

ROOT Implementation of the MINI (ESD)

Status Report

RAL Workshop
13.-14. Jan. 2003

Leif H. Wilden

What is the MINI ? (technically)

- Set of approx. 20 persistent classes.
Involved packages:
 - DchDataR, DrcDataR,
EmcDataR, IfrDataR,
L1GltDataR, L3TDataR,
PidDataR, SvtDataR,
TrkDataR, ...
- Prototype implementations exist for some classes
(thanks to OCP, Pete and Matthias)

Remaining Problems to be Solved

- How to deal with non-simple data member types:
 - write them directly using ROOT's foreign class support (see next slides)
 - write a wrapper TObject class with custom streamer
- There seem to be problems with I/O of doubly templated classes
 - I am look at it, most likely it is a mistake that I am making...

Foreign Class Support in ROOT

- *Foreign* means not-instrumented, i.e. regular, database-independent C++ classes
- ROOT supports I/O for foreign classes since V3.03/09:
 - no need to derive from TObject
 - no need for ClassDef or ClassImp Macros
 - improved STL support (according to recent FNAL workshop)
- CMS has used (and developed) this feature of ROOT

Foreign Classes: Limitations

- Require a default constructor
- Default constructor must not allocate memory to avoid memory leaks
- Custom streamers are (currently) not supported
- Need a wrapper TObject class for foreign classes
- C++ pointers are problematic in general:
 - requires that objects are read in a specific order
 - requires that objects are stored in the same ROOT file and branch

Foreign Classes: Pointers/References

- An alternative solution to C++ pointers:
TRef and TRefArray:
 - Pros:
 - Reading order does not matter
 - Objects can be in different files/branches
 - delayed reading (on demand, smart pointers)
 - Cons:
 - introduces ROOT dependency

Plans for the Near Future

- Implement or finish implementation of some classes:
TrkCalMiniCompositeR, TrkHelixDataR,
KalMiniTrkR, SvtHotDataR, DchHotDataR,
EmcMiniDigiCompR, etc. ...
- Adapt sequences to support ROOT Mini for BetaMiniUser, event display (?), ...
- Verify ROOT Mini implementation (compare with Objy)
- Get ROOT to work with our SUN platform

Outlook / Discussion

- Should we explore the possibility to make use of Eric's *new Kanga* scheme for the Mini as well?
 - Advantages:
 - More transparent access to the data,
 - Interactive access to the data,
 - ask Eric (-;

The *New Micro* - My 2 Cents

- General remarks:
 - It is important that the new micro satisfies the demands of users who do physics analysis!
 - Therefore it should be developed in close collaboration with AWGs and individual Physicists
 - Produce rapid prototypes, and then get user feedback
 - There should not be a philosophy behind the project, rather do what serves our needs best

The *New Micro* - My 2 Cents

- Identify standard tasks and provide standard tools for them. For example:
 - How to apply the final analysis cuts
 - How to select the “best candidate”
 - How to book and fill histograms
- N-tuple analyses are faster than Beta jobs, but with an ever increasing data set, they are getting slow, too:
 - The new micro should support parallel processing, i.e. the PROOF server !