General Purpose ROOT Utilities

Victor Perevoztchikov, BNL

BROOKHAVEN
NATIONAL LABORATORY

List of Utilities

Set of ROOT classes was developed in STAR experiment. These classes are experiment independent and, we hope, could be used useful in other experiments too. The names of classes are self described:

- TUnixTime;
- TDirIter;
- TMemStat;
- THelixTrack;
- TTreeIter;
- TJobOption.

TUnixTime & TMemStat

- TUnixTime: small simple class, allows simple conversion from/to local unix time to gmt(Greenwich time) and vice versa. User interface is very simple and clear:
  - SetUTime(UInt_t utime)   // Set Unix time;
  - SetLTime(Int_t idate, Int_t itime);   // Set local time:
  - SetGTime(Int_t idate, Int_t itime);   //Set global time:
  - GetLTime(Int_t &idate, Int_t &itime) //Get local time;
  - GetGTime(Int_t &idate, Int_t &itime) //Get global time;

  Internally class contains standard Unix time (time_t).
- TMemStat: memory statistics. Similar to ROOT TBenchmark, but for memory. Useful for leak search. Functionality:
  - Get the current heap memory size;
  - Print size and difference from the previous call;
  - Print summary statistics;
  - Get current program size;

TDirIter

It is iterator over Unix directories. Initialized by user request in wild card or regular expression form. Allows wild card not only for files but for directories too. Example:

```
TDirIter  iter(“/afs/dir1/dir*123/*any/myfile.*.root”);
char *file =0;
While(char *file = iter.NextFile()) {
   printf(“file=%s\n”,file);
 }
```

It is useful when user needs to process a lot of files from different directories with some wild card or regular expression selections for files and for directories too.

ThelixTrack

This class represents helix space line, i.e track in constant magnetic field. GEANT3 representation of helix used, no singularity at curvature~=0. Initialization by 1st space point,moving direction,curvature and field direction. By default field direction along Z. Functionality:

◆Step from the beginning to the given distance along the helix. End space point and direction is returned;

◆Step along the helix to the point nearest to the given one.

◆Step along the helix up to crossing the surface 1st or 2nd order;

◆Creation of helix as a fit to the given set of space points

TTreeIter

TTreeIter is an additional user interface. There are three TTree interfaces in ROOT: GetEntry, Draw and MakeClass:

- GetEntry is very general but low level interface;
- Draw is very convenient but limited to one line expression;
- MakeClass is very general but not too much flexible:
    - Special code must be generated;
    - If TTree is changed, user must regenerate MakeClass code;
    - User must activate used branches manually. Activation of all branches leads to performance degradation;

TTreeIter is the further development of MakeClass approach. But:

- No code is generated;
- TTree changing does not affect the user code;
- Needed branches are activated automatically, like it was in PAW;

TTreeIter Example

```
TTreeIter iter;
iter.AddFile("path1/raw*/minibias*event*.root");
iter.AddFile("path2/mc*/central*event*.root");

// init of user variables
const int         &ntrk  =   iter("fTracks");
const Float_t *&pz    =   iter("fTracks.fPz");


// Now the user variables ntrk and pz are initialized and needed branches
// automatically activated. These variables are referenced to internal
// hidden buffers, which will be automatically updated during iteration.
TH1F *hz = new TH1F("Z","Z distribution",50,-5.,+5.);
//         Now iterations
while (iter.Next())
{
   for (int itr=0;itr<ntrk;itr++) {
     hz->Fill( pz[itr]); }
}
```

TJobOption

Each experiment usually develops own application to define job option for production. In fortran era was often used FFREAD package. Now it is outdated. To make such application is not simple at all. But now, using power of ROOT/CINT, such application could be very simple and very powerful. TJobOption is such class. Example of input job option file:

```
namespace RawData {
  int trigger = 12345; char *mode = “TPC_ONLY”;
}
namespace V0Cuts {
  double deltaMass = 0.01; VCut *cut = new VCut(11,12,13,14);
}
```

The complication of the input language is limited only by CINT;

TJobOption (Continue)

Now how to get values:

```
TJobOption::ReadFile(“MyJobOpt.jopt”);
TJobOption::ReadFile(“HisJobOpt.jopt”);
….
//                    for RawData namespace
TJobOption *raw = new TJobOption(“RawData”);
 int trig       =  raw->GetI(“trigger”,-1)          // 2nd argument is default
 char *mode     =  raw->GetS(“mode”,”ALL”);

//                    for “V0Cuts” namespace
TJobOption v0c (“V0Cuts”);
 double dm     =  v0c.GetD(“deltaMass”,0.1);
VCut *cut      =  (Vcut*) v0c.GetV(“cut”,0);
```

Conclusions

- Five experiment independent and ROOT oriented classes were developed and tested;
- All ROOT users invited to try them. All feedback is very appreciated;
- How to get them?
  - Send E-mail to perev@bnl.gov or
  - Try AFS: /afs/rhic/star/packages/.DEV/StRoot/StarRoot.