Bruno: status and new developments

- Packaging
- Improving production workflow

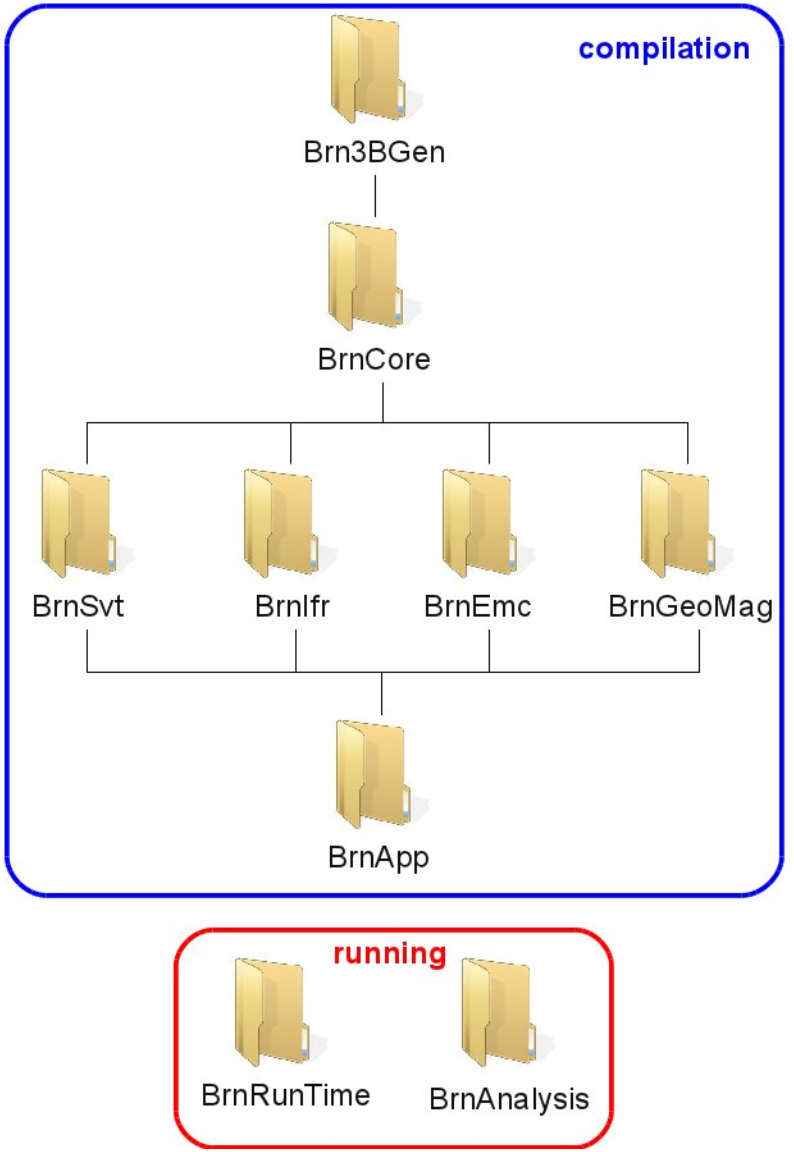
A. Di Simone
INFN Tor Vergata

Packaging – at last!

- Main activity in the past weeks was to complete the packaging of the "old" Bruno software
- Reminder:
  - Packaging means dividing the code base into modular, self consistent units
  - It helps code maintenance
  - It helps code development, in the sense that it enforces clear separation between different functionality, possibly handing over responsibility to different persons
  - It encourages detectors to take more control of their own simulation, by creating detector-specific packages
- Status:
  - Used production (i.e. "code freeze") to perform the migration, starting from a prototype I prepared some months ago
  - Packaged version is now committed/validated, in sync with r481 of the "old" version
  - Feedback and bugfixes from production not included, though
    - Will need some level of porting forward the changes

Package structure

- **BrnCore**: implements the core simulation functionality, including file/tree management, MCTruth recording, physics-list-related code.
- **Brn3BGen**: implements the RadBhabha generator.
- **BrnIfr**: IFR-specific code (like hits and sensitive detectors)
- **BrnSvt**: SVT-specific code (like hits and sensitive detectors)
- **BrnEmc**: EMC-specific code (like hits and sensitive detectors)
- **BrnGeoMag**: contains the code needed for geometry and magnetic field construction.
- **BrnApp**: contains the code for the main executable
- **BrnRunTime**: this is where the executable must be run from. Contains gdml and mac files
- **BrnAnalysis**: container to hold analysis code (i.e. ROOT macros)



Comments

- Brn3BGen should depend on BrnCore, not the other way around
  - This will need to be addressed in the context of a revision of the way we deal with external generators
- Packaging was not just moving files around: a significant amount of code needed to be modified
  - AnalysisManager was largely reduced. It is now a “normal” G4 user action (not a singleton anymore), and it is used only to create histograms on the fly
    - Please keep it like this
  - All file/tree creation is now done centrally through a FileManager:
    - **refrain** from opening/closing/filling by hand your own root files/trees. Files/trees are created in several places across Brn, and you have no way of being sure that when you fill a tree or close a file all other clients have really finished
    - All files/trees accessed through the FileManager are closed/filled at the right moment

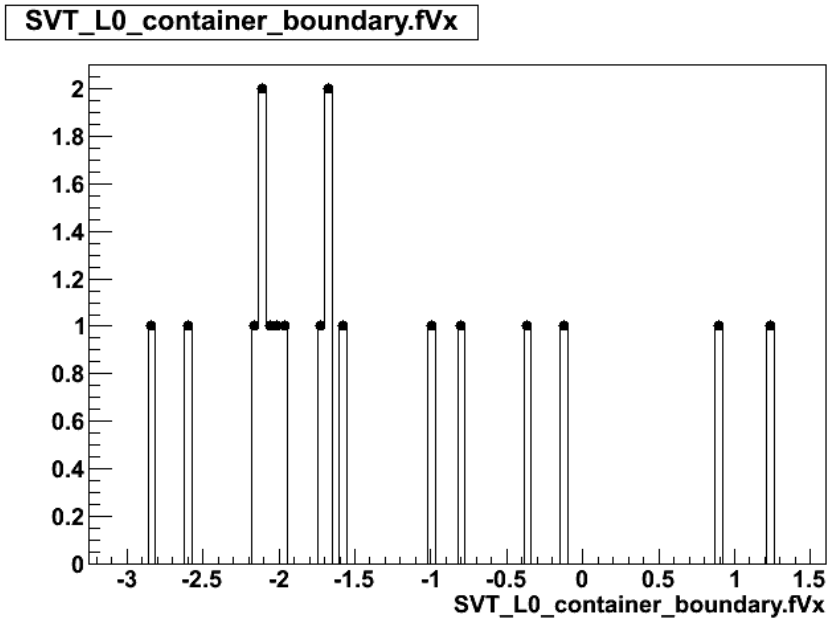
Comments (2)

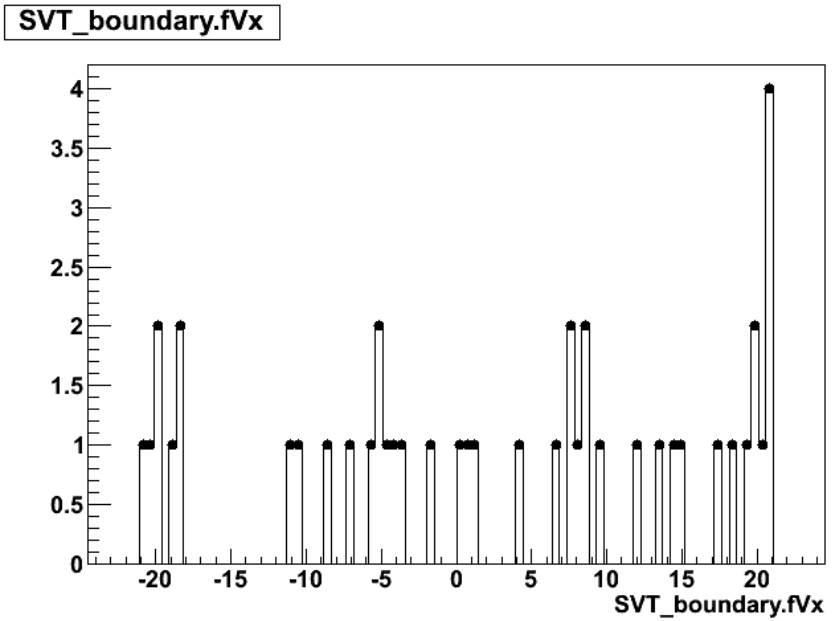
- Writing of hits to ROOT file used to be done through AnalysisManager
  - All that code has been moved to the SensitiveDetectors
  - This means that hit writing is now under the responsibility of the subdetectors
    - Keep it in mind when writing a new SensitiveDetector
- No central facility for histogram creation (yet), but appending histograms to a file should be done through the FileManager
  - This ensures that you are not cd()-ing into a folder somebody else may not want to write into
  - See BrnDetSurvey for an example
- Migration was also an occasion to do some cleanup
  - For example, all Bruno* and My* files have been renamed to the (now default) Brn* prefix.

Validation

- Validated with some single-e and RadBhabha events, comparing the results of original and packaged versions
  - Most convincing evidence: *the random number sequence is the same*
  - However, some physics-related plots have been prepared as well
    - Truth validation: look at boundary information
    - Hit validation: look at number of hits/hits energy

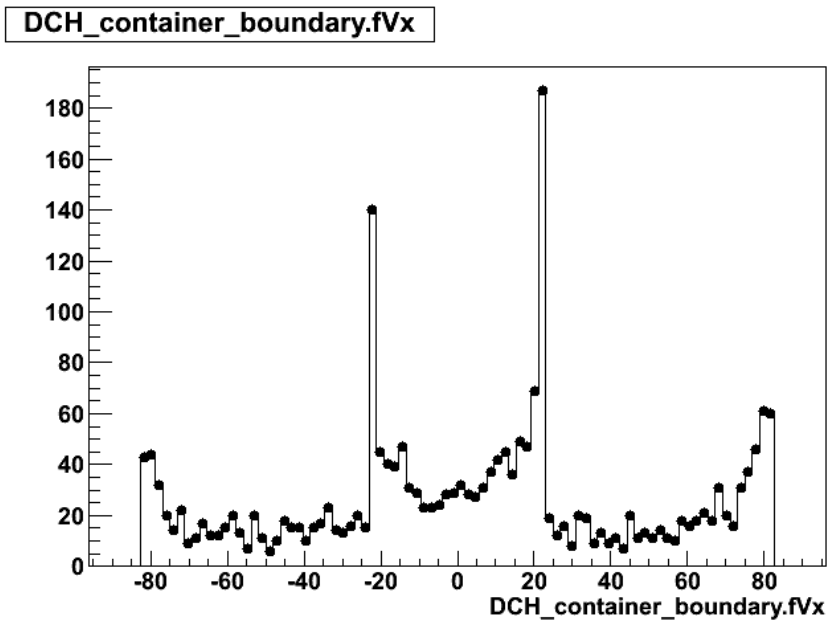
Boundaries: SVT

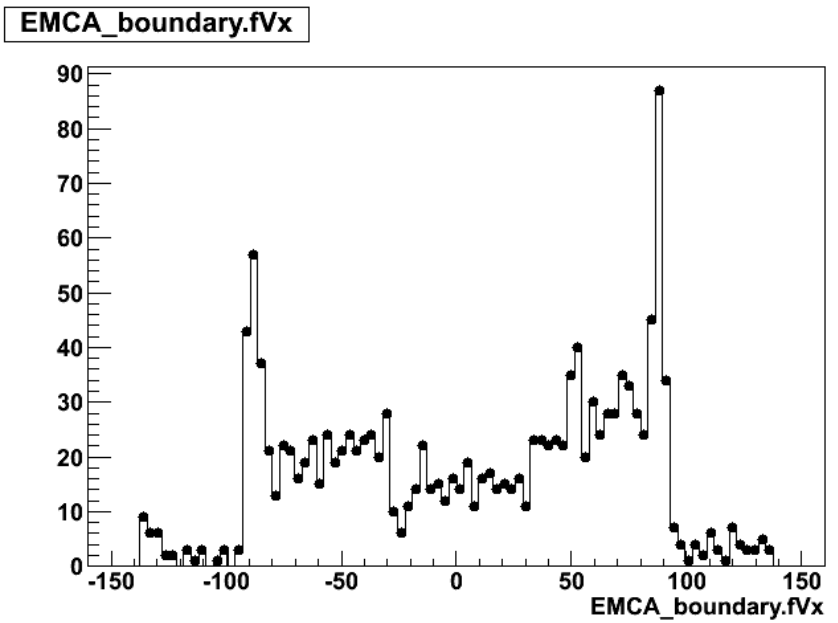




- x-coordinate of particle crossing point with SVT (+L0)
  - Markers: packaged version
  - Line: original version

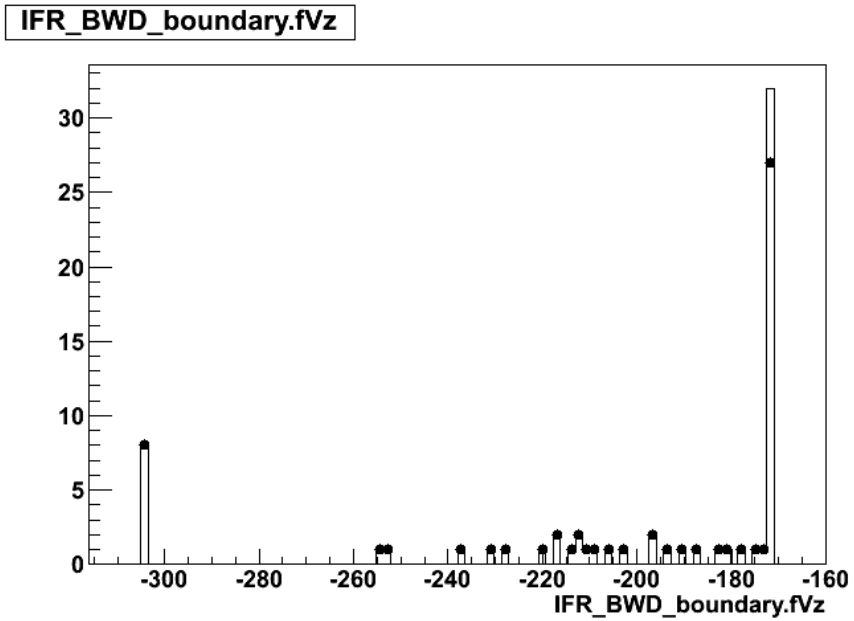
Boundaries: DCH/EMC

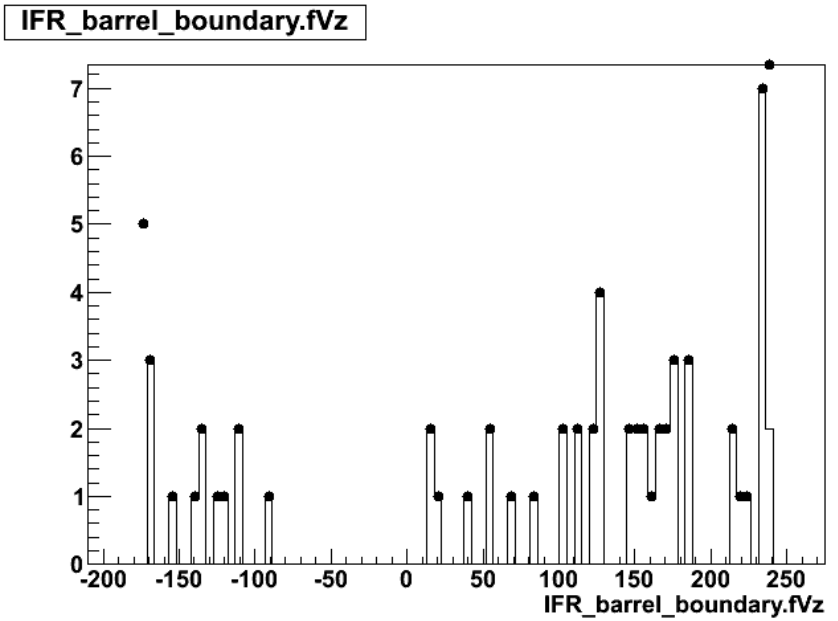




- x-coordinate of particle crossing point with DCH and EMC
  - Markers: packaged version
  - Line: original version

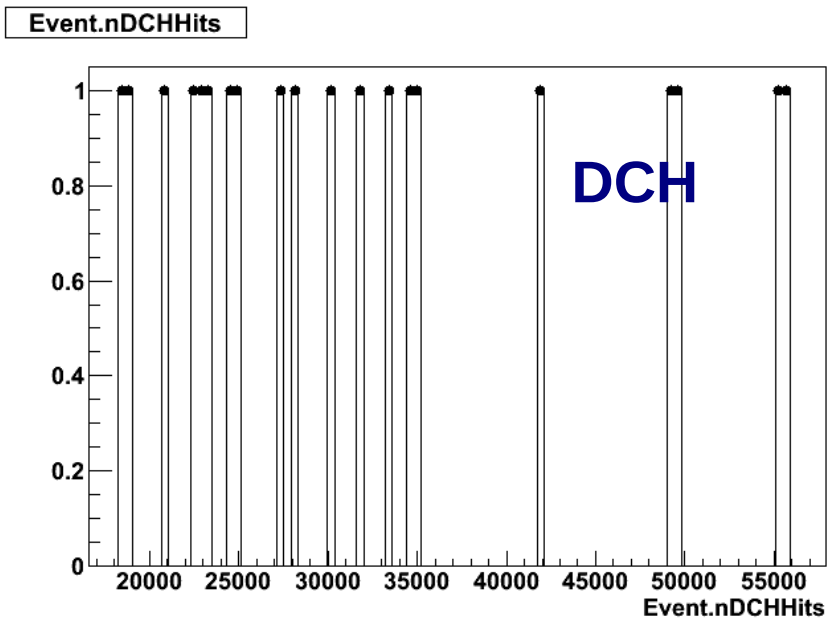
Boundaries: IFR

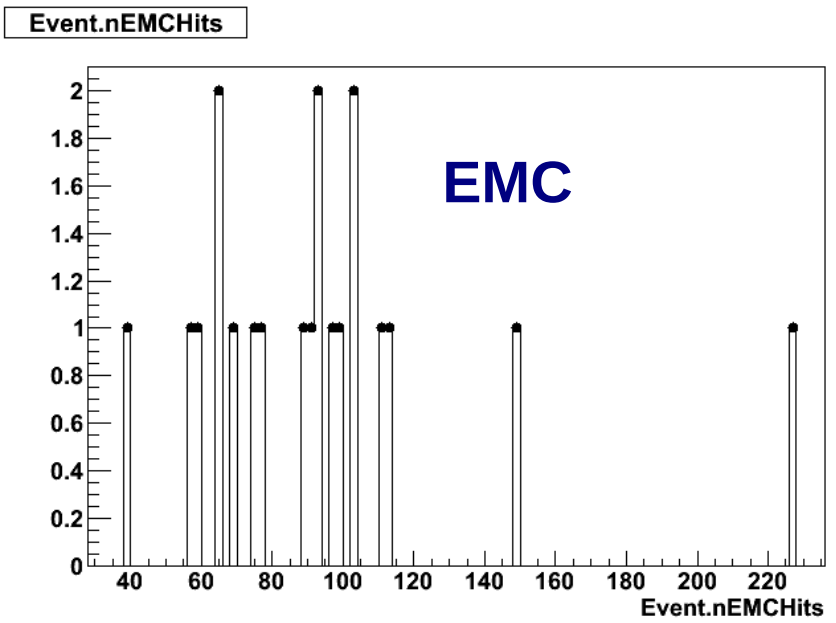


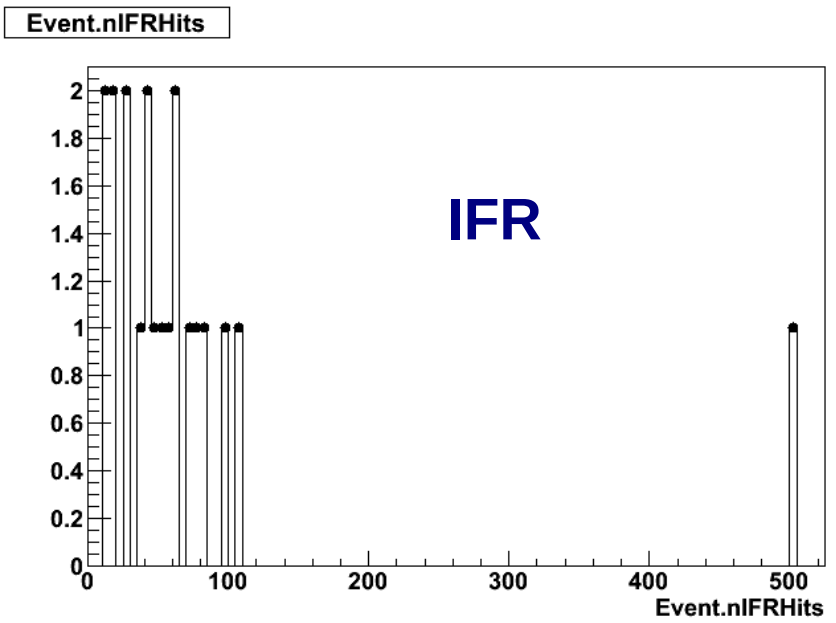


- z-coordinate of particle crossing point with IFR
- some discrepancies in the number of particles crossing barrel and fwd/bwd, due to touching surfaces
  - In these conditions particle is recorded only in one of the two surfaces: differences between the two versions point to some minor issue probably worth investigating
    - Note that no physical information is lost (i.e. all particles are properly recorded, they are just moved from one surface to the coinciding one)

Hits: number

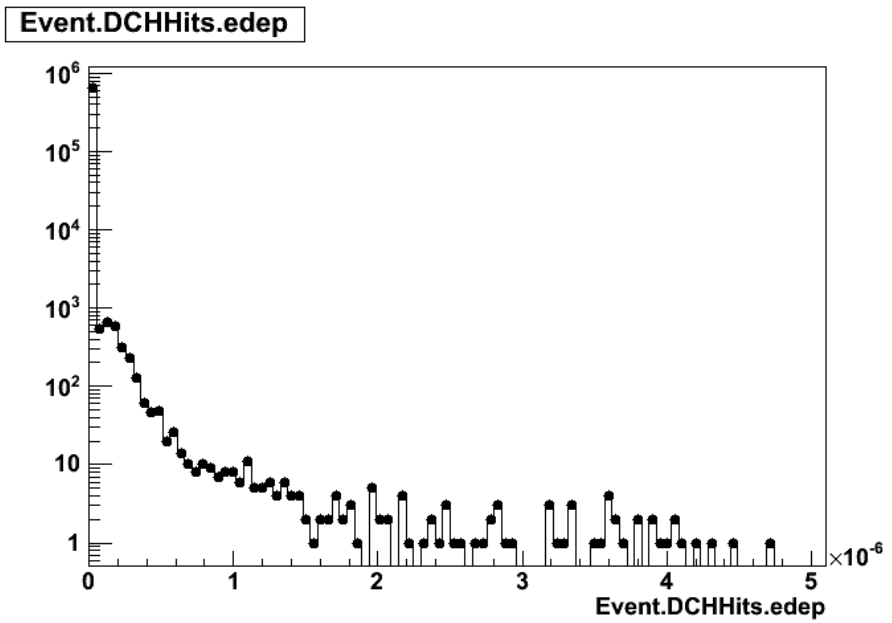


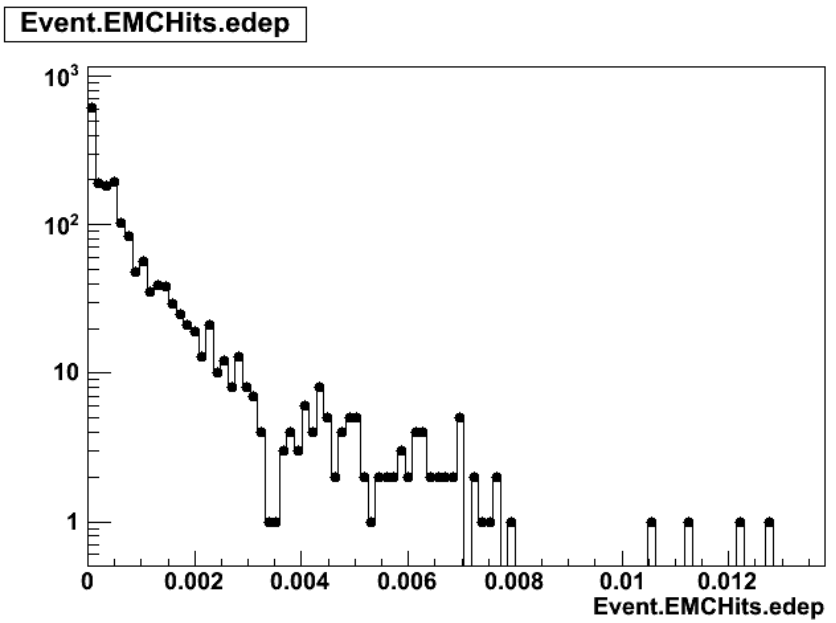




- Number of hits recorded by some subdetectors

Hits: energy





- Energies deposited in DCH and EMC

Plans

- We should consider completely blocking write access to the old Bruno svn
  - Now possible only to “selected” users, in order to allow fixes needed for production
- Last-minute commits to “old” Bruno must be ported forward to the packaged version
  - I suggest that whoever did the commits takes care of sync-ing them
- Integration with prodsys must be ensured
  - From the point of view of an end user, the only change is the name of the executable, so hopefully this will not be too hard, but needs to be tested
- More subdetectors may want to create their own package
  - Consult with us to evaluate the best way to proceed
- Automatic unit test is a very interesting possibility
  - Contact with M. Corvo to understand how much can be done using CTest

Plans

- In the medium term, I strongly suggest we start improving our persistent structures
  - Both hits and truth information
- Do we still need a monolithic event?
  - Now just a wrapper around hits collections from subdetectors
  - Maybe we could just store the collections as branches, without gluing them in the event
    - Advantage is improved modularity, i.e. one could run with only some detectors on, and only find the corresponding hits in the output
      - As opposed to having empty collections for the missing detectors
    - It also makes package dependency cleaner, although present usage of TClonesArray avoids most compile-time dependencies

Path to production

- Luckily, we have reached a point where productions will become more frequent, and more mission-critical
- We need a clearly established path to be followed in order to ensure good-quality results
- Idea could be to identify a set of steps to be taken, with explicit sign-off by relevant contacts
- Trac milestones could be used to track/document/bookkeep/archive the process
- Will show in the following a preliminary list of tasks, based on experience from past productions

Path to production: 1

- software preparation/validation:
  - feature-freeze: a sensible amount of time before production, production software needs to go into feature freeze. only critical bugfixes can be accepted
  - memory leaks should be assessed, and either fixed or declared tolerable. Explicit sign-off required.
  - estimate the expected CPU time per job, and the total disk space needed. Explicit sign-off required.
- release validation:
  - productions should be run on releases (+patches)
  - after a release is built, the software validation needs to be repeated on it. Explicit sign-off required.
  - release deployment: if distributed resources are used, some level of release validation needs to available on remote sites as well. Explicit sign-off required.

Path to production: 2

- automation of software/release validation is highly desirable. In lack of an automatic test, however, validation MUST be carried-on anyway. By hand, if necessary. Without explicit sign-off, no productions should start.
- pre-production: a (possibly small) fraction of the total events needs to be produced before launching the production. feedback from the pre-production must to be twofold:
  - software-oriented: job crash rate, memory/CPUtime requirements, disk space requirements, further check on memory leaks. It is foreseeable that rare problems would be spotted only at this stage, being unobservable on the smallest scale validation. Hopefully, though, these will tend to be tolerable. In the worst case, patches must be provided. Explicit sign-off required.
  - physics oriented: compliance of physics results with production motivation. Concerning patches, same considerations as above apply. Explicit sign-off required.

Path to production: example

2011 Prod: 3. pre-production (2 matches)

Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified
#6	Runtime performance ok	component1	new	None		task	major	somebody	05/27/11
#7	Physics results ok	component1	new	None		task	major	somebody	05/27/11

2011 Prod: 2. Release validation (2 matches)

Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified
#4	Release validated	component1	new	None		task	major	somebody	05/27/11
#5	Remote sites validated	component1	new	None		task	major	somebody	05/27/11

2011 Prod: 1. Software preparation (3 matches)

Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified
#3	Resources estimate	component1	new	None		task	major	somebody	05/27/11
#2	Software quality ok	component1	closed	fixed		task	major	somebody	05/27/11
#1	Feature Freeze	component1	closed	fixed		task	major	somebody	05/27/11

- Example of workflow mapping into our trac system
- Better options may be available on the market, but the message is that it takes five minutes to setup

Path to production: comments

- Discussed briefly with some other collaborators, and consensus was reached that this could be a good starting point
- Your first reaction may be that this takes too much time, and production results tend to be needed ASAP
  - Of course exceptions can be made
  - However, recent experience showed that performing incomplete validation to gain time actually results in time being lost
    - Problems don't disappear just because you don't have time to look into them
- I propose to start experimenting this workflow during next production round
  - Needs some level of enforcement (and tolerance) by all of us

Conclusions

- Bruno packaging is presently complete
  - All efforts must be taken so that the “old” version is not developed anymore
  - At the very least, whoever commits anything to the old version MUST port it forward to the new one
  - Documentation provided at
    - https://sbrepo.pd.infn.it:8911/projects/BRN
- Recent production showed us the limits of our present production workflow
  - A more rigorous approach may benefit the effectiveness of production efforts