

Introduction to FMWK Framework for LAr Experiments and Test-stands

Brian Rebel
July 2008

Outline

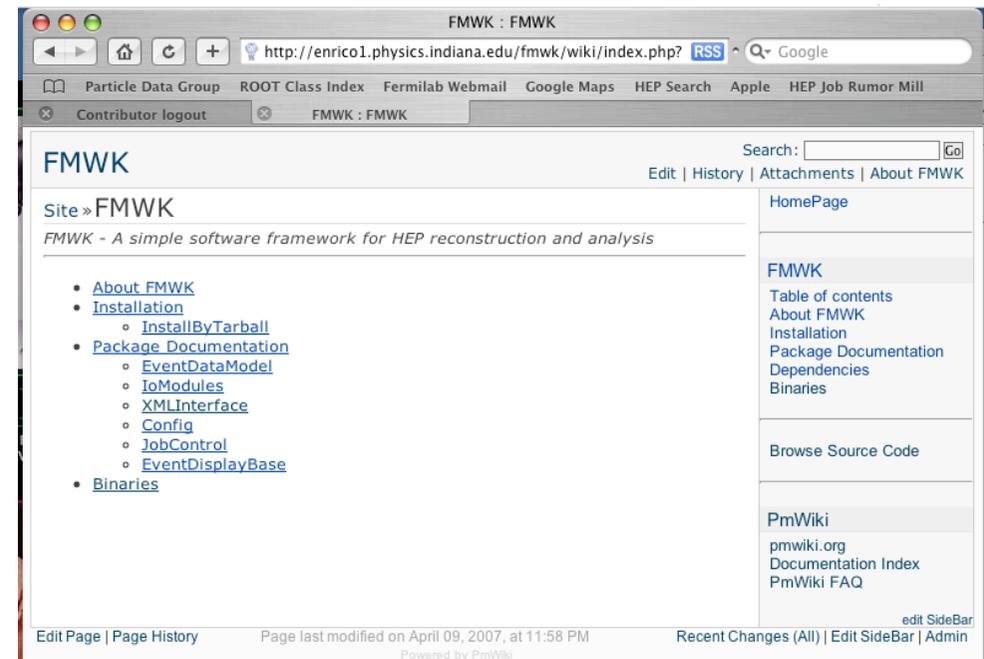


- Description of FMWK - what it is and what it does
- larsoft - SRT based distribution for software
- Progress so far and plans

FMWK



- FMWK is a generic analysis software framework designed for particle physics experiments
- The home page for FMWK is <http://enrico1.physics.indiana.edu/fmwk/wiki/index.php?n=Site.FMWK>
- Makes use of ROOT
- Designed to be lightweight
- Provides several basics packages
- Website has some documentation on all these classes



The FMWK packages

Most users will want to learn about the EventDataModel, Config, and JobControl packages. The IoModules, XMLInterface, and EventDisplayBase packages typically work behind the scenes.

"User interface" packages

- [EventDataModel](#) provides an interface to the event data store and allows storage of user-defined ROOT objects.
- [Config](#) provides a uniform interface to run-time configuration data
- [JobControl](#) defines the basic units of user reconstruction and analysis code and provides tools for assembling those modules into jobs

"Expert level" packages

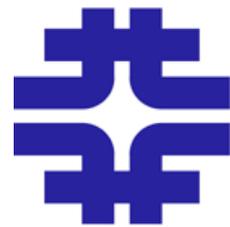
- [IoModules](#) provides the interface between the EventDataModel and I/O streams such as files
- [XMLInterface](#) provides an interface between the xerces-c XML parser and user code that uses XML to create and manipulate C++ objects
- [EventDisplayBase](#) provides GUI interfaces to the Config and JobControl packages and contains boilerplate code for generating menus and detector views
- [SRT FMWK](#)² SRT build configuration files

FMWK - Details

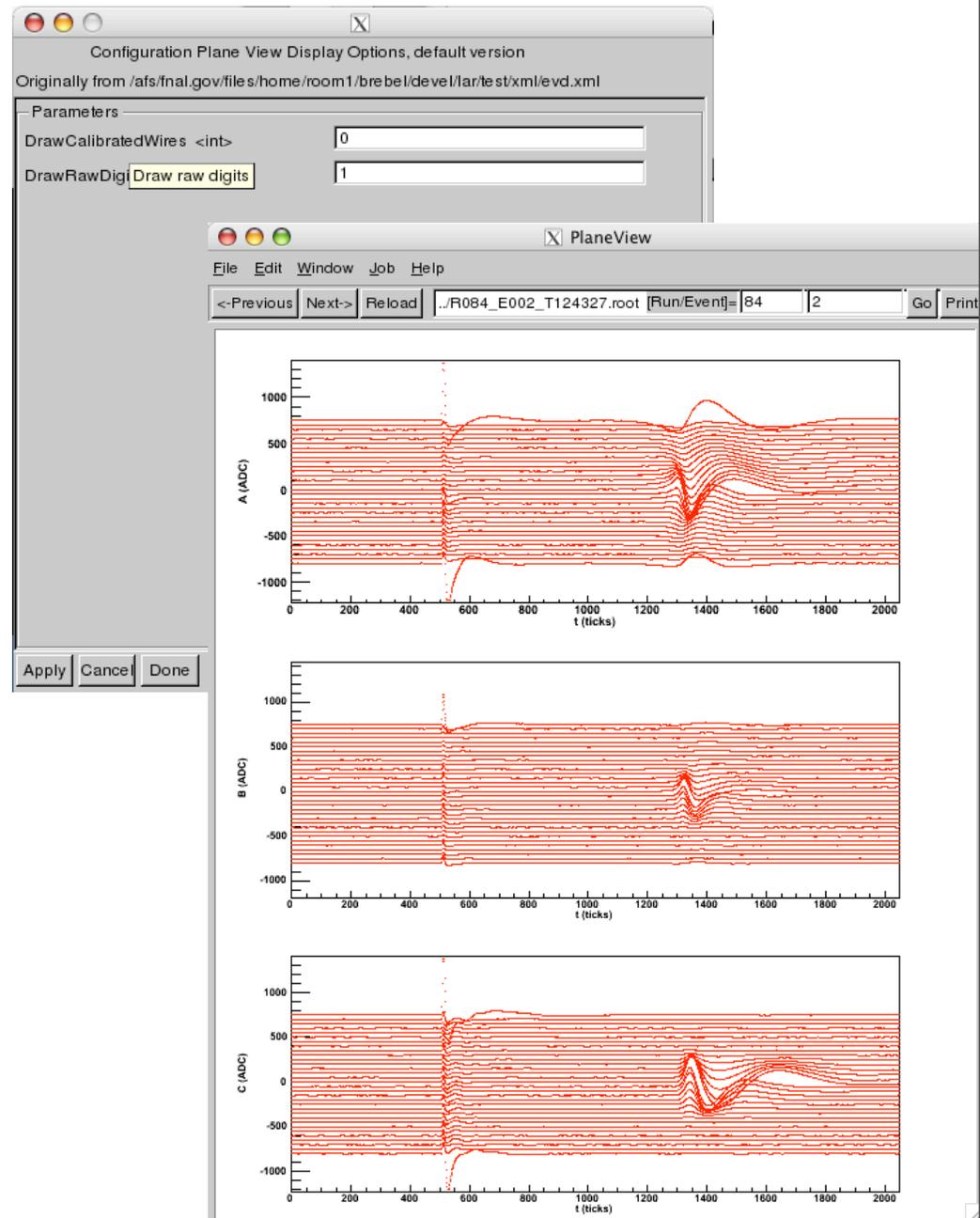


- All basic jobs are completed in FMWK
 - EVD handles keeping track of data, reconstruction objects, MC, etc.
 - Config allows for run time adjustment of parameters without recompiling
 - JobControl allows one to build up simple or complex jobs
- Expert level classes handle
 - I/O of ROOT files
 - XML interface for configuration files
 - GUI interfacing for event displays
- Requires 2 external packages
 - ROOT (Cern package for analysis/display)
 - Xerxes-C (XML interpreter)

LArSoft



- FMWK does not provide classes to do reconstruction or analysis - each experiment has to write their own
- larsoft is an SRT based distribution of code for the LAr experiments/efforts
- Currently has 4 additional packages to FMWK
 - EventDisplay - event display package using EventDisplayBase
 - RawData - package for defining raw data objects
 - CalData - package to take raw data and calibrate it (includes shaping code)
 - RecoBase - package to define basic reconstruction objects (tracks, showers, wires, etc)



LArSoft - Details



- larsoft is available on afs space - `/afs/fnal.gov/files/data/argoneut/d01/lar/larsoft`
- There is a setup script that can be run (source) from either cshell or bash environments - `larsoft/setup/setup_larsoft_fnal.(c)sh`
- To develop code you need a test directory (run the setup script first)
 - In your favorite directory do: `newrel -t XXX YYY`
 - XXX is the name of the larsoft release, usually development
 - YYY is the name of your new directory, usually test or something similar
 - cd into your directory and do `srt_setup -a`
 - Then `addpkg -h ZZZ`, where ZZZ is the name of the package you want to work on
 - If you want to start a new package, send email to brebel@fnal.gov with the package name and purpose and I will add it to the repository. Then you can do the addpkg step

Progress and Plans



- Basics for reconstruction and event display have been implemented
- Need to flesh out the detector Geometry
 - Kent has provided GDML code for Bo TPC, working on ArgoNeuT as well
 - Need to write package to load that code into the framework and navigate the detector geometries (should be flexible enough to work for Bo, ArgoNeuT, microBooNE, etc)
- Need actual reconstruction routines for picking out hits, making tracks, etc
 - Should liberally steal from Taritree and Bruce if possible
- Need code to make basic ROOT files from DAQ output - something is there for Bo, not for ArgoNeuT