EPICS DATA ACQUISITION SOFTWARE AT THE CLS

G. Wright, R. Igarashi, Canadian Light Source, Saskatoon, Saskatchewan, Canada

Abstract

The Canadian Light Source (CLS) Data Acquisition library provides a simple scan and store interface for CLS beamlines. Originally intended as a tool for testing and commissioning, it has been used in *QT* and *GTK+* user applications at the beamlines. The current version supports dynamic loading of custom output modules to allow re-definable data transport methods and multiple simultaneous output formats.

INTRODUCTION

During the construction phase of the initial beamlines at the CLS, the need for a simple graphical user interface to perform simple motion control and record results became apparent. An application was written using the GTK+ toolkit. As time progressed, the scanning code was moved into its own library, and this library provides the building block for a number of applications at CLS beamlines. These applications now provide a signification portion of the data acquisition toolkit available at a number of the CLS beamlines.

KEY FEATURES

Configuration

A simple configuration file defines the scan and data collection (**events**).

All data acquisition structures can be accessed and manipulated by the calling program. Standard configuration files can be used to define the initial scan, and with simple data structure manipulation the acquisition can be modified without requiring a custom configuration file.

All Process Variables and range values for acquisition can be specified by macro strings. Again, a standard configuration file can be updated for different data ranges without dealing directly with the configuration.

The data output stream is passed to **Output Handlers** to determine where and how the data is dealt with. This provides the opportunity for customized handling for new data formats or visualization without rebuilding the library.

All scans and events run in independent threads. This allows simultaneous collection of data or recording of large data sets simultaneously with a positioner update.

Scans and Events

The acquisition library has two main components. The first component, the scan, is the definition of the control for the experiment – typically moving a device, and then requesting a detector to detect. A single scan typically only controls a single device. Scans can be nested, allowing multi-dimensional scans.

The second component is the event. An event is triggered from a scan, typically when a detector has finished reading. The event collects data from a list of process variables and requests an Output Handler to deal with the data.

Output Handlers

The section of code that generates the most controversy is the part that defines the output data format. In the data acquisition library, this part of the code has been generalized to a set of function calls that can be set at run time. An object-oriented "factory" for creating links to different handlers interfaces different data formats and different data transports to the acquisition library, allowing a great deal in flexibility on the calling applications' part to deal with new display and storage requirements, and even allow multiple simultaneous data files to be written in different formats.

The second benefit of using output handlers is that any viewer becomes generalized, and can either be integrated directly in the application (as was done for the *Motor Scan* screen in IDA (Interactive Data Acquisition, described later) or streamed to an independent application (such as BLGraph or Grace).

The output handlers have two components: the data format, and the data stream. The data format defines the appearance of the data, whether it's a simple comma separated value text output, or a compact binary output. The data stream is the destination, such as a data file, a named pipe, or a TCP/IP port.

Each instance of an output handler has a set of properties that can be updated through a standard application interface. An application can obtain information on properties and allow these to be controlled directly by the user, so new handlers can be configured without needing to update the user interface software.

Data Visualization

The primary tool used for data visualization with the acquisition library is **BLGraph**. This ROOT-based display tool is highly configurable. Dynamic selection of data fields for display, and manipulation of multiple fields with functions, as well as customizing configurations for quick set up of standard acquisition runs makes this a good match for acquisition library. As well, previously recorded data can be retrieved and viewed.

APPLICATIONS

QT Widget Support

The Qt Widget library from Nokia offers a powerful Rapid Application Development tool (**designer**) and the

ability to create new widgets that inherit from existing widgets to be used directly with **designer**. The acquisition widget has signal and slot support for the main functionality of the acquisition library.

The Data Acquisition GUI

The early separation of the acquisition library from the GUI has, in many ways, allowed more features to be added to the GUI. Initially written as a GTK+ application, the GUI was rewritten using QT.

IDA

The IDA application was written to handle fairly standard synchrotron beamline scans. In addition to setting run parameters for data acquisition, it also includes simple positioner scans (usually used to position a sample before scanning), run time estimates, detector selection, and callouts for energy detuning. The application has no hard-coded Process Variables: it uses a macro definition file to determine Process Variable names. This has allowed quick setup of synchrotron experiments at other beamlines.

IDAV

IDAV is customized for each beamline it runs on (currently SGM and SXRMB beamlines at the CLS). This minimizes the number of applications that need to be running to control and monitor the beamline while data acquisition is in progress.

Science Studio

The Science Studio project being developed at the CLS uses the acquisition library. This is currently in use at the VESPERS beamline.

nD Scanner

The nD scanner grew out of testing the multi-dimension scanning of the acquisition library and the ROOT support for the acquisition library. The application was originally intended as a commissioning tool, but has grown into an application for end-user data collection.

CERN ROOT SUPPORT

ROOT supports dynamic library loading. By providing a C++ class that calls the acquisition library, many of the ROOT features – including a C++ interpreter – are available when using the library. The rich display capabilities of ROOT make for quick work building a custom viewer which extends beyond the capabilities of existing viewers.

RECENT DATA CONFIGURATION FILE CHANGES

One of the most common pieces of code rewritten in each application is an input parser. Using an XML-compliant configuration file gives greater flexibility in input formats, and makes it easier to adapt to future changes in file format. The greatest potential is recording of property values for output handlers, avoiding separate configuration files for the output handlers or having a higher-level application need to rewrite the save-and-restore of output handler properties.

GOING FORWARD

Any system has many opportunities for improvement.

- Despite the gain of a simple configuration file to describe a complex data scan, there are still many opportunities for streamlining the configurations. To this end, support for partial configuration file loading would allow a configuration file per detector (or detector type) that would set up and read back data without knowing the type of experiment, and a partial file that knows how to run a scan for a beamline (but not know the type of detector) to quickly and easily expand the capabilities of a beamline when a new detector is added. The possibility exists that multiple detectors could be easily used together without an additional configuration being manually created.
- Expanding the possible actions to include calls to a language interpreter (e.g. python) would give greater dynamic capabilities to scans.
- Adding a python module to call the library would give benefits very similar to ROOT, but approachable to individuals who are familiar with python but not with ROOT.
- Scan types that aren't based on a start and end value, but rather some other condition would improve the simultaneous scan capability to have a master 'scan' with simultaneous scans that would recognize the completion of the master scan.
- Creating a 'sscan'-type EPICS record would allow embedding the scan in EPICS applications when necessary.

ACKNOWLEDGEMENTS

Jeff Warner provided the initial design work for IDA. David Chevrier created IDAV. The enthusiasm of many beamline scientists provided the inspiration to continuously improve the capabilities of the data acquisition system.

The Canadian Light Source is funded by

- University of Saskatchewan
- Canadian Institutes of Health Research (CIHR)
- National Research Council (NRC)
- Natural Sciences and Engineering Research Council (NSERC)
- Government of Saskatchewan
- Western Economic Diversification