Hall A/C Analysis Software
Introduction and Status

Ole Hansen

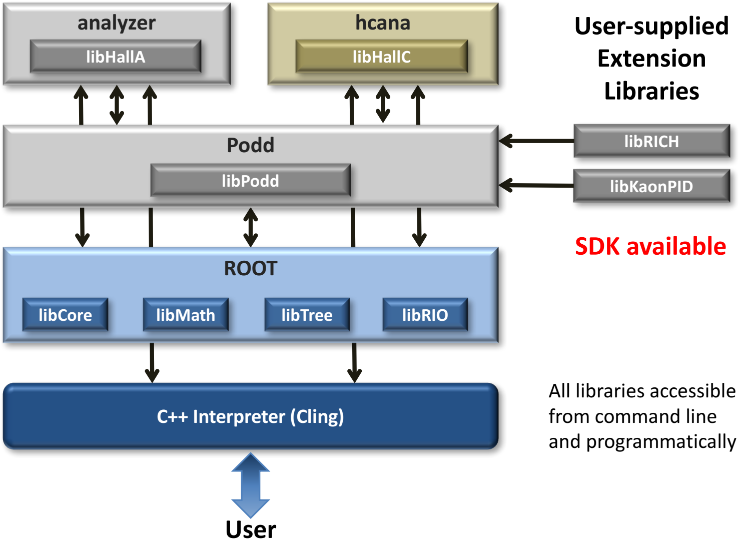
Jefferson Lab

Hall A/C Summer Collaboration Meeting
June 28, 2019

Hall A C++ Analyzer Framework (“Podd”)

- C++ class library built on top of ROOT
- Features
  - Modular. Easily accommodates different experimental setups
  - Run-time configurable via ROOT script & text files. No recompilation necessary
  - Light-weight. Minimal dependencies, small memory footprint
  - Supported on Linux and macOS with ROOT 5 & 6
  - Non-standard equipment analyzed with experiment-specific plug-in libraries
- Hall C analyzer “hcana” implemented as a special Podd library
  - Different database format
  - Special “report” feature

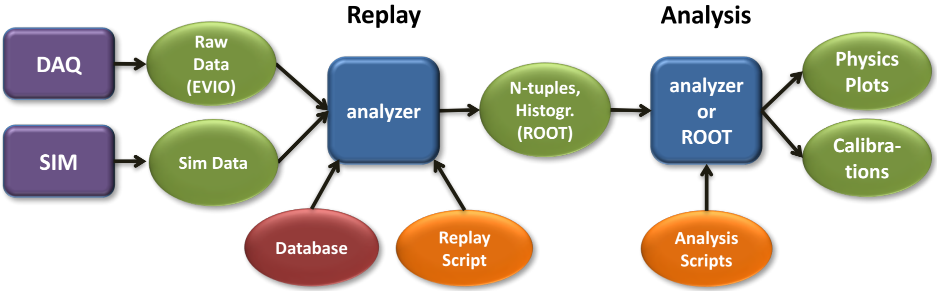
Podd/hcana as ROOT Extensions



SDK available

All libraries accessible
from command line
and programmatically

Analysis Flow



- Limitations
  - One-pass replay only: EVIO raw data → Flat ntuple-style ROOT trees + histograms
  - Currently single-threaded (serial) replay on single node only

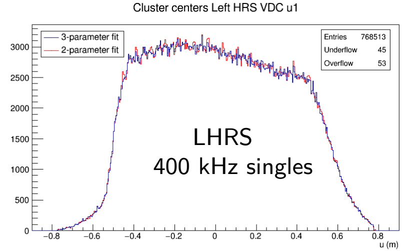
Podd Status

- Stable release: 1.6.6 (22 Feb 2019)
  - Used extensively by recent experiments (Tritium, APEX)
  - 1.6.6 adds optional `CMake` build system (from 1.7 branch)
  - Downloads and documentation at `https://redmine.jlab.org/projects/podd/wiki`
  - To be included in next release of JLab "Common Environment"
- Development version: 1.7.0-devel
  - Important new features (see next page)
  - Available on GitHub: `https://github.com/JeffersonLab/analyzer`
  - ETA: later in 2019

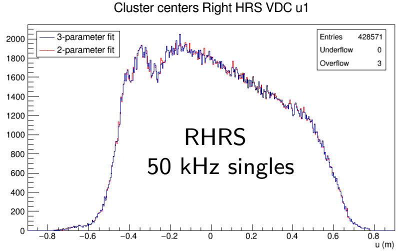
New in Podd 1.7

- High-rate VDC analysis (for APEX)
- Decoder upgrades
  - Supports CODA 3 data format, bank data and event block decoding (Bob Michaels)
  - Installs EVIO version 5.2 by default (better I/O performance and many bugfixes)
  - FADC decoders from Tritium experiments
- Abstracted database API
  - Lets hcana reuse Podd database readers
  - Allows easy integration of other backends (*e.g.* ccdb)
- Build system overhaul
  - `CMake` build system added (used by SBS, for example)
  - `SCons` build system significantly improved (used by `hcana`)
  - Old `make` system removed
- Extensive code cleanup & reorganization

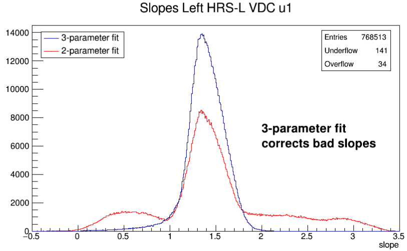
VDC Cluster Analysis—2-Parameter vs. 3-Parameter Fit


Cluster centers Left HRS VDC u1

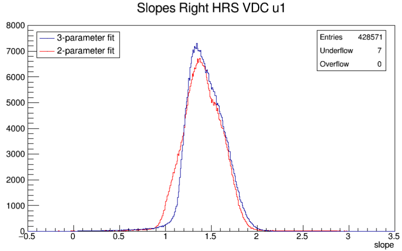
LHRS
400 kHz singles


Cluster centers Right HRS VDC u1

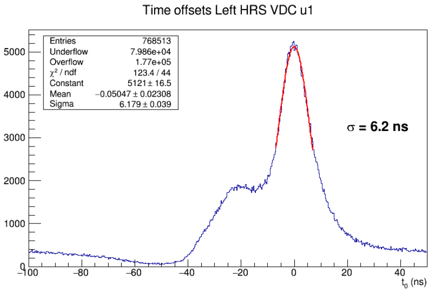
RHRS
50 kHz singles


Slopes Left HRS-L VDC u1

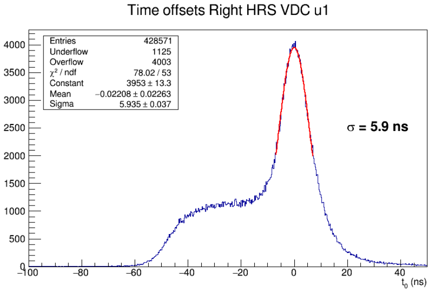
3-parameter fit corrects bad slopes


Slopes Right HRS VDC u1

VDC Cluster Analysis—Drift Time Offsets from 3-Parameter Fits



Time offsets Left HRS VDC u1



Time offsets Right HRS VDC u1

- Shoulders at negative t_0 not yet understood

Recent hcana Developments (from Mark Jones)

- Nov 2018. Add delayed helicity decoding. See commit and commit
- Jan 2019. Fix large number of memory leaks. See commit and talk
- Feb 2019. Fix wrong sign in coincidence time pathlength correction. See commit and talk
- April 2019. Fix mistake in THcCoinTime.h which had variable as Int_t that should be Double_t. See commit
- May 2019. Modify THcDC.cxx . Previously hardcoded if combined number of spacepoints in chambers >10 then no track was made. Changed limit of maximum combined number of spacepoints to 100. See commit

Plans for Podd 1.8+

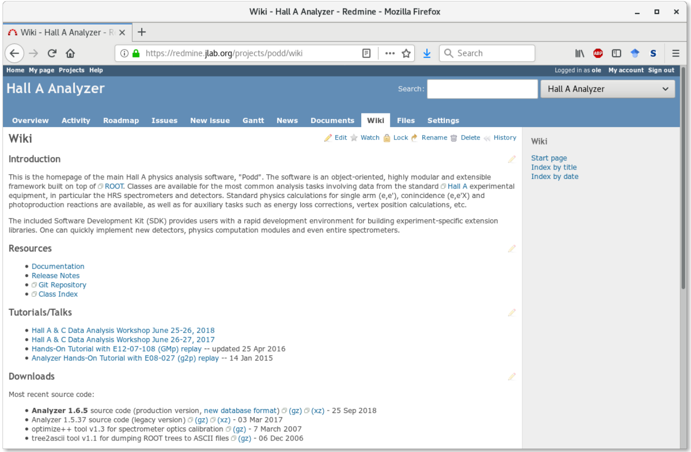
- Started/Considering (in anticipation of SBS)
  - Multithreading (10% done)
  - Output system upgrade (all data types, object variables; 75% done)
  - TBD: HIPO output file format support
  - TBD: EVIO 6 support (HIPO-like raw data files)
- Nice to have
  - Test suite (unit & integration tests)
  - Output metadata (configuration parameters, source & replay information)
  - Message facility (consistent log messages)

SBS Software Status & Plans

- SBS plan to use Podd framework. Anticipate to have multithreading available
- Standalone simulation well developed (`g4sbs`)
- Reconstruction library underway: `https://github.com/JeffersonLab/SBS-offline`
  - Decoders implemented for all subsystems
  - Optics & spin transport models done
  - GEM cluster finding & tracking under development (main challenge!)
  - Later: event display, online analysis
- Data handling will be challenging (by Hall A standards)
  - Raw data rates several GB/s. Will need preprocessing
  - Storage 200–1300 TB per experiment (sim+raw+prod) (4+ planned)
  - Simulation and analysis CPU requirements 1–4 M-core-hours (MCH) per experiment
  - Hall A farm quota is currently 4 MCH/year (5% of farm), probably need to double

Project Home: Redmine Wiki

- `https://redmine.jlab.org/projects/podd/wiki/`
- Integrated wiki, bug tracker, document database and more
- Old Podd website completely migrated (documentation etc.)
- *hcana* docs on Hall C wiki


Wiki - Hall A Analyzer - Redmine - Mozilla Firefox
Wiki - Hall A Analyzer - R
https://redmine.jlab.org/projects/podd/wiki
Search
Home My page Projects Help
Logged in as ole My account Sign out
Hall A Analyzer
Search:
Hall A Analyzer
Overview Activity Roadmap Issues New issue Gantt News Documents Wiki Files Settings
Edit Watch Lock Rename Delete History
Wiki
Start page
Index by title
Index by date

Wiki

Introduction

This is the homepage of the main Hall A physics analysis software, "Podd". The software is an object-oriented, highly modular and extensible framework built on top of ROOT. Classes are available for the most common analysis tasks involving data from the standard Hall A experimental equipment, in particular the HRS spectrometers and detectors. Standard physics calculations for single arm (e,e'), conincidence (e,e'X) and photoproduction reactions are available, as well as for auxiliary tasks such as energy loss corrections, vertex position calculations, etc.

The included Software Development Kit (SDK) provides users with a rapid development environment for building experiment-specific extension libraries. One can quickly implement new detectors, physics computation modules and even entire spectrometers.

Resources

- Documentation
- Release Notes
- Git Repository
- Class Index

Tutorials/Talks

- Hall A & C Data Analysis Workshop June 25-26, 2018
- Hall A & C Data Analysis Workshop June 26-27, 2017
- Hands-On Tutorial with E12-07-108 (GMp) replay -- updated 25 Apr 2016
- Analyzer Hands-On Tutorial with E08-027 (g2p) replay -- 14 Jan 2015

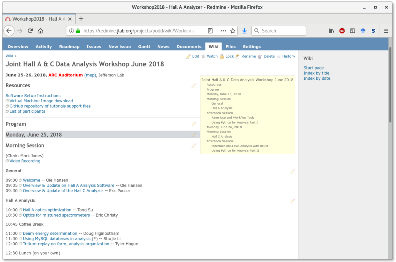
Downloads

Most recent source code:

- **Analyzer 1.6.5** source code (production version, new database format) (gz) (xz) - 25 Sep 2018
- Analyzer 1.5.37 source code (legacy version) (gz) (xz) - 03 Mar 2017
- optimize++ tool v1.3 for spectrometer optics calibration (gz) - 7 March 2007
- tree2ascii tool v1.1 for dumping ROOT trees to ASCII files (gz) - 06 Dec 2006

Good Starting Point for New Users: Analysis Workshops 2017/2018

- Workshop pages linked on main wiki
- Joint Hall A & C analysis workshops in summers 2017 & 2018
- Live hands-on tutorials, using preconfigured virtual machine environment
- Simulation, calibration, on- & offline data analysis, ROOT basics, etc.
- BlueJeans recordings available (linked on workshop page, CUE login required)



Joint Hall A & C Data Analysis Workshop June 2018

June 25-26, 2018, ARC Auditorium (map), Jefferson Lab

Resources

Software Setup Instructions
Virtual Machine Image download
GitHub repository of tutorials support files
List of participants

Program

Monday, June 25, 2018

Morning Session

(Chair: Mark Jones)
Video Recording

General

09:00 Welcome -- Ole Hansen
09:05 Overview & Update on Hall A Analysis Software -- Ole Hansen
09:30 Overview & Update of the Hall C Analyzer -- Eric Pooser

Hall A Analysis

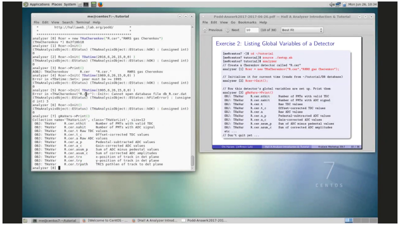
10:00 Hall A optics optimization -- Tong Su
10:30 Optics for mistuned spectrometers -- Eric Christy

10:45 Coffee Break

11:00 Beam energy determination -- Doug Higinbotham
11:30 Using MySQL databases in analysis (*) -- Shujie Li
12:00 Tritium replay on farm, analysis organization -- Tyler Hague

12:30 Lunch (on your own)

Joint Hall A & C Data Analysis Workshop June 2018
Resources
Program
Monday, June 25, 2018
Morning Session
General
Hall A Analysis
Afternoon Session
Farm Use and Workflow Tools
Using Python for Analysis Part I
Tuesday, June 26, 2018
Morning Session
Hall C Analysis
Afternoon Session
Intermediate-Level Analysis with ROOT
Using Python for Analysis Part II


Applications Places System
Mon Jun 26, 10:34
me@centos7:~/tutorial
File Edit View Search Terminal Help
Podd-Anawrk2017-2017-06-26.pdf — Hall A Analyzer Introduction & Tutorial
File Edit View Go Bookmarks Help
Previous
Next
10
(10 of 38)
Best Fit
Ole Hansen (Jefferson Lab)
Hall A Analyzer Introduction & Tutorial
Analysis Workshop 2017
10 / 36
7
CENTOS

```
 *        http://hallaweb.jlab.org/podd/        *
 *                                              *
 ************************************************
analyzer [0] Rcer = new THaCherenkov("R.cer","RHRS gas Cherenkov")
(THaCherenkov *) 0x2f18b10
analyzer [1] Rcer->Init()
(THaAnalysisObject::EStatus) (THaAnalysisObject::EStatus::kOK) : (unsigned int)
0
analyzer [2] Rcer->Init( TDatime(2016,6,26,15,0,0) )
(THaAnalysisObject::EStatus) (THaAnalysisObject::EStatus::kOK) : (unsigned int)
0
analyzer [3] Rcer->Print()
AOBJ: THaCherenkov      R.cer   "R.cer."         RHRS gas Cherenkov
analyzer [4] Rcer->Init( TDatime(1989,6,26,15,0,0) )
Error in <TDatime::Set>: year must be >= 1995
(THaAnalysisObject::EStatus) (THaAnalysisObject::EStatus::kOK) : (unsigned int)
0
analyzer [5] Rcer->Init( TDatime(1995,6,26,15,0,0) )
Error in <THaCherenkov("R.cer")::Init>: Cannot open database file db_R.cer.dat
(THaAnalysisObject::EStatus) (THaAnalysisObject::EStatus::kFileError) : (unsigne
d int) 3
analyzer [6] Rcer->Init()
(THaAnalysisObject::EStatus) (THaAnalysisObject::EStatus::kOK) : (unsigned int)
0
analyzer [7] gHaVars->Print()
Collection name='THaVarList', class='THaVarList', size=12
 OBJ: THaVar    R.cer.nthit     Number of PMTs with valid TDC
 OBJ: THaVar    R.cer.nahit     Number of PMTs with ADC signal
 OBJ: THaVar    R.cer.t Raw TDC values
 OBJ: THaVar    R.cer.t_c       Offset-corrected TDC values
 OBJ: THaVar    R.cer.a Raw ADC values
 OBJ: THaVar    R.cer.a_p       Pedestal-subtracted ADC values
 OBJ: THaVar    R.cer.a_c       Gain-corrected ADC values
 OBJ: THaVar    R.cer.asum_p    Sum of ADC minus pedestal values
 OBJ: THaVar    R.cer.asum_c    Sum of corrected ADC amplitudes
 OBJ: THaVar    R.cer.trx       x-position of track in det plane
 OBJ: THaVar    R.cer.try       y-position of track in det plane
 OBJ: THaVar    R.cer.trpath    TRCS pathlen of track to det plane
analyzer [8]
```

Exercise 2: Listing Global Variables of a Detector

```
[me@centos7 ~]$ cd ~/tutorial
[me@centos7 tutorial]$ source ./setup.sh
[me@centos7 tutorial]$ analyzer
// Create a Cherenkov detector called "R.cer"
analyzer [1] Rcer = new THaCherenkov("R.cer","RHRS gas Cherenkov");

// Initialize it for current time (reads from ~/tutorial/DB database)
analyzer [2] Rcer->Init();

// Now this detector's global variables are set up. Print them
analyzer [3] gHaVars->Print()
 OBJ: THaVar    R.cer.nthit     Number of PMTs with valid TDC
 OBJ: THaVar    R.cer.nahit     Number of PMTs with ADC signal
 OBJ: THaVar    R.cer.t         Raw TDC values
 OBJ: THaVar    R.cer.t_c       Offset-corrected TDC values
 OBJ: THaVar    R.cer.a         Raw ADC values
 OBJ: THaVar    R.cer.a_p       Pedestal-subtracted ADC values
 OBJ: THaVar    R.cer.a_c       Gain-corrected ADC values
 OBJ: THaVar    R.cer.asum_p    Sum of ADC minus pedestal values
 OBJ: THaVar    R.cer.asum_c    Sum of corrected ADC amplitudes
 etc ...
// Don't quit yet ...
```

Next Workshop

- Two most recent analysis workshops drew strong interest. Do another one?
- Would like your input as to
  - Interest
  - Scope
  - Level
  - Date
- Instead of show of hands, please complete a quick online survey (5 minutes of your time):

  `https://www.surveymonkey.com/r/3YHR3HD`

  Thanks :) The survey will be available until next Friday, July 5.

Summary

- Hall A & C analysis software is alive and well. It is actively maintained and used by current experiments in both halls.
- Significant development work (*e.g.* multithreading) is advisable for Hall A's upcoming SBS program.
- Many learning resources, documentation and examples exist, in part thanks to recent analysis workshops.
- Please complete the online survey to help us plan a future analysis workshop.