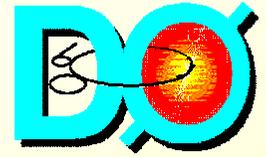


# SEED data



## TTree::MakeClass

```
struct {  
  Int_t JCCG_JCCGnjets;  
  Float_t JCCG_JCCGPx[200];  
  ...  
  Float_t JCCG_JCCGWPphi[200];  
  ...  
}
```

Code ntuple layout dependent  
No object oriented code  
C++ code but FORTRAN data

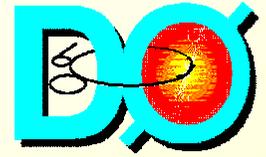
## SEED data objects

```
class TJet:  
  TLorentzVector{  
    Double_t GetWPhi();  
    ...  
  }
```

Inheritance hierarchy allows  
*object oriented, re-usable,  
readable, and ntuple  
independent* analysis code



# MakeClass annoyances



**non structured** data access

**cryptic** data member names

MakeClass generates its class's code automatically, overwriting the old version, so the **code you added is lost** each time you have to regenerate it

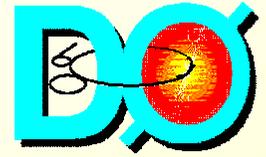
ntuple has an **additional value** that was not there calling MakeClass : Your code is **broken**.

ntuple contains **one branch less** than the ntuple on which MakeClass was called: Your code is **broken**.

With *SEED*: ***Not anymore!***



# *SEED* transformations



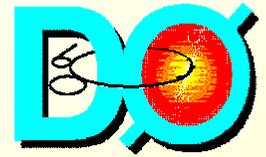
Purpose: transfer ntuple data into objects, store resulting objects in ROOT file

Done once, it decreases the ntuple file size (factor 10-100 is common) and allows access of objects via Root, *SEED*, and your own programs

*SEED* provides complete, easy to use framework for transformation and later data access, separates (and keeps you away from) ROOT layer – so you only have to deal with *SEED* and your data objects



# *SEED* analyses



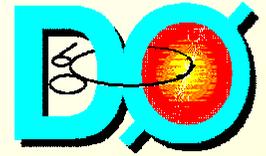
Works with Root both interactively and using compiled programs, *SEED* comes with build environment for transformations and analyses (independent of DØ SRT build system: Fast & easy!)

Using objects your analysis code becomes independent of the data source (MC gen level, online sim, offline sim, real data) as they all fill the same classes

```
Double_t dPhiJet = jet[iJet].Phi();
for (Int_t iPrt = 0; iPrt < prt.Size(); iPrt++) {
    Double_t dPhiTr = prt[iPrt].GetTrack().Phi();
    h1.Fill((dPhiTr - dPhiJet) / kPI);
}
```



# *SEED* availability



**setup seedlib** available on d0mino and clued0

cvs package **seed** and **seed\_framework**, compiles with all compilers on all platforms for which a Root build exist

Or download from the web

Born 2000, many users, fully documented, planned to be integrated into Root: *Trust it!*

Download, documentation, reference:  
<http://www-d0.fnal.gov/nikhef?seed>

