# Hall A Analysis Software

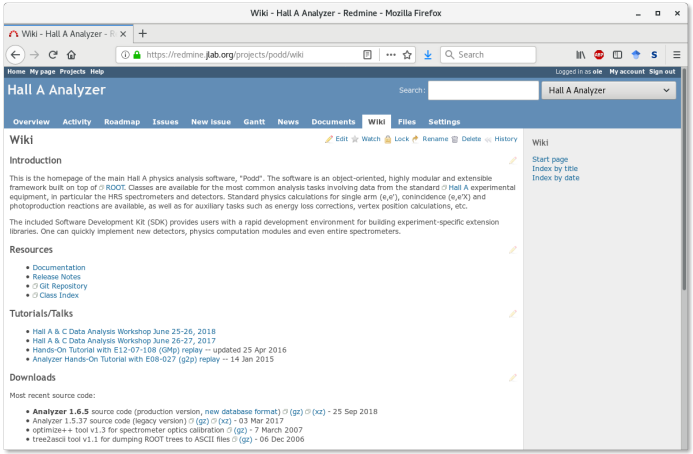
Ole Hansen

Jefferson Lab

Hall A Collaboration Meeting
January 31, 2019

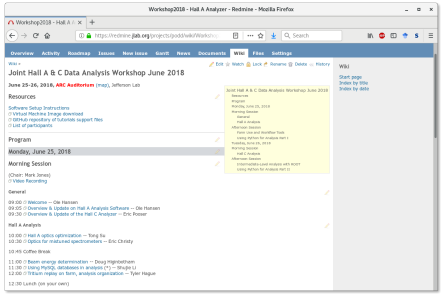
# New Home on the Web: Redmine Wiki

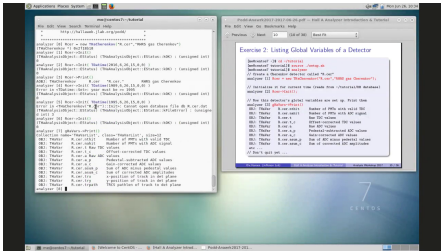
- `https://redmine.jlab.org/projects/podd/wiki/`
- Integrated wiki, bug tracker, document database and more
- Old website completely migrated (documentation etc.)



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

- 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, login required)

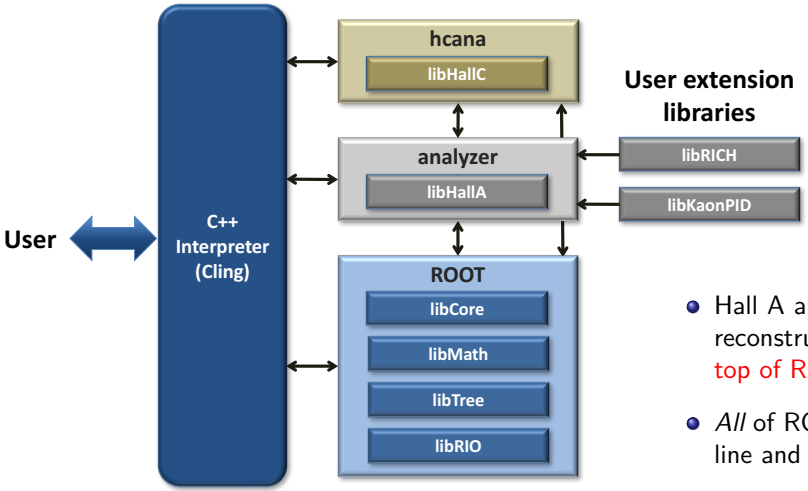


# Hall A C++ Analyzer ("Podd") Framework

- Design goals:
  - Highly modular to accommodate frequently changing experimental setups
  - Run-time configurable
- C++ class library built on top of ROOT. Steering via ROOT's C++ interpreter.
- Strengths
  - Light-weight: minimal dependencies, small memory footprint
  - Apparently quite user-friendly: students learn easily (but correct me if I'm wrong)
  - Output & cuts configurable at run time via text files. Flat text file database
  - Works with ROOT 5 & 6, on current and older Linux and macOS
  - Adequate for Hall A & C-style spectrometer analyses
- Limitations
  - Designed for one-pass analysis only:
    EVIO raw data $\rightarrow$ Flat ntuple-style ROOT trees + histograms
  - Single-threaded & not distributed

# Plug-In Architecture




**SDK available**

- Hall A analyzer = Library of reconstruction & analysis classes on top of ROOT
- *All* of ROOT available at command line and programmatically

# Hall A Analyzer Status

- Stable release: 1.6.5 (25 Sep 2018)
  - Improved VDC track reconstruction (properly handles multi-cluster events)
  - Modular decoder (Bob Michaels)
  - Decoder support for pipelined electronics (FADC250, F1TDC, etc.)
  - New database format
  - Used extensively by recent Tritium experiments
- Development version: 1.7.0-devel
  - Mostly a maintenance release (see next page)
  - Available on GitHub: `https://github.com/JeffersonLab/analyzer`
  - ETA: Spring 2019

# New in 1.7

- Decoder improvements
  - Experimental support for CODA 3 data format and bank data (Bob Michaels)
  - EVIO updated to version 5.2 (better I/O performance and many bugfixes)
- Contributions from Tritium experiments
  - Support for decoding FADC data in many detector classes
  - (TBD)
- 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 "under the hood" code maintenance
  - Database API generalized
  - HallA and Podd libraries separated
  - Fixed "defects" found by static code analysis tools (mostly trivial)

# Roadmap

- Completed with version 1.7
  - Database interface
  - Code reorganization
- Started (probably relevant for SBS)
  - Output system overhaul (all data types, object variables; 75% done)
  - Multithreading (10% done)
- Nice to have
  - Unit & integration tests, test suite
  - Metadata & data provenance in output file, self-documenting output
  - Message facility (consistent log messages)

# Output Module Overhaul

Currently:

- Only "D" (`Double_t`) data type
- Multiple, redundant array size variables with parallel arrays

```
*Br   15 :L.vdc.u1.time : data[Ndata.L.vdc.u1.time]/D                    *
*Entries :   265460 : Total  Size=   11970559 bytes  File Size  =    3662638 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=  3.27     *
*Br   16 :L.vdc.u1.trdist : data[Ndata.L.vdc.u1.trdist]/D                *
*Entries :   265460 : Total  Size=   11970731 bytes  File Size  =   10832974 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=  1.10     *
*Br   17 :L.vdc.u1.wire : data[Ndata.L.vdc.u1.wire]/D                    *
*Entries :   265460 : Total  Size=   11970559 bytes  File Size  =    2295168 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=  5.21     *
*Br   18 :L.vdc.u1.nclust : L.vdc.u1.nclust/D                            *
*Entries :   265460 : Total  Size=    2136991 bytes  File Size  =      45063 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression= 47.36     *
*Br   19 :L.vdc.u1.nhit : L.vdc.u1.nhit/D                                *
*Entries :   265460 : Total  Size=    2136721 bytes  File Size  =     184378 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression= 11.57     *
analyzer [2]
```

Improved:

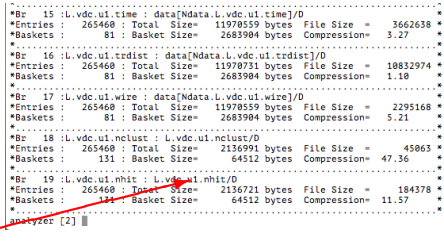
- All basic data types
- Optional automatic detection of parallel arrays, only one size variable (with limitations), saves space
- Optional automatic basket size adjustment (overriding ROOT default)

```
analyzer [6] T->Print("L.vdc.*")
*Tree    :T         : Hall A Analyzer Output DST                        *
*Entries :   280231 : Total =       325369054 bytes  File Size = 186808850 *
*        :          : Tree compression factor =   1.74                  *
*Br    0 :L.vdc.u1.nhit : L.vdc.u1.nhit/I                               *
*Entries :   280231 : Total  Size=    1122414 bytes  File Size  =     160443 *
*Baskets :       12 : Basket Size=     537088 bytes  Compression=  6.99     *
*Br    1 :L.vdc.u1.wire : L.vdc.u1.wire[Ndata.L.vdc.u1.wire]/I          *
*Entries :   280231 : Total  Size=    7420715 bytes  File Size  =    2064016 *
*Baskets :       49 : Basket Size=    1556480 bytes  Compression=  3.59     *
*Br    2 :L.vdc.u1.rawtime : L.vdc.u1.rawtime[Ndata.L.vdc.u1.wire]/I    *
*Entries :   280231 : Total  Size=    7420874 bytes  File Size  =    3614951 *
*Baskets :       49 : Basket Size=    1556992 bytes  Compression=  2.05     *
*Br    3 :L.vdc.u1.time : L.vdc.u1.time[Ndata.L.vdc.u1.wire]/D          *
*Entries :   280231 : Total  Size=   13718165 bytes  File Size  =    4116377 *
*Baskets :       80 : Basket Size=    2575872 bytes  Compression=  3.33     *
*Br    4 :L.vdc.u1.dist : L.vdc.u1.dist[Ndata.L.vdc.u1.wire]/D          *
*Entries :   280231 : Total  Size=   13718165 bytes  File Size  =   12032644 *
*Baskets :       80 : Basket Size=    2575872 bytes  Compression=  1.14     *
```

# Output Module Overhaul

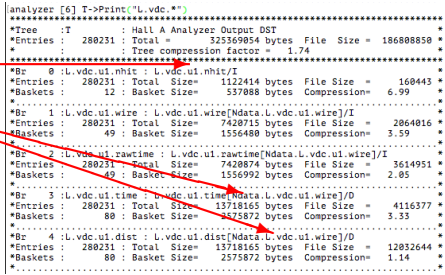
Currently:

- Only "D" (`Double_t`) data type
- Multiple, redundant array size variables with parallel arrays



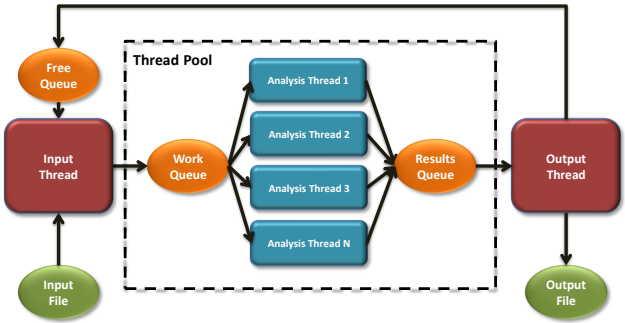

Improved:

- All basic data types
- Optional automatic detection of parallel arrays, only one size variable (with limitations), saves space
- Optional automatic basket size adjustment (overriding ROOT default)



# Parallelization/Multithreading

- Thread Pool with three thread-safe queues
- Queues hold working sets: event object, analysis chain & modules
- Option to sync event stream at certain events (*e.g.* scaler events, run boundaries)
- Option to preserve strict event ordering (at a performance penalty)
- More advanced designs exist, *e.g.* task-based dynamic worker allocation (in use at LHC). Podd would not be able to support those.



# Summary

- Hall A analysis framework has largely entered maintenance mode for the remainder of the HRS era (thru 2020)
- Recent extensive code quality checks did not reveal significant defects. Code is stable and reliable and appears to meet the needs of current experiments
- Upgrade possibilities exist (as always), some of which will be important for the upcoming SBS program