EVIO Banks

Identified by two integers, a TAG and a NUM
Data is row based, so each variable is a bank with nentries = number of hits

Need Conventions to define Bank structure

Example: SVT “mother” Bank, TAG = 50, NUM 0

Banks of banks

EVIO Banks

```
1   <bank content="bank" tag="50" num="0">
2       <bank content="bank" tag="50" num="100">
3          <int32 tag="50" num="20">
4                2          26
5          </int32>
6          <int32 tag="50" num="21">
7                934        422
8          </int32>
9          <int32 tag="50" num="22">
10               2301       4334
11         </int32>
12      </bank>
21  </bank>
```

The Bank C++ Library

Bank Definitions in DB

Reads Definitions, Converts to Strings

Creates Objects from the EVIO format

Advantage: users can forget about EVIO I/O (trick):
deal with banks/variables names only

The Bank C++ Library

map<string, Mbank> bank;
vector<Mdgt> dgt; STL vector of digitized objects

```
Mevent
   bank["SVT"]
         Mdgt[0]:
              name: ADC
              vector<int> content
         Mdgt[1]:
              name: TDC
              vector<int> content
         [...]
```

The Bank C++ Library

```
Mevent evt(database, hostname, 1);

try
{
     evioFileChannel *chan = new evioFileChannel(file, "r", 3000000);
     chan->open();
     while(chan->read())
     {
          evioDOMTree EDT(chan);
          evt.fill(EDT);

          cout << endl << " >> Event number: " << evt.evn << endl << endl;

          Mbank abank = evt.get_bank(“SVT”);
          Mdgt   adgt = abank.get_dgt_var(“ADC”);
     }
     chan->close();
}
```

HPS Loop

Event Ring

C++ Objects (multithreaded)

Online ROOT Histos, Ntuple

HPS Loop (MultiCPUs)

Event Ring

C++ Objects (multithreaded)

Online ROOT Histos, Ntuple

ROOTSPY: can add the ROOT objects on the fly while program is running on the farm