Steps for creating an application using GamryCOM

Introduction

The basic steps for creating a basic application with GamryCOM are detailed below. Please see the documentation and/or provided example code for more information. The specifics of each step will vary depending on the programming environment used.

Initialize Device List

1. Create an `IGamryDeviceList` object.
2. Before proceeding, a 3-second delay is recommended in order to allow all of the connected Gamry Devices to initialize.
3. `IGamryDeviceList.EnumSections()` returns a list of available devices.

Initialize the Pstat

1. Create an `IGamryPstat` object.
2. Initialize the object by calling `IGamryPstat.Init()` and pass in the requested device's Section.
3. Open the Pstat via `IGamryPstat.Open()`.

At this Point, the Pstat's settings may be modified via appropriate methods.

Initialize a Dtaq

1. Create an `IGamryDtaq` object appropriate for the application, `IGamryDtaqIv`, for example.
2. Initialize the Dtaq by calling the `IGamryDtaq`'s Init function and passing it your initialized Pstat object.
3. Register callback functions for `IGamryDtaq.OnDataDone` and `IGamryDtaq.OnDataAvailable`

Initialize a Signal

1. Create an `IGamrySignal` object appropriate for the application, `IGamrySignalArray`, for example.
2. Initialize the Signal by calling the `IGamrySignal`'s `Init()` function and passing it your initialize Pstat object, as well as its other required parameters.

Set Signal

1. The Pstat needs to be notified of the Signal being used. Call the Pstat's `SetSignal()` function, passing it the Signal Object that was initialized in the previous steps.

Depending on the programming environment being used, it may be necessary to cast the initialized Signal to the more generic `IGamrySignal` for this step.

Data Acquisition

1. Call the Dtaq's Run function.
2. Begin Data Acquisition by either:
   - Polling for data on a certain interval
   - Using the `OnDataAvailable` callback mechanism
3. Retrieve data from the Dtaq by calling its `Cook`() method, passing in the number of data points requested.

`Cook`() returns a two-dimensional array of data points. See the Dtaq's documentation for details on its contents.

4. Complete acquisition by:
   - Waiting for the `OnDataDone` callback mechanism
   - Calling the Dtaq's `Stop()` method.

Close and clean up all objects.