Open
Pstat Close
OnPstatUnavailable

IGamryDtaq

Overview

The IGamryDtaq interface is exported by the GAMRYCOM library. It enables applications to determine information about the Gamry devices connected to the system.

Member	Description
Stop	Stops the currently running Dtaq
Run	Run the currently selected Dtaq
Cook	Retrieves points from the data acquisition queue

Stop Method

Stops the currently running Dtaq

Definition

```
HRESULT
   Stop();
```

Comments

Stop is usually called by an application which wants to prematurely end a data acquisition. This event may be connected to a user interface object such as a stop button. When data acquisition is completed due to a signal having no more data points there is no need for the application to call Stop.

See Also

Run

Run Method

Runs the currently selected Dtaq

Definition

```
HRESULT
Run(
   [in, defaultvalue(-1)] VARIANT_BOOL AutoRun,
   );
```

Parameter

AutoRun
Parameter Description

Comments

Comment text

See Also

Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

The format of this array is dependent upon the specific Dtaq in use. Some of the values that may be returned are as follows:

Value	Description
Ach	Measured Aux channel voltage
Idif	Difference between Ifwd and Irev
Ifwd	Measured forward current
Im	Applied cell current
Irev	Measured reverse current
Overload	Hexadecimal number representing overload status
Vf	Measured E vs. Eref
Vfwd	Forward voltage applied
Vsig	Signal sent to Control Amp
Vstep	Voltage step size
Vrev	Reverse voltage applied
Vu	Uncompensated voltage

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

The format for the array of data returned depends upon the actual Dtaq object used for data acquisition.

See Also

Run
Stop

_IGamryDtaqEvents

Overview

The _IGamryDtaqEvents interface provides the event call-back functions required to handle events issued by GamryDtaq objects.

The following table summarizes the members of _IGamryDtaqEvents. The methods are described in detail in this section.

Member	Description
OnDataAvailable	The Dtaq has data available for cooking
OnDataDone	Data acquisition has completed

OnDataAvailable Event

This event is fired when a Dtaq has data available for cooking.

Definition

```
void
OnDataAvailable();
```

Comments

As data is acquired by a Dtaq, this event is fired to let the application know data has been placed in the data acquisition queue. The application should then make a call to Cook to retrieve this data. There may not be a one-one relationship with this event and the acquired data points. When data is being acquired quickly, one event may be fired even though multiple data points have been added to the queue.

Polling for data with the Cook method can be used in lieu of this event should there be threading issues. Some applications do not allow other events to be processed when certain user interface events are being generated. In this case, polling will provide a smoother reception method for the data.

See Also

Cook
OnDataDone

OnDataDone Event

This event is fired when data acquisition has completed

Definition

```
void
OnDataDone();
```

Comments

When a data acquisition cycle has been completed, this event is fired. This event signals the application that no more data will be placed in the data acquisition queue. This event does not signify that the queue is empty, however, and a final call to Cook should be made to process any points still in the queue.

See Also

Cook
OnDataAvailable

IGamryDtaqIv

Overview

The IGamryDtaqIv interface is exported by the GAMRYCOM library. It allows applications to acquire Voltage and Current data for a variety of simple experiments including controlled current and controlled potential modes.

Member	Description
Init	Initialize the DtaqIv object
Cook	Retrieves points from the data acquisition queue

Init Method

Initialize a DtaqIv object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Iv Dtaq will run.

Comments

This method is called to initialize a DtaqIv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4

See Also

Run
Stop

IGamryDtaqUniv

Overview

The IGamryDtaqUniv interface is exported by the GAMRYCOM library. It allows applications to acquire data for a variety of simple experiments including controlled current and controlled potential modes.

Member	Description
Init	Initialize the DtaqUniv object
Cook	Retrieves points from the data acquisition queue

Init Method

Initialize a DtaqUniv object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Univ Dtaq will run.

Comments

This method is called to initialize a DtaqUniv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints

Number of Points to cook, returned as number of points actually cooked.

Data

SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4

See Also

Run
Stop

IGamryDtaqEfm

Overview

The IGamryDtaqEfm interface is exported by the GAMRYCOM library. It allows applications to acquire Electrochemical frequency modulation data for a specified base frequency, harmonics, and amplitude.

Member	Description
Init	Initialize a DtaqEfm object
Cook	Retrieves points from the data acquisition queue

Init Method

Initialize the DtaqEfm object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float BaseFrequency,
   [in] long Harmonic1,
   [in] long Harmonic2,
   [in] float Amplitude,
   [in] long SampPerCycle,
   [in] long NumCycles
   );
```

Parameter

Pstat
Pstat object on which the EIS Dtaq will run.

BaseFrequency
The Base Frequency determines the repeat time of the EFM waveform. It helps to define the EFM waveform applied to the cell. The two frequencies simultaneously applied to the cell will be

Frequency A = Harmonic1 * Base Frequency

and

Frequency B = Harmonic2 * Base Frequency

Harmonic1 and Harmonic2
Harmonic2 must be at least two times Harmonic1, and the two harmonics may not share any common factors. Suitable harmonic pairs are 2 and 5 or 2 and 7.

Amplitude
This is the Amplitude of each of the two sine waves, in mV. The overall EFM waveform amplitude will be between 1. and 2. times this number.

The Amplitude should be selected so that it is small compared to the Tafel constants, but large enough to give a reasonable signal. For many systems 10 to 40 mV is a good starting place.

`SampPerCycle`

This is how many samples are to be taken during each cycle. A normal number would be 256 samples.

`NumCycles`

This is the number of cycles of the given waveform which will be applied. There is an inherent cycle delay of 1, so the total number of cycles will be 1 plus this number.

Comments

This method is called to initialize a DtaqEfm object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Stop
Cook

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	V	VT_R4
2	I	VT_R4
3	Overload	VT_I4

See Also

Run
Stop

IGamryDtaqEis

Overview

The IGamryDtaqEis interface is exported by the GAMRYCOM library. It allows applications to acquire impedance data for a specified frequency and amplitude.

Member	Description
Init	Initialize a DtaqEis object
Cook	Retrieves points from the data acquisition queue
SetCycleMin	Set the minimum number of cycles to run
SetCycleMax	Set the maximum number of cycles to run
SetFilterOverride	Set a flag which allows overriding the filter selection
Vreal	Return the real component of the measured Voltage
Vimag	Return the imaginary component of the measured Voltage
Vsig	Standard deviation of the measured Voltage
Vdc	Return the DC component of the measured Voltage
Ireal	Return the real component of the measured Current
Iimag	Return the imaginary component of the measured Current
Isig	Standard deviation of the measured Current
Idc	Return the DC component of the measured Current
Zreal	Return the real component of the measured Impedance
Zimag	Return the imaginary component of the measured Impedance
Zsig	Standard deviation of the measured Impedance
Zfreq	Return the DC component of the measured Impedance
CycleLim	Set the minimum and maximum cycles to run
Imod	Return the modulus of the measured Current
Iphz	Return the phase of the measured Current
Vmod	Return the modulus of the measured Voltage
Vphz	Return the phase of the measured Voltage
Zmod	Return the modulus of the measured Impedance
Zphz	Return the phase of the measured Impedance
Result	Return the status of the measurement
OverIac	Return the overload status of the AC Current
OverIdc	Return the overload status of the DC Current
OverVac	Return the overload status of the AC Voltage
OverVdc	Return the overload status of the DC Voltage

Init Method

Initialize the DtaqEis object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Frequency,
   [in] float Amplitude,
   [in] float Precision
   );
```

Parameter

Pstat
Pstat object on which the EIS Dtaq will run.

Frequency
Test Frequency in Hertz. The driver may use a frequency slightly different from the requested frequency due to rounding errors or in order to minimize noise.

Amplitude
The RMS signal to be applied in either Volts or Amps

Precision
Requested measurement precision

Comments

This method is called to initialize a DtaqEis object. All Dtaq objects must be initialized prior to further use. The precision is used in a Correlation Coefficient (residual power) based test to determine whether or not to measure another cycle. If the precision is not met, and the maximum number of cycles is not exceeded, another cycle will be run. The precision is commonly set to 0.001.

See Also

Run
Stop
Cook

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	I	VT_R4
1	V	VT_R4

See Also

Run
Stop

SetCycleMin Method

Set the minimum number of cycles the EIS Dtaq can use to make its measurements

Definition

```
HRESULT
SetCycleMin(
   [in] long CycleMin,
   );
```

Parameter

`CycleMin`

Minimum number or cycles to be used in the measurement.

Comments

The first cycle which contains a startup transient distortion will not be counted.

See Also

SetCycleMin
CycleLim

SetCycleMax Method

Set the maximum number of cycles the EIS Dtaq can use to make its measurements

Definition

```
HRESULT
SetCycleMax(
   [in] long CycleMax,
   );
```

Parameter

CycleMax
Maximum number or cycles to be used in the measurement.

Comments

The maximum number of cycles is usually set based on the frequency being measured. Higher frequencies which take less time to measure, usually have a larger number of cycles than lower frequencies.

See Also

SetCycleMax
CycleLim

SetFilterOverride Method

Specify whether or not to override the automatic filter selection

Definition

```
HRESULT
SetFilterOverride(
   [in] VARIANT_BOOL FilterOverride,
   );
```

Parameter

FilterOverride
Parameter Description

VARIANT_TRUE – Filter Override is enabled
VARIANT_FALSE – Filter Override is disabled

Comments

This function is rarely used. It allows an application to override the automatic filter selection for the voltage and current channels. By default the filters are set based upon the frequency being measured.

See Also

SetVchFilter
SetIchFilter

Vreal Method

Returns the real portion of the AC voltage

Definition

```
HRESULT
Vreal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC Voltage in Volts

See Also

Vimag
Vsig
Vdc
Ireal
Zreal

Vimag Method

Returns the imaginary portion of the AC voltage

Definition

```
HRESULT
Vimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC Voltage in Volts

See Also

Vreal
Vsig
Vdc
Iimag
Zimag

Vsig Method

Return the standard deviation of the measured voltage.

Definition

```
HRESULT
Vsig(
   [out,retval] float* Value
   );
```

Parameter

`Value`
Standard deviation in volts.

See Also

Vreal
Vimag
Vdc
Ireal
Zreal

Vdc Method

Return the measured DC voltage

Definition

```
HRESULT
Vdc(
   [out,retval] float* Value
   );
```

Parameter

Value
DC Voltage in Volts

See Also

Vreal
Vimag
Vsig
Ireal
Zreal

Ireal Method

Returns the real portion of the AC current

Definition

```
HRESULT
Ireal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC current in Amps

See Also

Iimag
Isig
Idc
Vreal
Zreal

Iimag Method

Returns the imaginary portion of the AC current

Definition

```
HRESULT
Iimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC current in Amps

See Also

Ireal
Isig
Idc
Vimag
Zimag

Isig Method

Return the standard deviation in the measured current

Definition

```
HRESULT
Isig(
   [out,retval] float* Value
   );
```

Parameter

Value
Value in amps.

See Also

Ireal
Iimag
Idc
Vreal
Zreal

Idc Method

Return the measured DC current

Definition

```
HRESULT
Idc(
   [out,retval] float* Value
   );
```

Parameter

Value
DC current in Amps

See Also

Ireal
Iimag
Isig
Vreal
Zreal

Zreal Method

Returns the real portion of the AC impedance

Definition

```
HRESULT
Zreal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC impedance in Ohms

See Also

Zimag
Zsig
Zfreq
Vreal
Ireal

Zimag Method

Returns the imaginary portion of the AC impedance

Definition

```
HRESULT
Zimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC impedance in Ohms

See Also

Zreal
Zsig
Zfreq
Vimag
Iimag

Zsig Method

Return the standard deviation of the measured impedance.

Definition

```
HRESULT
Zsig(
   [out,retval] float* Value
   );
```

Parameter

Value
Value in ohms.

Comments

This value is set to 1.0 by default.

See Also

Zreal
Zimag
Zfreq
Vreal
Ireal

Zfreq Method

Return the measured frequency as measured

Definition

```
HRESULT
Zfreq(
   [out,retval] float* Value
   );
```

Parameter

Value
Frequency in Hertz

See Also

Zreal
Zimag
Zsig
Vreal
Ireal

CycleLim Method

Set the maximum and minimum number of cycles the EIS Dtaq can use to make its measurements

Definition

```
HRESULT
CycleLim(
   [in] long CycleMin,
   [in] long CycleMax
   );
```

Parameter

CycleMin
Minimum number of cycles to be used in the measurement

CycleMax
Maximum number of cycles to be used in the measurement

Comments

The first cycle which contains a startup transient distortion will not be counted. This function is available simply to allow setting the minimum and maximum number of cycles in a single call.

See Also

SetCycleMin
SetCycleMax

Imod Method

Return the modulus of the AC current

Definition

```
HRESULT
Imod(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS Magnitude of AC current in Amps

See Also

Vmod
Zmod

Iphz Method

Return the phase of the AC current

Definition

```
HRESULT
Iphz(
  [out,retval] float* Value
  );
```

Parameter

Value
Phase of AC current in Degrees

See Also

Vphz
Zphz

Vmod Method

Return the modulus (magnitude) of the AC voltage

Definition

```
HRESULT
Vmod(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS Magnitude of AC voltage in Volts

See Also

Imod
Zmod

Vphz Method

Return the phase of the AC voltage

Definition

```
HRESULT
Vphz(
   [out,retval] float* Value
   );
```

Parameter

Value
Phase of AC voltage in Degrees

See Also

Iphz
Zphz

Zmod Method

Return the modulus (magnitude) of the AC impedance

Definition

```
HRESULT
Zmod(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS Magnitude of AC impedance in Ohms

See Also

Vmod
Imod

Zphz Method

Return the phase of the AC impedance

Definition

```
HRESULT
Zphz(
   [out,retval] float* Value
   );
```

Parameter

Value
Phase of AC impedance in Degrees

See Also

Vphz
Iphz

Result Method

Return the measurement status of the DtaqEis object

Definition

```
HRESULT
Result(
   [out,retval] short* Result
   );
```

Parameter

Result

DtaqEISStatusInvalid=0
DtaqEISStatusDelay=1
DtaqEISStatusMeasuring=2
DtaqEISStatusMeasOk=3
DtaqEISStatusCommErr=4
DtaqEISStatusTimeout=5
DtaqEISStatusCycleLim=6
DtaqEISStatusControl=7
DtaqEISStatusOverrun=8
DtaqEISStatusOverrange=9
DtaqEISStatusOverrunQ=10

Comments

The result of the DtaqEis object should be checked once the DataDone event has been fired. The result will then let the application determine whether or not the result measurement was successful. A successful result (*DtaqEISStatusMeasOk*) does not symbolize that the measured value is good, just that a measurement was able to be taken. A result of *DtaqEISStatusCycleLim* is returned when the maximum number of cycles have been run and the precision has not yet been met. Other results usually indicate some error in the measurement.

See Also

_IGamryDtaqEvents::OnDataDone

OverIac Method

Return the current overload status

Definition

```
HRESULT
OverIac(
   [out,retval] VARIANT_BOOL* Result
   );
```

Parameter

Result

AC overload status of the current signal

VARIANT_TRUE – Overload
VARIANT_FALSE – No Overload

Comments

If the signal conditioning and/or analysis circuitry is overloaded both for maximum peak and minimum peak current, this function will return VARIANT_TRUE, indicating the need for a correction of the current range.

See Also

OverIdc
OverVac
OverVdc

OverIdc Method

Return the current overload status

Definition

```
HRESULT
OverIdc(
   [out,retval] VARIANT_BOOL* Result
   );
```

Parameter

Result
DC overload status of the current signal

VARIANT_TRUE – Overload
VARIANT_FALSE – No Overload

Comments

If the signal conditioning and/or analysis circuitry is overloaded on either maximum or minimum current but not both, this function will return VARIANT_TRUE.

See Also

OverIac
OverVac
OverVdc

OverVac Method

Return the voltage overload status

Definition

```
HRESULT
OverVac(
   [out,retval] VARIANT_BOOL* Result
   );
```

Parameter

Result
AC overload status of the voltage signal

VARIANT_TRUE – Overload
VARIANT_FALSE – No Overload

Comments

If the signal conditioning and/or analysis circuitry is overloaded both for maximum peak and minimum peak voltage, this function will return VARIANT_TRUE, indicating the need for a correction of the voltage range.

See Also

OverVdc
OverIac
OverIdc

OverVdc Method

Return the voltage overload status

Definition

```
HRESULT
OverVdc(
   [out,retval] VARIANT_BOOL* Result
   );
```

Parameter

Result

DC overload status of the voltage signal

VARIANT_TRUE – Overload
VARIANT_FALSE – No Overload

Comments

If the signal conditioning and/or analysis circuitry is overloaded on either maximum or minimum voltage but not both, this function will return VARIANT_TRUE.

See Also

OverVac
OverIac
OverIdc

IGamryDtaqCpiv

Overview

The IGamryDtaqCpiv interface is exported by the GAMRYCOM library. It allows applications to acquire data for standard controlled potential experiments.

Member	Description
Init	Initialize a DtaqCpiv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopDIMin	Set DIMin stop test value
SetStopDIMax	Set DIMax stop test value
SetStopADIMin	Set absolute value DIMin stop test value
SetStopADIMax	Set absolute value DIMax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test
SetStopAtDelayDIMin	Set number of delay points for DImin stop test
SetStopAtDelayDIMax	Set number of delay points for DImax stop test
SetStopAtDelayADIMin	Set number of delay points for ADImin stop test
SetStopAtDelayADIMax	Set number of delay points for ADImax stop test

Init Method

Initialize the DtaqCpiv object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Cpiv Dtaq will run.

Comments

This method is called to initialize a DtaqCpiv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4
8	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopDIMin Method

Change in current limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopDIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if dI/dt < Value

Comments

Signed version of SetStopADIMin

See Also

SetStopADIMin
SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopDIMax Method

Change in current limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopDIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if dI/dt > Value

Comments

Signed version of SetStopADIMax

See Also

SetStopADIMax
SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopADIMin Method

Change in current limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if abs(dI/dt) < Value

Comments

The SetStopADIMin criterion is used run a sample until it shows stable behavior.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopADIMax Method

Change in current limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if abs(dI/dt) > Value

Comments

The SetStopADIMax criterion limits the rate of change in current.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayDIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayDIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayDIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayDIMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayADIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayADIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayADIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayADIMax(
   [in] long Value
   );
```

Parameter

`Value`
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqCiiv

Overview

The IGamryDtaqCiiv interface is exported by the GAMRYCOM library. It allows applications to acquire data for standard controlled current experiments.

Member	Description
Init	Initialize a DtaqCiiv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopVMin	Set Vmin stop test value
SetStopVMax	Set Vmax stop test value
SetStopDVMin	Set DVMin stop test value
SetStopDVMax	Set DVMax stop test value
SetStopADVMin	Set absolute value DVMin stop test value
SetStopADVMax	Set absolute value DVMax stop test value
SetStopAtDelayVMin	Set number of delay points for Vmin stop test
SetStopAtDelayVMax	Set number of delay points for Vmax stop test
SetStopAtDelayDVMin	Set number of delay points for DVmin stop test
SetStopAtDelayDVMax	Set number of delay points for DVmax stop test
SetStopAtDelayADVMin	Set number of delay points for ADVmin stop test
SetStopAtDelayADVMax	Set number of delay points for ADVmax stop test

Init Method

Initialize the DtaqCiiv object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Ciiv Dtaq will run.

Comments

This method is called to initialize a DtaqCiiv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4
8	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopVMin Method

Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E < Value

Comments

The SetStopVMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMax Method

Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E > Value

Comments

The SetStopVMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopDVMin Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopDVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if dE/dt < Value

Comments

Signed version of SetStopADVMin

See Also

SetStopADIMin
SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopDVMax Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopDVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if dE/dt > Value

Comments

Signed version of SetStopADVMax

See Also

SetStopADIMax
SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopADVMin Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if abs(dE/dt) < Value

Comments

The SetStopADVMin criterion is used run a sample until it shows stable behavior.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopADVMax Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if abs(dE/dt) > Value

Comments

The SetStopADIMax criterion limits the rate of change in voltage.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayDVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayDVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayDVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayDVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayADVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayADVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayADVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayADVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqChrono

Overview

The IGamryDtaqChrono interface is exported by the GAMRYCOM library. It allows applications to acquire data for Chronoamperometry, Chronopotentiometry, and Chronocoulometry experiments.

Member	Description
Init	Initialize a DtaqCiiv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopXMin	Set Xmin stop test value (X varies by chrono type)
SetStopXMax	Set Xmax stop test value (X varies by chrono type)
SetStopAtDelayXMin	Set number of delay points for Xmin stop test
SetStopAtDelayXMax	Set number of delay points for Xmax stop test
SetDecimation	Enable decimation for longer run-time experiments

Init Method

Initialize a DtaqChrono object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] gcDTAQCHRONO_TYPE Type
   );
```

Parameter

Pstat
Pstat object on which the Chrono Dtaq will run.

Type
Mode for Chrono Dtaq

`ChronoAmp`	ChronoAmperometry Mode
`ChronoCoul`	ChornoCoulometry Mode
`ChronoPot`	ChronoPotentiometry Mode

Comments

The Type is used to control the mode in which the Dtaq will process the raw data. When the ChornoAmperometry or ChornoCoulometry Dtaqs are run, the potentiostat must be in Potentiostatic mode. When the ChronoPotentiometry Dtaq is run, the potentiostat must be in Galvanostatic mode.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Q	VT_R4
5	Vsig	VT_R4
6	Ach	VT_R4
7	IERange	VT_I4
8	Overload	VT_I4
9	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopXMin Method

Current, Charge, or Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopXMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Depending on mode, terminate if I, Q, or V < Value

Comments

The SetStopXMin criterion is used to limit a negative current, charge, or voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopXMax Method

Current, Charge, or Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopXMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Depending on mode, terminate if I, Q, or V > Value

Comments

The SetStopXMax criterion is used to limit a positive current, charge, or voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayXMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayXMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I, Q, or V < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayXMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayXMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I, Q, or V > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetDecimation Method

Used to allow for a log-time display during the Chrono experiment.

Definition

```
HRESULT
SetDecimation(
   [in] VARIANT_BOOL Enable,
   [in,defaultvalue(50)] long PreStepPts,
   [in,defaultvalue(50)] long UndecStepPts
   );
```

Parameter

Enable
VARIANT_TRUE
VAIRANT_FALSE

PreStepPts
The number of data points to show in an undecimated format prior to the step.

UndecStepPts
The number of data points to show in an undecimated format after the step

Comments

This method, when enabled, allows for longer experiments with fewer points, but still acquires data quickly around each step.

IGamryDtaqGalv

Overview

The IGamryDtaqGalv interface is exported by the GAMRYCOM library. It allows applications to acquire data for galvanic corrosion experiments.

Member	Description
Init	Initialize a DtaqGalv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopVMin	Set Vmin stop test value
SetStopVMax	Set Vmax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test
SetStopAtDelayVMin	Set number of delay points for Vmin stop test
SetStopAtDelayVMax	Set number of delay points for Vmax stop test

Init Method

Initialize the DtaqGalv object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Galv Dtaq will run.

Comments

This method is called to initialize a DtaqGalv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

`NumPoints`
Number of Points to cook, returned as number of points actually cooked.

`Data`
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4
8	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMin Method

Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E < Value

Comments

The SetStopVMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMax Method

Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E > Value

Comments

The SetStopVMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value

Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

`Value`
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dI/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt < Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset. For slope tests, such as dE/dt > Limit, the slope will be evaluated over N points.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqIvt

Overview

The IGamryDtaqIvt interface is exported by the GAMRYCOM library. It allows applications to acquire generic current and voltage data.

Member	Description
Init	Initialize the DtaqIvt object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshAMin	Set Amin threshold criteria for stop tests
SetThreshAMax	Set Amax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopVMin	Set Vmin stop test value
SetStopVMax	Set Vmax stop test value
SetStopAMin	Set Amin stop test value
SetStopAMax	Set Amax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test
SetStopAtDelayVMin	Set number of delay points for Vmin stop test
SetStopAtDelayVMax	Set number of delay points for Vmax stop test
SetStopAtDelayAMin	Set number of delay points for Amin stop test
SetStopAtDelayAMax	Set number of delay points for Amax stop test

Init Method

Initialize the DtaqIvt object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Ivt Dtaq will run.

Comments

This method is called to initialize a DtaqIvt object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4
8	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshAMin Method

Auxiliary voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshAMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if Aux V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshAMax Method

Auxiliary voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshAMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if Aux V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMin Method

Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E < Value

Comments

The SetStopVMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMax Method

Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E > Value

Comments

The SetStopVMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAMin Method

Auxiliary voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopAMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if Aux V < Value

Comments

The SetStopAMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAMax Method

Auxiliary voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopAMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if Aux V > Value

Comments

The SetStopAMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMax(
   [in] long Value
   );
```

Parameter

Value

Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayAMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayAMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as Aux V < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayAMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayAMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as Aux V > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqOcv

Overview

The IGamryDtaqOcv interface is exported by the GAMRYCOM library. It allows applications to acquire data for open circuit experiments.

Member	Description
Init	Initialize the DtaqOcv object
Cook	Retrieves points from the data acquisition queue
SetStopADVMin	Set ADVmin stop test value
SetStopADVMax	Set ADVmax stop test value

Init Method

Initialize the DtaqOcv object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Ocv Dtaq will run.

Comments

This method is called to initialize a DtaqOcv object. All Dtaq objects must be initialized prior to further use. This Dtaq is commonly used to perform open circuit measurements. The cell state for the potentiostat should be left in the off position.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vm	VT_R4
3	Vsig	VT_R4
4	Ach	VT_R4
5	Overload	VT_I4
6	StopTest	VT_I4

See Also

Run
Stop

SetStopADVMin Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if abs(dE/dt) < Value

Comments

The SetStopADVMin criterion is used run a sample until it shows stable behavior.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopADVMax Method

Change in voltage limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopADVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

`Enable`
Enable or disable this Stop criteria

`Value`
Terminate if abs(dE/dt) > Value

Comments

The SetStopADIMax criterion limits the rate of change in voltage.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqPv

Overview

The IGamryDtaqPv interface is exported by the GAMRYCOM library. It allows applications to acquire data for pulse voltammetry experiments.

Member	Description
Init	Initialize the DtaqPv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopVMin	Set Vmin stop test value
SetStopVMax	Set Vmax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test
SetStopAtDelayVMin	Set number of delay points for Vmin stop test
SetStopAtDelayVMax	Set number of delay points for Vmax stop test
SetReverseDifference	Change the difference current calculation.

Init Method

Initialize the DtaqPv object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Pv Dtaq will run.

Comments

This method is called to initialize a DtaqPv object. All Dtaq objects must be initialized prior to further use. This Dtaq is commonly used to perform pulse voltammetry experiments.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vstep	VT_R4
2	Vfwd	VT_R4
3	Vrev	VT_R4
4	Ifwd	VT_R4
5	Irev	VT_R4
6	Idif	VT_R4
7	Vsig	VT_R4
8	Ach	VT_R4
9	IERange	VT_I4
10	Overload	VT_I4
11	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMin Method

Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E < Value

Comments

The SetStopVMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMax Method

Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E > Value

Comments

The SetStopVMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetReverseDifference Method

Changes the method in which the difference current is calculated.

Definition

```
HRESULT
SetReverseDifference(
   [in] VARIANT_BOOL Reverse,
   );
```

Parameter

Reverse
VARIANT_TRUE means that Idifference is calculated using Irev-Ifwd.
VARIANT_FALSE means that Idifference is calculated using Ifwd-Irev.

Comments

Different pulse voltammetry techniques or research groups may require a change in the way the different current is calculated.

IGamryDtaqRcv

Overview

The IGamryDtaqRcv interface is exported by the GAMRYCOM library. It allows application to acquire data for cyclic voltammetry experiments

Member	Description
Init	Initialize the DtaqRcv object
Cook	Retrieves points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test

Init Method

Initialize the DtaqRcv object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Rcv Dtaq will run.

Comments

This method is called to initialize a DtaqPv object. All Dtaq objects must be initialized prior to further use.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Vf	VT_R4
2	Vu	VT_R4
3	Im	VT_R4
4	Vsig	VT_R4
5	Ach	VT_R4
6	IERange	VT_I4
7	Overload	VT_I4
8	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

Value

Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

IGamryDtaqSqwv

Overview

The IGamryDtaqSqwv interface is exported by the GAMRYCOM library. It is designed for applications to acquire data for square wave voltammetry.

Member	Description
Init	Initialize the DtaqSqwv object
Cook	Retrieve points from the data acquisition queue
SetThreshIMin	Set Imin threshold criteria for stop tests
SetThreshIMax	Set Imax threshold criteria for stop tests
SetThreshVMin	Set Vmin threshold criteria for stop tests
SetThreshVMax	Set Vmax threshold criteria for stop tests
SetThreshTMin	Set Tmin threshold criteria for stop tests
SetThreshTMax	Set Tmax threshold criteria for stop tests
SetStopIMin	Set Imin stop test value
SetStopIMax	Set Imax stop test value
SetStopVMin	Set Vmin stop test value
SetStopVMax	Set Vmax stop test value
SetStopAtDelayIMin	Set number of delay points for Imin stop test
SetStopAtDelayIMax	Set number of delay points for Imax stop test
SetStopAtDelayVMin	Set number of delay points for Vmin stop test
SetStopAtDelayVMax	Set number of delay points for Vmax stop test
SetReverseDifference	Change the difference current calculation.

Init Method

Initialize the DtaqSqwv object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which the Sqwv Dtaq will run.

Comments

This method is called to initialize a DtaqSqwv object. All Dtaq objects must be initialized prior to further use. This Dtaq is commonly used to run square wave experiments.

See Also

Run
Cook
Stop

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	Time	VT_R4
1	Ifwd	VT_R4
2	Irev	VT_R4
3	Idif	VT_R4
4	Vfwd	VT_R4
5	Vrev	VT_R4
6	Vstep	VT_R4
7	Vsig	VT_R4
8	Ach	VT_R4
9	IERange	VT_I4
10	Overload	VT_I4
11	StopTest	VT_I4

See Also

Run
Stop

SetThreshIMin Method

Current minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshIMax Method

Current maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if I > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMin Method

Voltage minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshVMax Method

Voltage maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if V > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMin Method

Time minimum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T < Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetThreshTMax Method

Time maximum threshold that, once met, enables stop criteria.

Definition

```
HRESULT
SetThreshTMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this threshold test

Value
Enable StopAt if T > Value

Comments

Once any of the threshold tests is met, the Stop criteria are evaluated for the rest of the scan. Note that after the threshold test is passed, all Stop tests are enabled. Once Stop testing is enabled, it is never disabled, even if the Threshold test that is enabled is no longer valid.

See Also

SetThresh functions
SetStop functions
SetStopAtDelay functions

SetStopIMin Method

Current minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I < Value

Comments

The SetStopIMin criterion is used to limit a negative current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopIMax Method

Current maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopIMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if I > Value

Comments

The SetStopIMax criterion is used to limit a positive current swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMin Method

Voltage minimum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMin(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E < Value

Comments

The SetStopVMin criterion is used to limit a negative voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopVMax Method

Voltage maximum limit that, once exceeded, terminates data acquisition.

Definition

```
HRESULT
SetStopVMax(
   [in] VARIANT_BOOL Enable,
   [in, defaultvalue(0.0)] float Value
   );
```

Parameter

Enable
Enable or disable this Stop criteria

Value
Terminate if E > Value

Comments

The SetStopVMax criterion is used to limit a positive voltage swing.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayIMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayIMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as I > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMin Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMin(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E < Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetStopAtDelayVMax Method

Require StopAt criterion to be met for a number of data points prior to termination.

Definition

```
HRESULT
SetStopAtDelayVMax(
   [in] long Value
   );
```

Parameter

Value
Delay for this number of points

Comments

Used to avoid premature termination of data acquisition because of noise, creating a false reading for a StopAt test. For absolute tests, such as E > Limit, the criterion must be met for N points consecutively before the criterion will be accepted. If one point does not fall within the criterion, the point counter will be reset.

See Also

SetStop functions
SetThresh functions
SetStopAtDelay functions

SetReverseDifference Method

Changes the method in which the difference current is calculated.

Definition

```
HRESULT
SetReverseDifference(
   [in] VARIANT_BOOL Reverse,
   );
```

Parameter

Reverse
VARIANT_TRUE means that Idifference is calculated using Irev-Ifwd.
VARIANT_FALSE means that Idifference is calculated using Ifwd-Irev.

Comments

Different pulse voltammetry techniques or research groups may require a change in the way the different current is calculated.

IGamrySignalArray

Overview

The IGamrySignalArray interface is exported by the GAMRYCOM library. This signal is capable of being used for a wide variety of experiments. It can accept an array of arbitrary values which are then used by the signal generator of the potentiostat.

Member	Description
Init	Creates the signal
Tweak	Adjusts the signal

Init Method

Initializes a Universal Signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] long Cycles,
   [in] float SampleRate,
   [in] long SamplesPerCycle,
   [in] SAFEARRAY(float)* SignalArray,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Cycles
Number of cycles to run

SampleRate
Time between data acquisition samples in seconds.

SamplesPerCycle
Number of points in the Signal Array

SignalArray
Array of size SamplesPerCycle containing the points that make up the array.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode.

Comments

The array signal is a simple signal that can be used to make virtually any type of waveform for application through a potentiostat. An array of voltages or currents is passed into **SignalArray** and each value is then applied to the potentiostat at the same rate as data acquisition. The maximum size of the **SignalArray** is 262143 points. The maximum number of cycles is (2^16 – 1). **Cycles** is entered as a positive number. A (-1) symbolizes a "continuous" signal, where the signal is repeated for a maximum of (2^32 – 1) cycles. The values passed in for

`SignalArray` will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

See Also

Tweak

Tweak Method

Adjusts the parameters of an Array Signal object

Definition

```
HRESULT
Init(
   [in] long Cycles,
   [in] float SampleRate,
   [in] long SamplesPerCycle,
   [in] SAFEARRAY(float)* SignalArray,
   [in] gcSIGTWEAKSTYLE Style
   );
```

Parameter

Cycles
Number of cycles to run

SampleRate
Time between data acquisition samples in seconds.

SamplesPerCycle
Number of points in the Signal Array

SignalArray
Array of size SamplesPerCycle containing the points that make up the array.

Style
The style of the tweak.

Comments

The array signal is a simple signal that can be used to make virtually any type of waveform for application through a potentiostat. An array of voltages or currents is passed into **SignalArray** and each value is then applied to the potentiostat at the same rate as data acquisition. The maximum size of the **SignalArray** is 262143 points. The maximum number of cycles is (2^16 – 1). **Cycles** is entered as a positive number. A (-1) symbolizes a “continuous” signal, where the signal is repeated for a maximum of (2^32 – 1) cycles. The values passed in for **SignalArray** will be interpreted as volts for potentiostat mode or amps for galvanostat mode. The Style of the tweak can be one of the values specified by the gcSIGTWEAKSTYLE enum. The most general type is SigTweakStyleReset which causes the signal to be applied from the beginning of the array. The other two styles are used to have the signal begin to be applied from within a location in the array. SigTweakStyleContinue continues the signal from the same point number it was applying from the previous array. SigTweakStyleScale continues the signal from a point number scaled to the new signal. This scaling is a ratio between the number of points for the new array and the number of points for the old array. For example, Lets say the old array had 100 points and the signal was being applied at point 35. The new array has 200 points. The scale style would have the ratio of 200/100 or 2. The new signal would be applied starting at point 35 * 2 or 70.

See Also

Init
gcSIGTWEAKSTYLE

IGamrySignalUniv

Overview

The IGamrySignalUniv interface is exported by the GAMRYCOM library. This signal is capable of being used for a wide variety of experiments. It can accept an array of arbitrary values which are then used by the signal generator of the potentiostat.

Member	Description
Init	Creates the signal

Init Method

Initializes a Universal Signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Vo,
   [in] long Cycles,
   [in] float SampleRate,
   [in] long SamplesPerCycle,
   [in] long Sections,
   [in] SAFEARRAY(float)* SignalArray,
   [in] SAFEARRAY(unsigned char)* SectionsArray,
   [in] SAFEARRAY(float)* StepsArray,
   [in] SAFEARRAY(long)* RLEArray,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Vo
Starting Voltage

Cycles
Number of cycles to run

SampleRate
Time between data acquisition samples in seconds.

SamplesPerCycle
Number of points in the Signal Array

Sections
Number of different sections in the Signal Array.

SignalArray
Array of size SamplesPerCycle containing the points that make up the array.

SectionsArray

Array of size SamplesPerCycle containing the section number for a corresponding entry in the Signal Array.

StepsArray

Array of size Sections, containing the step value for each section.

RLEArray

Array of size SamplesPerCycle, containing the run time length for each sample, in the lower 30 bits. The 31st bit controls whether or not the sample is to be passed through when using the Universal Dtaq. The 32nd bit is the cell control state when using the Universal Dtaq.

CtrlMode

Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode.

Comments

The universal signal is a very versatile signal that can be used to make virtually any type of waveform for application through a potentiostat. The maximum size of the **SignalArray** is 262143 points. The maximum number of **Sections** is 255. The maximum number of cycles is (2^16 – 1). **Cycles** is entered as a positive number. A (-1) symbolizes a "continuous" signal, where the signal is repeated for a maximum of (2^32 – 1) cycles. The **SectionsArray** is used to specify a section number for each entry in the **SignalArray**. Section numbers control whether or not a Step value is added to the **SignalArray** entry. Standard use of the signal is to not make use of the **SectionsArray**, and to have it be filled with zeros. The **StepsArray** is used to specify a step value to be added to the signal array as the signal enters that section. Steps are not allowed to be used with a continuous signal. The **RLEArray** serves triple duty, in specifying not only the runtime length encoding, but also the cell control state and the pass through state when using the Universal Dtaq. The runtime length is entered in number of points (x). This means that a **SignalArray** value is repeated (x) number of times. Cell control state is a bit simply means that the cell will be either turned on (1), or turned off (0) when the specific signal array point is applied. The pass through state simply means that the data will be passed back by the dtaq (1) or not (0). For a single runtime length with pass-through and the cell turned on, this value should be 0xC00000001. The values passed in for **Vo**, **SignalArray**, and **StepsArray** will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

IGamrySignalConst

Overview

The IGamrySignalConst interface is exported by the GAMRYCOM library.

The Const signal describes a constant signal waveform which can be applied by a potentiostat. The object encapsulates information about the applied value, the elapsed time, and the data acquisition rate.

Member	Description
Init	Creates a new constant voltage signal

Init Method

Initializes a constant signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Svalue,
   [in] float TotalTime,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Constant Signal will run.

Svalue
Applied signal in either volts or amps.

TotalTime
Total time in seconds.

SampleRate
Time between data acquisition samples in seconds.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

The constant signal is used when an application does not need to change the value used by the signal generator.

IGamrySignalStep

Overview

The IGamrySignalStep interface is exported by the GAMRYCOM library.

The Step signal describes a two-part step waveform which can be applied by a potentiostat. The object encapsulates information about the initial and final values, the elapsed time at each value, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes a step signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Tinit,
   [in] float Sfinal,
   [in] float Tfinal,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Sinit
Initial value in volts or amps.

Tinit
Initial time in seconds

Sfinal
Final value in volts or amps.

Tfinal
Final time in seconds

SampleRate
Time between data acquisition samples in seconds.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is commonly used for potentiostatic step experiments where a single step is required. The signal starts at **Sinit** and then steps to **Sfinal**.

IGamrySignalRamp

Overview

The IGamrySignalRamp interface is exported by the GAMRYCOM library.

The Ramp signal describes a ramp waveform which can be applied by a potentiostat. The object encapsulates information about the starting and ending values, the ramp rate, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes a Ramp Signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Sfinal,
   [in] float ScanRate,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Sinit
Initial value in volts or amps.

Sfinal
Final value in volts or amps.

ScanRate
Scan rate in volts/second or amps/second.

SampleRate
Time between data acquisition samples in seconds.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is commonly used for potentiostatic step experiments where a single step is required. The signal starts at **Sinit** and then ramps to **Sfinal**.

IGamrySignalDstep

Overview

The IGamrySignalDstep interface is exported by the GAMRYCOM library.

The Dstep signal describes a three-part step waveform which can be applied by a potentiostat. The object encapsulates information about the initial, second, and final values, the elapsed times at each value, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes a double step signal object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Tinit,
   [in] float Sstep1,
   [in] float Tstep1,
   [in] float Sstep2,
   [in] float Tstep2,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Dstep Signal will run.

Sinit
Initial value in volts or amps.

Tinit
Initial time in seconds

Sstep1
Step 1 voltage in volts or amps.

Tstep1
Step 1 time in seconds

Sstep2
Final value in volts or amps.

Tstep2
Final time in seconds

SampleRate
Time between data acquisition samples in seconds.

`CtrlMode`

Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is commonly used for chronoamp or chronopot experiments where a double step is required. The signal starts at **`Sinit`** and then steps to **`Sstep1`** and further to **`Sstep2`**.

IGamrySignalMstep

Overview

The IGamrySignalMstep interface is exported by the GAMRYCOM library.

The Mstep signal describes a multi-part step waveform which can be applied by a potentiostat. The object encapsulates information about the initial and step values, the elapsed time at each value, the number of steps using the step value as an increment, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes a multi-step signal object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Sstep,
   [in] float Tinit,
   [in] float Tstep,
   [in] long Nstep,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Mstep Signal will run.

Sinit
Initial value in volts or amps.

Sstep
Step value in volts or amps.

Tinit
Initial step time in seconds.

Tstep
Subsequent step time in seconds.

Nstep
Number of steps to run using the step information.

SampleRate
Time between data acquisition samples in seconds.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is used to replicate a constant step-height staircase waveform, with a user specified number of steps.

IGamrySignalPv

Overview

The IGamrySignalPv interface is exported by the GAMRYCOM library.

The Pv signal describes a ramp and square wave waveform that can be applied by a potentiostat. The object encapsulates information about the starting values, value step, pulse size, pulse duration, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Creates the Pv Signal

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Spv,
   [in] float Spulse,
   [in] VARIANT_BOOL EnableOverrideA,
   [in] float SOverrideA,
   [in] VARIANT_BOOL EnableOverrideB,
   [in] float SOverrideB,
   [in] long MaxCycles,
   [in] float TimerRes,
   [in] float PulseTime,
   [in] float CycleTime,
   [in] float IntPeriod,
   [in] VARIANT_BOOL DropKnockSignalEnable,
   [in] float DropKnockSignalDuration,
   [in] VARIANT_BOOL DropKnockSignalPolarity,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Sinit
Initial value in volts or amps.

Spv
Step in volts or amps.

Spulse
Pulse size in volts or amps.

EnableOverrideA
Override the A portion of the signal

SOverrideA
Override value in volts or amps.

EnableOverrideB
Override the B portion of the signal

SOverrideB
Override value in volts or amps.

MaxCycles
Number of cycles to run this acquisition.

TimerRes
Time between data samples in seconds.

PulseTime
Duration of the pulse (B)

CycleTime
Duration of a cycle (A + B)

IntPeriod
Time for averaging data

DropKnockSignalEnable
Use the digital output

DropKnockSignalDuration
Time to apply a signal

DropKnockSignalPolarity
Signal polarity

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is used to run pulse voltammetry experiments such as Normal Pulse or Differential Pulse voltammetry. It is normally combined with the PV dtaq for data acquisition. The values passed in for **Sinit** and **Spv**, **Spulse**, **SOverrideA**, and **SOverrideB** will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

IGamrySignalRupdn

Overview

The IGamrySignalRupdn interface is exported by the GAMRYCOM library.

The Rupdn signal describes a tri-value ramp waveform which can be applied by a potentiostat. The signal is ramped from an initial value to an apex 1 value, then to an apex 2 value and back again to a final value. Rupdn is normally used for Cyclic Voltammetry scans. The object encapsulates information about the initial, apex and final values, the ramp rates, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes a research up-down signal object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Sapex1,
   [in] float Sapex2,
   [in] float Sfinal,
   [in] float ScanInit,
   [in] float ScanApex,
   [in] float ScanFinal,
   [in] float HoldTime0,
   [in] float HoldTime1,
   [in] float HoldTime2,
   [in] float SampleRate,
   [in] long Cycles,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Sinit
Initial value in volts or amps.

Sapex1
Apex 1 value in volts or amps.

Sapex2
Apex 2 value in volts or amps.

Sfinal
Final value in volts or amps.

ScanInit
Initial scan rate in volts/second or amps/second.

ScanApex
Apex scan rate in volts/second or amps/second.

`ScanFinal`
Final scan rate in volts/second or amps/second.

`HoldTime0`
Time to hold at Apex 1 in seconds

`HoldTime1`
Time to hold at Apex 2 in seconds

`HoldTime2`
Time to hold at **`Sfinal`** in seconds

`SampleRate`
Time between data acquisition steps.

`Cycles`
The number of cycles the signal is to be run

`CtrlMode`
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is commonly used to perform cyclic voltammetry experiments. It is usually combined with the RCV dtaq for acquisition.

IGamrySignalUpdn

Overview

The IGamrySignalUpdn interface is exported by the GAMRYCOM library.

The Updn signal describes a dual-value ramp waveform which can be applied by a potentiostat. The signal is ramped from an initial value to an apex value and back again to a final value. Updn is normally used for cyclic scans such as pitting tests. The object encapsulates information about the initial, apex and final values, the ramp rates, and the data acquisition rate.

Member	Description
Init	Creates a new signal

Init Method

Initializes an up-down signal object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Sapex,
   [in] float Sfinal,
   [in] float ScanRateFwd,
   [in] float ScanRateRev,
   [in] float SampleRate,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the Univ Signal will run.

Sinit
Initial value in volts or amps.

Sapex
Apex value in volts or amps.

Sfinal
Final value in volts or amps.

ScanRateFwd
Forward scan rate in volts/second or amps/second.

ScanRateRev
Reverse scan rate in volts/second or amps/second.

SampleRate
Time between data acquisition samples in seconds.

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

The up-down signal is useful for experiments like cyclic polarization.

IGamrySignalSqwv

Overview

The IGamrySignalSqwv interface is exported by the GAMRYCOM library.

The Sqwv signal describes a square wave waveform that can be applied by a potentiostat. The object encapsulates information about the starting and final values, value step, pulse size, and pulse duration.

Member	Description
Init	Creates a new signal

Init Method

Initialize a square wave signal object.

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   [in] float Sinit,
   [in] float Sfinal,
   [in] float Sstep,
   [in] float Spulse,
   [in] float Tstep,
   [in] gcCTRLMODE CtrlMode
   );
```

Parameter

Pstat
Pstat object on which the square wave signal will run.

Sinit
Initial value in volts or amps.

Sfinal
Final value in volts or amps.

Sstep
Step in volts or amps.

Spulse
Pulse size in volts or amps.

Tstep
Step duration in seconds

CtrlMode
Potentiostat's control mode. The signal automatically adjusts the GstatRatio for signals in Galvanostat mode, so values are entered in amps, not volts.

Comments

This signal is commonly used to perform Square Wave Voltammetry. It is normally used with IGamryDtaqSqwv for data acquisition. The values passed in for **`Sinit`**, **`Sfinal`**, **`Spulse`**, and **`Sstep`** will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

See Also

IGamryDtaqSqwv

IGamryReadZ

Overview

The IGamryReadZ interface is exported by the GAMRYCOM library. It allows applications to acquire impedance data for a specified frequency and amplitude.

Member	Description
Init	Initialize a ReadZ object
Cook	Retreives points from the data acquisition queue
Measure	Take an impedance measurement
SetCycleLim	Set the minimum and maximum cycles to run
SetSpeed	Set the speed.
SetZmod	Set the initial guess of Impedance modulus.
Vreal	Return the real component of the measured Voltage
Vimag	Return the imaginary component of the measured Voltage
Vsig	Standard deviation of the measured Voltage
Vdc	Return the DC component of the measured Voltage
Ireal	Return the real component of the measured Current
Iimag	Return the imaginary component of the measured Current
Isig	Standard deviation of the measured Current
Idc	Return the DC component of the measured Current
Zreal	Return the real component of the measured Impedance
Zimag	Return the imaginary component of the measured Impedance
Zsig	Standard deviation of the measured Impedance
Zfreq	Return the DC component of the measured Impedance
Imod	Return the modulus of the measured Current
Iphz	Return the phase of the measured Current
Vmod	Return the modulus of the measured Voltage
Vphz	Return the phase of the measured Voltage
Zmod	Return the modulus of the measured Impedance
Zphz	Return the phase of the measured Impedance
Gain	Return the Gain value
VNoise	Returns the expected noise for the V channel measurement.
INoise	Returns the expected noise for the I channel measurement.
IENoise	Returns the expected noise for the IE channel measurement.
IERange	Returns the current IE Range used.
SetGain	Set the requested Gain.
SetVNoise	Set the expected noise for the V channel measurement.
SetINoise	Set the expected noise for the I channel measurement.
SetIENoise	Set the expected noise for the IE channel measurement.
SetIdc	Set the initial DC component of the measured Current
StatusMessage	Get the status message as set by a call to Measure

Init Method

Initialize the ReadZ object

Definition

```
HRESULT
Init(
   [in] IGamryPstat* Pstat,
   );
```

Parameter

Pstat
Pstat object on which ReadZ will run.

Comments

This method is called to initialize a ReadZ object. ReadZ objects must be initialized prior to further use.

See Also

Measure
Cook

Cook Method

Retrieve data points from the data acquisition queue.

Definition

```
HRESULT
Cook(
   [in, out] long* NumPoints,
   [out] SAFEARRAY(VARIANT)* Data
   );
```

Parameter

NumPoints
Number of Points to cook, returned as number of points actually cooked.

Data
SAFEARRAY containing cooked points.

Comments

This method is used by the application to retrieve data points from the data acquisition queue. This method must be called or data points may be lost. The acquisition queue is not sized to hold the entire acquisition, rather it is sized to hold enough points to give the application time to retrieve them.

Element	Value	Variant Type
0	I	VT_R4
1	V	VT_R4

See Also

Run
Stop

Measure Method

Performs an Impedance measurement at the given Frequency and Amplitude.

Definition

```
HRESULT
Measure(
   [in] float Frequency,
   [in] float Amplitude,
   );
```

Parameter

Frequency
Test Frequency in Hertz. The driver may use a frequency slightly different from the requested frequency due to rounding errors or in order to minimize noise.

Amplitude
The RMS signal to be applied in either Volts or Amps

Comments

The Measure method is used to take an impedance measurement. When a measurement is performed, certain changes to the hardware settings are required. The GamryReadZ object determines and sets these hardware setting (Gain stages, Current Range, Filters, Channel Offsets) appropriately. While the measurement is being performed, the _IGamryReadZEvent, OnDataAvailable, is fired when Lissajous (I,V) data is available. When the measurement has completed, an OnDataDone event is fired. This event contains information regarding the status of the measurement. Based on the return value of this event, Measure may need to be called again in order to achieve an actual Impedance value.

See Also

Init
Cook
Result
_IGamryReadZEvents::OnDataDone

SetCycleLim Method

Set the maximum and minimum number of cycles ReadZ can use to make its measurements

Definition

```
HRESULT
SetCycleLim(
   [in] long Min,
   [in] long Max
   );
```

Parameter

Min
Minimum number of cycles to be used in the measurement

Max
Maximum number of cycles to be used in the measurement

Comments

The first cycle which contains a startup transient distortion will not be counted. This function is available simply to allow setting the minimum and maximum number of cycles in a single call.

SetSpeed Method

Set the speed

Definition

```
HRESULT
SetSpeed(
   [in] gcREADZSPEED Speed
   );
```

Parameter

Speed

Set the speed:

0 ReadZSpeedFast

1 ReadZSpeedNorm

2 ReadZSpeedLow

See Also

ReadZ Speed Enum

SetZmod Method

Set the initial guess of Impedance modulus.

Definition

```
HRESULT
SetZmod(
   [in] float Zmod
   );
```

Parameter

Zmod
Initial Guess

See Also

Zmod

Vreal Method

Returns the real portion of the AC voltage

Definition

```
HRESULT
Vreal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC Voltage in Volts

See Also

Vimag
Vsig
Vdc
Ireal
Zreal

Vimag Method

Returns the imaginary portion of the AC voltage

Definition

```
HRESULT
Vimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC Voltage in Volts

See Also

Vreal
Vsig
Vdc
Iimag
Zimag

Vsig Method

Return the standard deviation of the measured voltage.

Definition

```
HRESULT
Vsig(
   [out,retval] float* Value
   );
```

Parameter

Value
Standard deviation in volts.

See Also

Vreal
Vimag
Vdc
Ireal
Zreal

Vdc Method

Return the measured DC voltage

Definition

```
HRESULT
Vdc(
   [out,retval] float* Value
   );
```

Parameter

`Value`
DC Voltage in Volts

See Also

Vreal
Vimag
Vsig
Ireal
Zreal

Ireal Method

Returns the real portion of the AC current

Definition

```
HRESULT
Ireal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC current in Amps

See Also

Iimag
Isig
Idc
Vreal
Zreal

Iimag Method

Returns the imaginary portion of the AC current

Definition

```
HRESULT
Iimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC current in Amps

See Also

Ireal
Isig
Idc
Vimag
Zimag

Isig Method

Return the standard deviation in the measured current

Definition

```
HRESULT
Isig(
   [out,retval] float* Value
   );
```

Parameter

Value
Value in amps.

See Also

Ireal
Iimag
Idc
Vreal
Zreal

Idc Method

Return the measured DC current

Definition

```
HRESULT
Idc(
  [out,retval] float* Value
  );
```

Parameter

Value
DC current in Amps

See Also

Ireal
Iimag
Isig
Vreal
Zreal

Zreal Method

Returns the real portion of the AC impedance

Definition

```
HRESULT
Zreal(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC impedance in Ohms

See Also

Zimag
Zsig
Zfreq
Vreal
Ireal

Zimag Method

Returns the imaginary portion of the AC impedance

Definition

```
HRESULT
Zimag(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS AC impedance in Ohms

See Also

Zreal
Zsig
Zfreq
Vimag
Iimag

Zsig Method

Return the standard deviation of the measured impedance.

Definition

```
HRESULT
Zsig(
   [out,retval] float* Value
   );
```

Parameter

Value
Value in ohms.

Comments

This value is set to 1.0 by default.

See Also

Zreal
Zimag
Zfreq
Vreal
Ireal

Zfreq Method

Return the measured frequency as measured

Definition

```
HRESULT
Zfreq(
   [out,retval] float* Value
   );
```

Parameter

Value
Frequency in Hertz

See Also

Zreal
Zimag
Zsig
Vreal
Ireal

Imod Method

Return the modulus of the AC current

Definition

```
HRESULT
Imod(
   [out,retval] float* Value
   );
```

Parameter

Value
RMS Magnitude of AC current in Amps

See Also

Vmod
Zmod

Iphz Method

Return the phase of the AC current

Definition

```
HRESULT
Iphz(
   [out,retval] float* Value
   );
```

Parameter

Value
Phase of AC current in Degrees

See Also

Vphz
Zphz

Vmod Method

Return the modulus (magnitude) of the AC voltage

Definition

```
HRESULT
Vmod(
   [out,retval] float* Value
   );
```

Parameter

`Value`
RMS Magnitude of AC voltage in Volts

See Also

Imod
Zmod

Vphz Method

Return the phase of the AC voltage

Definition

```
HRESULT
Vphz(
  [out,retval] float* Value
  );
```

Parameter

Value
Phase of AC voltage in Degrees

See Also

Iphz
Zphz

Zmod Method

Return the modulus (magnitude) of the AC impedance

Definition

```
HRESULT
Zmod(
   [out,retval] float* Value
   );
```

Parameter

Value

RMS Magnitude of AC impedance in Ohms

See Also

Vmod
Imod

Zphz Method

Return the phase of the AC impedance

Definition

```
HRESULT
Zphz(
   [out,retval] float* Value
   );
```

Parameter

Value
Phase of AC impedance in Degrees

See Also

Vphz
Iphz

Gain Method

Return the Gain value.

Definition

```
HRESULT
Gain(
   [out,retval] float* Value
   );
```

Parameter

Value
Currently set Gain value.

See Also

SetGain

VNoise Method

Return the expected noise for the V channel measurement.

Definition

```
HRESULT
VNoise(
   [out,retval] float* Value
   );
```

Parameter

Value
Currently set VNoise value.

See Also

SetVNoise

INoise Method

Returns the expected noise for the I channel measurement.

Definition

```
HRESULT
INoise(
   [out,retval] float* Value
   );
```

Parameter

Value
Currently set INoise value.

See Also

SetINoise

IENoise Method

Returns the expected noise for the IE channel measurement.

Definition

```
HRESULT
IENoise(
   [out,retval] float* Value
   );
```

Parameter

Value
Currently set IENoise value.

See Also

SetIENoise

IERange Method

Returns the current IE Range used by ReadZ

Definition

```
HRESULT
IERange(
   [out,retval] long* Value
   );
```

Parameter

Value
Currently set IE Range requested by ReadZ

Comments

This function is basically the same as a call to IGamryPstat::IERange.

See Also

IGamryPstat::IERange

SetGain Method

Set the requested Gain value.

Definition

```
HRESULT
SetGain(
   [in] float Gain
   );
```

Parameter

Gain
Requested Gain

See Also

Gain

SetVNoise Method

Set the expected noise for the V channel measurement.

Definition

```
HRESULT
SetVNoise(
   [in] float VNoise
   );
```

Parameter

VNoise
Expected VNoise value to set.

See Also

VNoise

SetINoise Method

Set the expected noise for the I channel measurement.

Definition

```
HRESULT
SetINoise(
   [in] float INoise
   );
```

Parameter

INoise
Expected INoise value to set.

See Also

INoise

SetIENoise Method

Set the expected noise for the IE channel measurement.

Definition

```
HRESULT
SetIENoise(
   [in] float IENoise
   );
```

Parameter

IENoise
Expected IENoise value to set.

See Also

IENoise

SetIdc Method

Set the initial DC component of the measured Current

Definition

```
HRESULT
SetIdc(
   [in] float Idc
   );
```

Parameter

Idc
Initial Idc value

See Also

Idc

StatusMessage Method

This function returns the label, or friendly name, of the Pstat.

Definition

```
HRESULT
StatusMessage(
   [out,retval] BSTR* Message
   );
```

Parameter

Message
A variable that receives the status message.

Comments

The **StatusMessage** method is used return an explanation of the most recent status as set by a call to Measure. This message should only be called after receiving an OnDataDone event, otherwise, the message may be meaningless.

_IGamryReadZEvents

Overview

The _IGamryReadZEvents interface provides the event call-back functions required to handle events issued by GamryReadZ objects.

The following table summarizes the members of _IGamryReadZEvents. The methods are described in detail in this section.

Member	Description
OnDataAvailable	ReadZ has data available for cooking
OnDataDone	ReadZ acquisition has completed

OnDataAvailable Event

This event is fired when ReadZ has data available for cooking.

Definition

```
void
OnDataAvailable();
```

Comments

As data is acquired by ReadZ, this event is fired to let the application know data has been placed in the data acquisition queue. The application should then make a call to Cook to retrieve this data. There may not be a one-one relationship with this event and the acquired data points. When data is being acquired quickly, one event may be fired even though multiple data points have been added to the queue.

Polling for data with the Cook method can be used in lieu of this event should there be threading issues. Some applications do not allow other events to be processed when certain user interface events are being generated. In this case, polling will provide a smoother reception method for the data.

See Also

Cook
OnDataDone

OnDataDone Event

This event is fired when data acquisition has completed

Definition

```
void
OnDataDone(
   [out] gcREADZSTATUS Status
   );
```

Comments

When a data acquisition cycle has been completed, this event is fired. This event signals the application that no more data will be placed in the data acquisition queue. This event does not signify that the queue is empty, however, and a final call to Cook should be made to process any points still in the queue. This event will return a status of **`ReadZStatusRetry`** if Measure needs to be called again. **`ReadZStatusRetry`** means that the measurement was not yet successful, and that adjustments to the hardware settings or signal applied need to be made. These settings will be adjusted in a subsequent call to Measure and need not be adjusted manually. If **`ReadZStatusError`** is returned, a subsequent call to Measure will most likely continue to fail, and there is a fundamental problem with the measurement. Call IGamryReadZ::Result for information about the error.

See Also

Cook
OnDataAvailable