

Source code

- Test source code available at
`/work/clas12/mesonex/rhipo3.tar.gz`
- Requirements:
 - ROOT version 6,
 - your own copy of the rhipo directory
 - Copy of hipo libcpp source code
 - Currently a copy exists in `/work/clas12/mesonex/libcpp3.tar.gz`
 - ROOT will compile at runtime if required
 - Enviroment set as per `rhipo/SetEnv.csh`
 - `setenv CHIPO /my_path/libcpp`
 - `setenv RHIPPO /my_path/rhipo3`
 - `setenv LD_LIBRARY_PATH "$LD_LIBRARY_PATH":/work/clas12/mesonex/lz4-1.7.5/lib`
 - `setenv LZ4_h /work/clas12/mesonex/lz4-1.7.5/lib`
 - `alias hipo2root root -l $RHIPPO/Hipo2Root.C`
 - `alias allhipo2root root -l $RHIPPO/Hipo2Root.C $RHIPPO/ConvertAll.C`

Example CLAS12 Hipo Bank

- Bank definitions given `$CLARA_HOME/plugins/clas12/etc/bankdefs/hipo/`

- Example bank structure :

Used to get correct bank
From `hipo::event`

```
"bank": "RECHB::Particle",
```

```
  "group": 22,
```

```
  "info": "Reconstructed Particle Information",
```

```
  "items": [
```

Used to get correct item
from `hipo::event` group

```
    {"name": "pid", "id": 1, "type": "int32", "info": "particle id in LUND  
conventions"},
```

```
    {"name": "px", "id": 2, "type": "float", "info": "x component of the  
momentum"},
```

```
    {"name": "py", "id": 3, "type": "float", "info": "y component of the  
momentum"},
```

```
    {"name": "pz", "id": 4, "type": "float", "info": "z component of
```

- For example `Hipo.ConfigBank("RECHB::Particle");` will save all items in this bank to branches (**vectors of ints or floats**) in a ROOT tree.
- For example `Hipo.ConfigBank("REC");` will save all items of all branches that include the string REC

Hipo libcpp (Gagik)

Naive summary :

- Standalone c interface to hipo files
- Based on 3 classes
 - reader
 - Opens file; accesses records
 - record
 - Accesses individual events; handles decompression
 - event
 - Accesses bank data; returns either c++ vector of floats or ints for each group and item as requested
- Requires external liblz4 for reading compressed files
- Does not require additional COATJAVA/CLARA dependencies to run

THipo Brief Summary

- ROOT wrapper for hipo/libcpp
- THipo.h and THipo.C contain 4 classes
- THipoBankParser configures banks from hipo file dictionary (new from hipo.3)
- THipoBank class interfaces between hipo::event and ROOT tree branch
- THipoItem iterates over and returns value for item in event bank
- THipo interacts with hipo::reader and hipo::record and handles event iteration.

hipo2root

#Convert all banks defined in hipo file dictionary

```
allhipo2root file1.hipo file2.hipo ...
```

OR using directories

```
allhipo2root --in=/in/directory --out=/out/directory
```

#Convert only selected banks(groups) or items

#for example via `Hipo.ConfigBank("REC::Particle");`

```
hipo2root $RHIP0/ConvertSomeBanks.C file1.hipo ...
```

OR

```
hipo2root $RHIP0/ConvertSomeBanks.C --in=/in/directory
```

*Users can supply their own configuration macro instead of `$RHIP0/ConvertSomeBanks.C`

Hipo in ROOT

- Access data via THipoItem class
 - Handles the event vector for each item

```
//Get the necessary items from Particle Bank
THipoBank* bank1=Hipo.GetBank("REC::Particle");
THipoItem *px=bank1->GetItem("px");
THipoItem *py=bank1->GetItem("py");
THipoItem *pz=bank1->GetItem("pz");
...
while(Hipo.NextEvent()){ //events
    while(bank1->NextEntry()){ //particles
        part.SetXYZM(px->Val(),py->Val(),pz->Val(),mass->Val());
        vert.SetXYZ(vx->Val(),vy->Val(),vz->Val());
        //Search Rec::Detector bank for associated pindex
        while(d_pindex->FindEntry(bank1->GetEntry())){
            //Do something if find a particular detector
            if(d_det->Val()==1){
                part_time=d_time->Val();
            }
        }
    }
}
```