

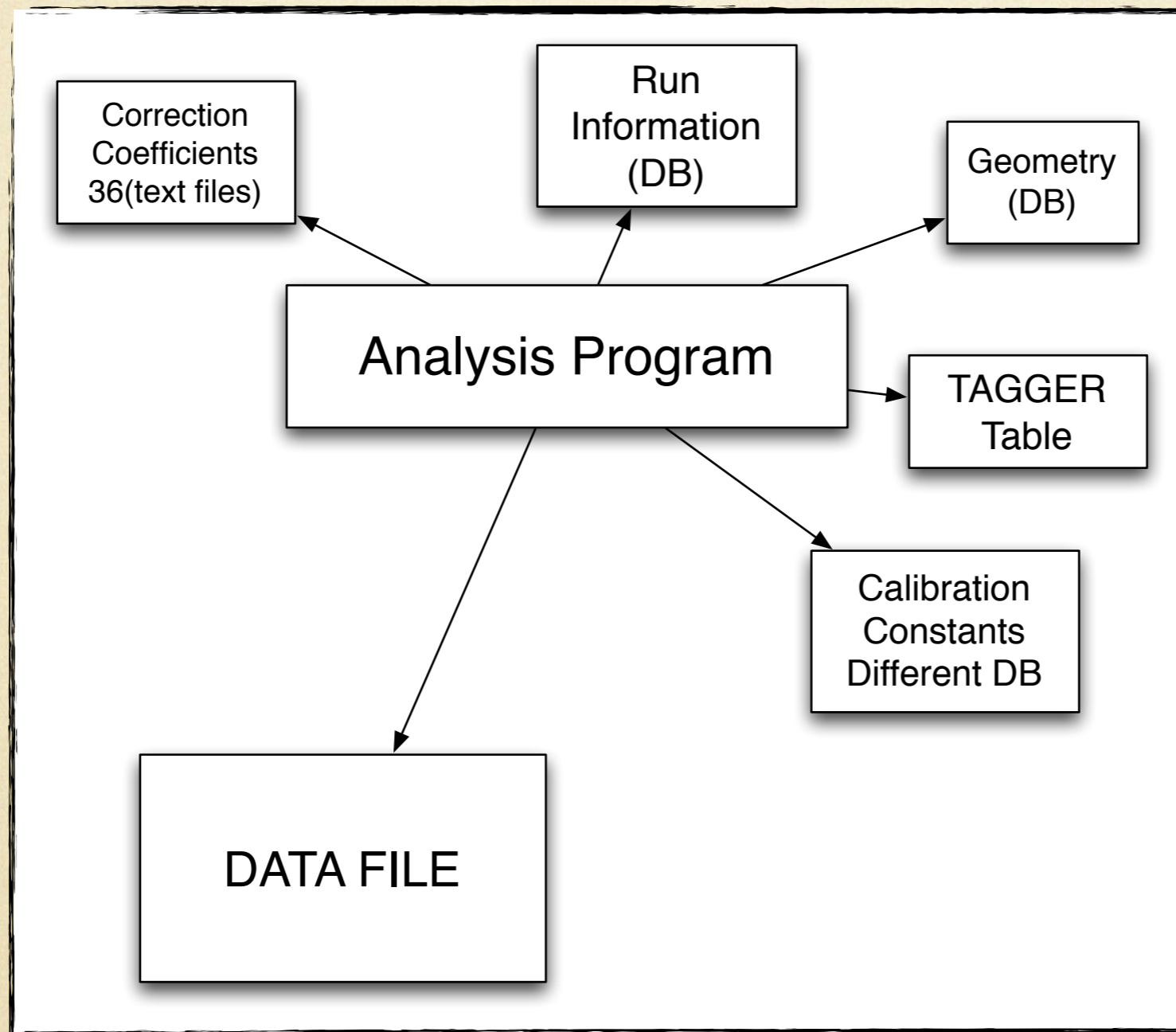
CLAS data format (HDF5)

G.Gavalian (ODU)

Motivation

- For higher level physics analysis more complicated DST structure is needed.
- Current DST structures do not provide tools for storing Physics Analysis Results.
- There are no universal structures defined for the CLAS data (correction codes, etc.).
- For data-mining project we need unified DST structure for all the run periods (acceptances, run information, flux , etc.).

Motivation



DATA FILE

The ~~good~~, the bad, the ugly

BOS

- if DDL file is modified by someone, the existing programs will get affected if new DST is written.
- if DDL file is changed the C++ structures have to be recompiled (bankdefs, clasbanks).
- If new Bank is added to DST one has to get the DDL file to use it.

CLASTOOL (ROOT)

- bank class is modified the CLASTOOL has to be recompiled to use the new version of DST.
- breaks compatibility with older DSTs.
- can not be used without compiled class dictionary.

The ~~good~~, the bad, the ugly

EVIO

- database (mysql) driven bank structure can not be used on the plane (though Airtran has internet)
- changes in the database of the bank structure will render older DSTs useless (unless version history is kept).
- no custom class or dataset can be added to the existing file (in principle it can be done but again, it is structure dependent).

Why another format ?

- Do we really need another data format ?
- Existing formats used at CLAS.

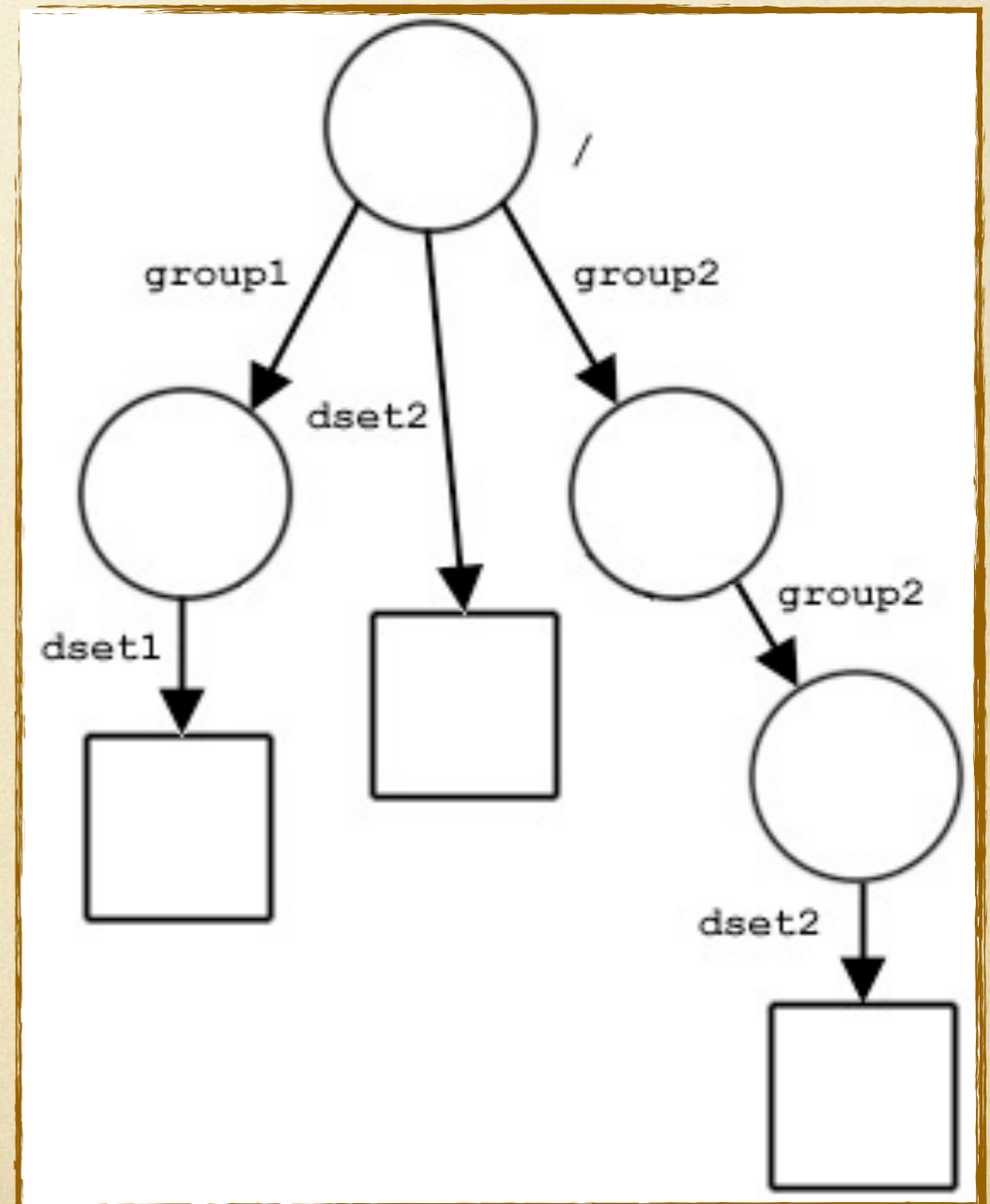
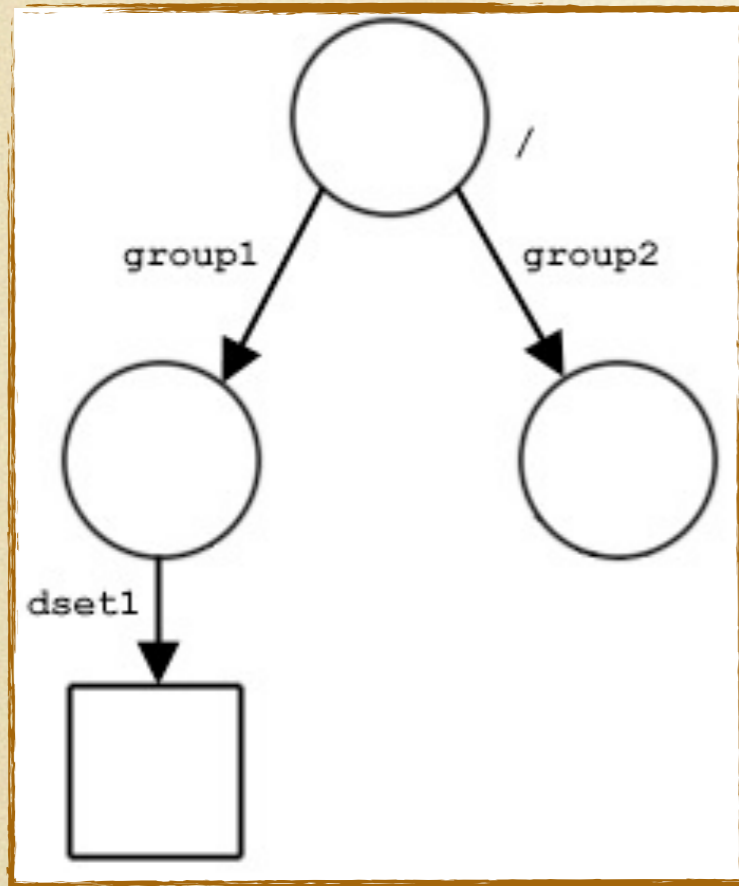
FORMAT	Browser	NS Read	Custom Obj	Versatile
EVIO	NO	N/A	NO	NO
BOS	NO	NO	NO	NO
CLASTOOL	YES	YES	NO	NO
ROOTBEER	N/A	N/A	N/A	N/A
HDF5	YES	YES	YES	YES

Why HDF5 ?

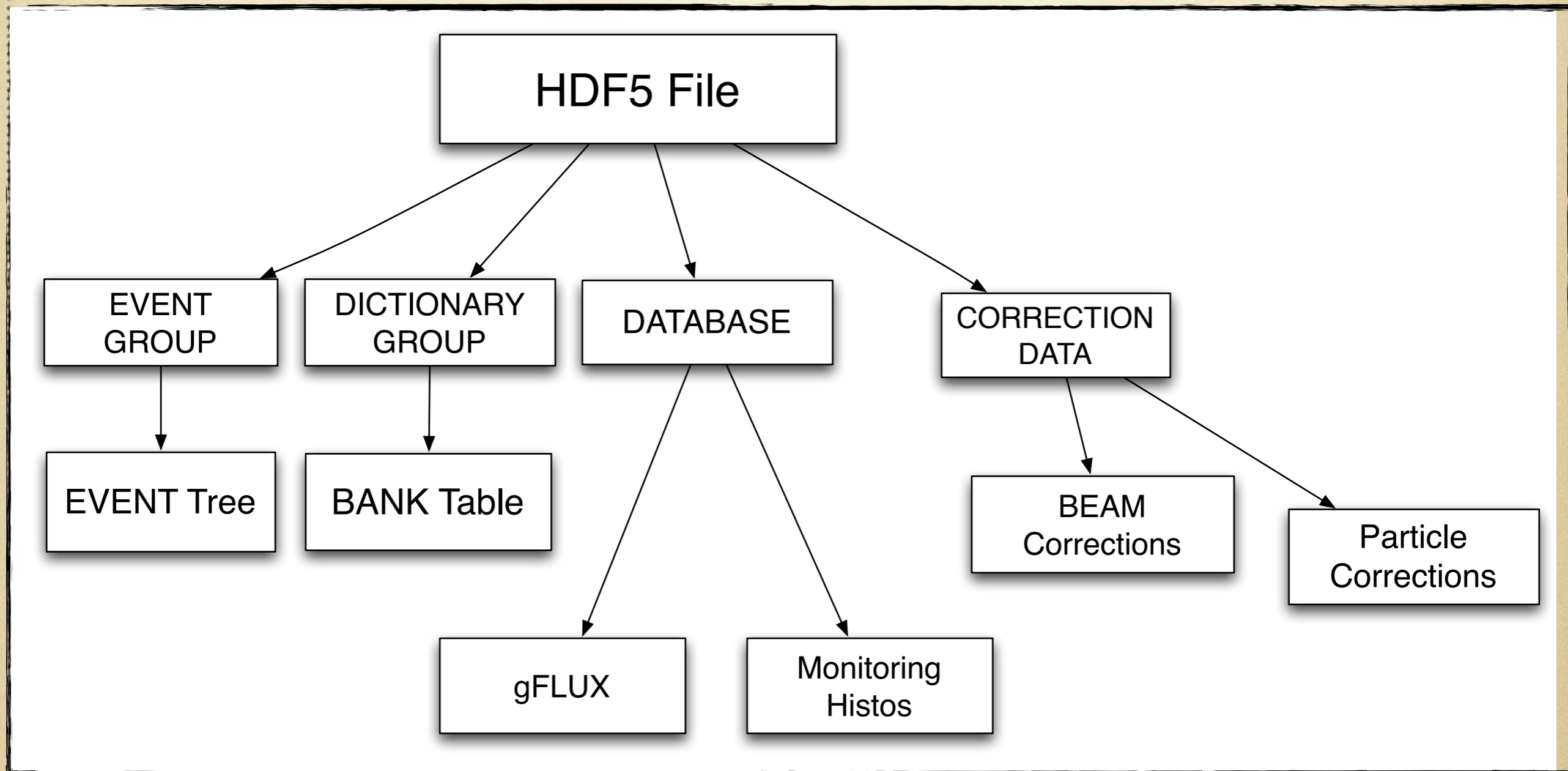
- A versatile data model that can represent complex data and objects.
- A completely portable file format with no limit on the number or size of data objects in the collection.
- A software library that runs on a range of computational platforms, from laptops to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.

Why HDF5 ?

- HDF - Hierarchical Data Format



Data Structure



gFLUX

Monitoring Histos

New Data Format

- Dictionary driven data storage (Tables in HDF)
- Random Access Read / Write .
- Data Compression with Zlib (3 times smaller than BOS).
- Allows Custom objects to be embedded in the file (ex. correction constants, flux constants).
- Highly extendable Bank structure.
- Cross-platform compatibility.

Bank Dictionaries

Reading Particle information from both banks will be the same.

Entry	Type
charge	int
pid	int
px	float
py	float
pz	float
vx	float
vy	float
vz	float

```
Vector mom;  
mom.setX(bank.get("Px"));  
mom.setY(bank.get("Py"));  
mom.setZ(bank.get("Pz"));  
Vector vert;  
vert.setX(bank.get("vx"));  
vert.setY(bank.get("vy"));  
vert.setZ(bank.get("vz"));
```

Entry	Type
vx	float
px	float
charge	int
py	float
vy	float
pid	int
vz	float
pz	float
quality	float
flag	int

Performance

- Big issue of implementation was performance.

benchmarks are done on MacBook (800 MHz bus),
5400 Rpm Hard Drive

Action	Time (sec)
Write (1M events)	188
Read / Analyze (1M events)	41
Read (1M events)	8
Scan (1M events)	7

Bank Creation

- To create a bank one needs to specify the bank variables and types.
- Register with the package **RUN-TIME** (it will be stored in the file as a dictionary table).
- End user has to know of the existence of the bank, and entry name to use it (DST printout provides the information).
- If a program is already using the bank entry, and the bank is modified in the new release, the code does not get affected.
- Easy to pass around DST's generated by different run groups.

The good, the ~~bad~~, the ~~ugly~~

HDF5

- Bank structure stored with the DST provides backward compatibility (no re-compile, no ddl).
- Easy to share custom DST's between the groups.
- Fast file scanning (**done**) for DST viewer (**work in progress**).
- Custom embedded objects in the DST file (**I like this one a lot**).
- A physics analysis framework can be build around it.

HDF5 !

So Easy,
a Caveman could use it !