

Rivet monthly dev meeting

1 February 2023

Recent activity / TODOs

- **heprivet.org!** Anjelo working on fixing current block to CI deployment
- **Rivet ongoing developments** ⇒ [Review MRs](#)
 - Tomek P more convenience-function support to header: ONNX-Runtime
 - Max K & Chris G identified issue in some B & tau decay analyses: switch to Particle::isLastWith(hasBottom) works nicely
 - AB extended linspace() for reversed ranges, add powspace(), powdbnspace(), fnspace()
 - Another large bulk arrival of Peter R analyses! Plus more from ATLAS, LHCb, etc.:
3.1.x release needed
 - Tomek's p3() -> ThreeMomentum rather than Vector3 branch finally fixed & merged
 - Move private members to protected in DISLepton: do this generally in v3.2+?
- CMS (Markus S & Hannes J) flagged a bug in dressed-lepton definition:
 - Change from fromDecay() to isDirect(fromTau=false,fromMuon=false) semantics prevents FSR photons from being dressed in Rivet3 **AB to investigate/propose**
- Minor build issues:
 - --with-hepmc3 flag-duplication "fix" reverted
 - rivet-herwig Docker: still a problem with Herwig bootstrap... from Rivet?

Recent activity / TODOs

- Other developments

- YODA2

- Chris G huge effort on YODA2 C++ and Python interface: done?!
- AB: Cython str/bytes autoconversion finished (was blocked by I/O detail)
- HepData output for YODA2: nearly done, validation in progress [MR](#)
- **Release strategy:** Rivet cannot support both YODA1 and YODA2 (ref data will change) → **release together with plotting as 3.2**
- HDF5 format development shelved until YODA 2.1

- New plotting machinery

- More from Jamie & Yoran on matplotlib plotting ⇒ **mpl in Y1.9.8 / R3.1.8!**

- Documentation

- **TODO:** rivet.core Python doc hack
- CG/AB to register rivetyoda and hepmcnet Mastodon handles?
 - Federated... so no need to pre-empt unless we want a particular server

- **Native LHEF input support** (maybe that was obvious to everyone else?)

- Via HepMC3 input auto-discovery — working as of today's HepMC3 master: [!57](#)

Misc + AOB

- **CERN gitlab accessibility via MCnet/LPCC**
 - **TODO:** reserve group/project/repo names on gitlab.cern
 - **TODO:** request Mac CI runners (Intel *and* M1) on CERN Gitlab!
- **Events, schools, talk requests? + working meetings...**
 - *QED object defs / MC workshop via IPPP, 7-9 Dec*
 - *BSM reinterpretation forum, CERN/hybrid, 12-15 Dec*
 - Both good fun, nice to be back
 - Keenness on more CMS / ATLAS / FCC / other Rivet & Contur tutorials
 - Lunga dropped the ball on replying with details, **AB to prod?** No money
 - Cumberland Lodge got in touch with JMB saying they have availability in the summer (still no money though...)
 - Other bids? Louie at Clermont...?
 - Anyone fancy a Mediterranean yacht-physics BSM workshop? (Seriously...)
 - Who is planning to go to [Les Houches](#) and which session?
 - + MCnet meeting CERN 24-28 Apr + MCnet school Durham 10-14 July

BACKUP

Major-release tasks

- In parallel: work toward v3.2.0 — baseline without YODA2
 - ~~CPU saving no copy of the HepMC event, with API constness change: merged~~
 - ~~Finish and merge thread safety branch (important for Gambit → Tomek Procter)~~
 - ~~Add early versions of automatic “object flattening”, and no width scaling~~
 - ~~Merge HDF5 analysis data, and live/dead conversion branches → AB~~
 - Plotting merge (+ CHC patches) → Jamie, Yoran, AB, CB, etc.: **meeting needed**
 - Jet clustering of any ParticleBase: some reclustering devel, nuanced due to need to propagate constituents / recluster, maybe needs a proj subclass → **AB**
 - Deprecation clean-out and enum rationalisation (started)
 - including “enumification” of the DISKinematics options arg :-/ → **AB**
 - Primary particles definition / enforcement
 - **mix of PIDs and decay time; Leif started tech discussion**

Path towards YODA2

- **Plan for major version release around summer!**
- ~~Finalise translation of the usual YODA 1.9 objects into YODA2 style objects in time for Easter~~
 - ~~Support all usual histogram/profile/scatter object types + new (continuously) binned Estimates~~
- Spend some time after Easter on (more validation and) syntactic sugaring of discretely binned axes
 - If this cannot be incorporated with reasonable turn-around (e.g. too complicated or other distractions get in the way), propose to postpone user-friendly support of discrete binning to a later YODA 2.1 release (autumn/winter?)
My current feeling is this won't be necessary, though ... 🙌
- Outstanding Todos:
 - ~~Finalise Estimate implementation~~
 - Syntactic sugaring for ~~BinnedStorage<Estimate, BinnedAxis>~~ (+ discretely binned axes)
 - Reduce operations (e.g. for live-to-dead conversion)
 - Update Python API (🤖)
 - Need new I/O reader and writer
 - Update docs with practical examples
 - Update build tests, tweak CI if necessary + validate, validate, validate, ...

Big picture tasks (near duplicate from April & May)

- **Stats objects are our major technical bottleneck**

- Integrate and extend new plotting system
- YODA type-extension (build on Nick R GSoC 2020 work)
- HDF analysis data and new YODA format
- `post-finalize()` always “flatten” stats objects to “binned measurement” type
- [finish multiweight-fill optimisation (Aditya GSoC 2020)]

- **Scaling**

- Analysis distribution system... again
- Ref-data and analysis data particularly problematic: decouple data from code??
- HD consistency

- **Standardising:**

- MCnet weight-name/structure [proposal](#): productive meeting on May 21, lots of agreement, AB to update and recirculate proposal
- [Event-record content: excessive size and physicality...]



Major (stats) work plans

- **“Flattening”**: convert finalize output to inert objects (scatters/binneameas)
 - Final objects really will mean “what was plotted/listed in the paper”
 - Allow eager conversion to solve “no-bin-width issue”
 - Best that we wait for binned measurement YODA2 types: no more scatters!
- **HDF5 analysis data machinery (Holger) Status?**
 - Also interested in HepMC and YODA HDF5 formats
 - **Holger to ping CMS, prototype interface**
- **Plotting (Christian B et al)**
 - Plan: generate Python MPL scripts *without* TeX, .plot styles → YAML
 - Rivet labels tested: MathText fails due to missing std symbols. Can we extend?
 - Stalled for a while... restarting? Possible student help from David Grellscheid
 - **Christian to prototype the Python-script generation**
 - **Chris to extract weight-handling logic from rivet-cmhistos**

Performance in Rivet and YODA (Aditya Kumar, AB)

- **Profiling revealed bottlenecks: thanks Aditya!**

- HepMC ASCII I/O (obviously) — taken out of tests by event-reuse
- GenEvent copying — for sanitising, but hardly used: removed from Rivet.
Could/should generators write smaller “essential” events by default?
Awkwardness: we still normalise GenEvent units... so not quite analysing a const GenEvent.
But can't justify an expensive copy for *unit conversion*...
- PID functions — sped up charge lookups by special-cases. Marginal gain
- Multiweight calls to histo fill() *very* expensive: ~40-50% CPU!
100+ consecutive fills with same x: tried caching in YODA but no benefit:
cache-check costs the same as linear bin lookup! *Maybe cache in Rivet?*

- **Thread-safety.** *“Just store a ProjectionHandler in AnalysisHandler: easy!”...?*

- But then who do Projection constructors (recursively) register their contained projections with, before they themselves have been bound to a PH?
- “Declare queue” implemented: not yet working (thx, unique_ptr), but should do
What should the Projection ownership be?!

YODA generalised datatypes (Nick Rozinsky, LC, AB)

- **Long-understood limitations of YODA types and design**
 - Overreach in attempted non-factorisable binnings: composed 1D axes are fine
 - Complexity/mess in 2D overflows: need “infinity binning”
 - Need for binned “dead” data objects... or any type, actually
 - Want programmatic access to axis number and global/local bin indexing
 - Want labelled/discrete binnings as well as continuous
 - Code duplication, particularly in Cython interface building
- **Major YODA redesign using modern C++ magic. Thanks Nick!**
 - E.g. Histo1D → wrapper of a BinnedStorage<CAxis, Dbn<1>> + sugar
 - + arbitrary mixtures, e.g. 3D binnings of doubles, discretely labelled counters, ...
 - Adaptors used to map fill/set behaviours. Can do the same for I/O read/write?
- **Path to a YODA2 release:**
 - Needs I/O adaptors and user-facing refinements. Tie in with HDF5 format?