GAMRYCOMLib::IGamrySignalArray Interface Reference
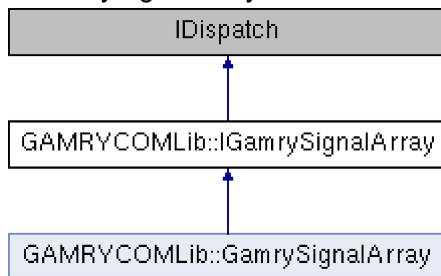
Array

Inheritance diagram for GAMRYCOMLib::IGamrySignalArray:

Public Member Functions

HRESULT	**Init** ([in] **IGamryPstat** *Pstat, [in] long Cycles, [in] float SampleRate, [in] long SamplesPerCycle, [in] SAFEARRAY(float)*SignalArray, [in] **gcCTRLMODE** CtrlMode) Initializes a Universal Signal object. More...
HRESULT	**Tweak** ([in] long Cycles, [in] float SampleRate, [in] long SamplesPerCycle, [in] SAFEARRAY(float)*SignalArray, [in] **gcSIGTWEAKSTYLE** Style) Adjusts the parameters of an Array Signal object. More...

Detailed Description

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 Function Documentation

HRESULT GAMRYCOMLib::IGamrySignalArray::Init ([in] IGamryPstat * **Pstat,**

	[in] long	**Cycles,**
	[in] float	**SampleRate,**
	[in] long	**SamplesPerCycle,**
	[in] SAFEARRAY(float)*	**SignalArray,**
	[in] gcCTRLMODE	**CtrlMode**
	)	

Initializes a Universal Signal object.

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.

`Cycles` is entered as a positive number or -1 for a "continuous" signal.

The values passed in for `SignalArray` will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

The following maximum sizes apply:

- The maximum size of the `SignalArray` is 262143 points.
- The maximum number of `Cycles` is (2^{16} - 1); continuous signals are repeated for a maximum of (2^{32} - 1) cycles.

See **IGamrySignalUniv** for a generalized version of this signal which includes options for cell state control and varying acquisition rates.

Parameters

`[in]`	**Pstat**	Pstat object on which the array signal will run
`[in]`	**Cycles**	Number of cycles to run
`[in]`	**SampleRate**	Time between data acquisition samples in seconds
`[in]`	**SamplesPerCycle**	Number of points in the `SignalArray`
`[in]`	**SignalArray**	Array of size `SamplesPerCycle` containing the points that make up the array
`[in]`	**CtrlMode**	Potentiostat's control mode; The signal automatically adjusts the GstatRatio for signals in Galvanostat mode.

Returns

HRESULT Error Status Code

See Also

Tweak

HRESULT GAMRYCOMLib::IGamrySignalArray::Tweak ([in] long **Cycles,**

[in] float **SampleRate,**

[in] long **SamplesPerCycle,**

[in] SAFEARRAY(float)* **SignalArray,**

[in] gcSIGTWEAKSTYLE **Style**

)

Adjusts the parameters of an Array Signal object.

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.

The values passed in for `SignalArray` will be interpreted as volts for potentiostat mode or amps for galvanostat mode.

Parameters

`[in]`	**Cycles**	Number of cycles to run
`[in]`	**SampleRate**	Time between data acquisition samples in seconds
`[in]`	**SamplesPerCycle**	Number of points in the `SignalArray`
`[in]`	**SignalArray**	Array of size `SamplesPerCycle` containing the points that make up the array
`[in]`	**Style**	The style of the tweak

Returns

HRESULT Error Status Code

See Also

Init

gcSIGTWEAKSTYLE