

# Draft: Write an Analysis Task



Ivonne Maldonado\*

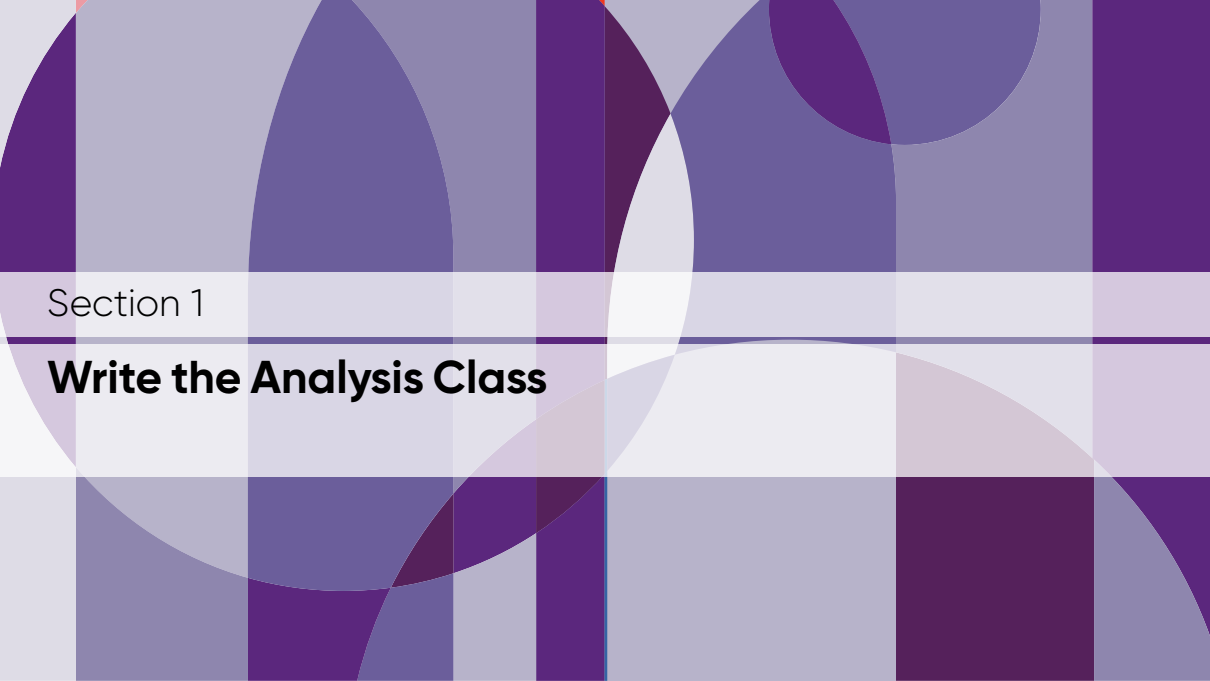
\*VBLHEP, JINR [ivonne.alicia.maldonado@gmail.com](mailto:ivonne.alicia.maldonado@gmail.com)

**MPD Experiment**

February 25<sup>th</sup>, 2022

# Outline

- 1 Write the Analysis Class
- 2 Compile mpdroot with the Analysis Class
- 3 Macro to run the analysis
- 4 Analysis in the cluster



Section 1

# **Write the Analysis Class**

# Introduction

The macros described in this presentation are in the git repository, in the mpdroot branch ana\_class in the folder physics

[https://git.jinr.ru/nica/mpdroot/-/tree/ana\\_class/physics/anaaclass](https://git.jinr.ru/nica/mpdroot/-/tree/ana_class/physics/anaaclass) A document with a more detailed description also is in progress

# Create the class files \*.h and \*.cxx

**The analysis class MpdPtTask is derived from FairTask, and shares the common base methods**

- MpdPtTask::MpdPtTask() – default constructor
- MpdPtTask::MpdPtTask(const char \*name, const char \*title) – constructor with names
- MpdPtTask::~~MpdPtTask() – destructor
- InitStatus MpdPtTask::Init()
- void MpdPtTask::Exec(Option\_t \* option)
- void MpdPtTask::Reset()
- void MpdPtTask::Finish()
- void MpdPtTask::Register()

# InitStatus MpdPtTask::Init()

**In this function you call the branches to be analyzed**

```
FairRootManager *manager = FairRootManager::Instance();  
fMCTracks = (TClonesArray *) manager->GetObject("MCTrack");  
fDstEvent = (MpdEvent *) manager->GetObject("MPDEvent.");  
Register();
```

**And define the output objects of the analysis for example histograms**

```
fhistPt = new TH1F("fhistPt","p_T distribution; p_T(GeV/c); 1/N_evdN/dp_T",400,0,10);  
fhistPtMC = new TH1F("fhistPtMC","MC p_T distribution; p_T(GeV/c);  
1/N_evdN/dp_T",400,0,10);
```

# MpdPtTask::Exec(Option\_t \* option)

## **This function is the event loop**

```
TClonesArray *mpdTracks = fDstEvent->GetGlobalTracks();  
Int_t nTracks = mpdTracks->GetEntriesFast();  
for (Int_t i = 0; i < nTracks; i++){  
    MpdTrack *track = (MpdTrack *) mpdTracks->UncheckedAt(i);  
    fhistpt->Fill(track->GetPt());  
}
```


# MpdPtTask::Finish()

## Store the output histograms

```
fhistPt->Write("");
```

```
fhistPtMC->Write("");
```





Section 2

## **Compile mpdroot with the Analysis Class**

# Modify CMakeList.txt to add the folder with your task

In the folder physics ***mpdroot/physics*** to compile the **MpdPtTask** located in the ***mpdroot/physics/anaclass*** folder you need to add in the **CMakeList.txt** file the line

```
add_subdirectory(anaclass)
```

```
add_subdirectory(femto/fsiTools)
add_subdirectory(femto/MpdFemtoMaker)
add_subdirectory(femto/MpdFemtoMakerUser)
add_subdirectory(photons)
add_subdirectory(anaclass)

# Create a library called "MpdPhysics" which includes the
# The extension is already found. Any number of sources can be added
set(INCLUDE_DIRECTORIES
  ${BASE_INCLUDE_DIRECTORIES}
  ${CMAKE_SOURCE_DIR}/mpdbase
  ${CMAKE_SOURCE_DIR}/mcstack
  ${CMAKE_SOURCE_DIR}/kalman
  ${CMAKE_SOURCE_DIR}/physics
)

set(SYSTEM_INCLUDE_DIRECTORIES
  ${ROOT_INCLUDE_DIR}
)

include_directories(${INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories(${LINK_DIRECTORIES})
```

# Folder with your class

The folder `anaclass` should contain the files:

- `CMakeList.txt`
- `AnaClassLinkDef.h`
- Your header class → `MpdPtTask.h`
- And its implementation file → `MpdPtTask.cxx`

# Content of CMakeList.txt in anaclass folder

```
[maldonado@ncx103 anaclass]$ more CMakeLists.txt
set(INCLUDE_DIRECTORIES
  ${BASE_INCLUDE_DIRECTORIES}
  ${CMAKE_SOURCE_DIR}/mpdbase
  ${CMAKE_SOURCE_DIR}/kalman
  ${CMAKE_SOURCE_DIR}/mcstack
  ${CMAKE_SOURCE_DIR}/physics
  ${CMAKE_SOURCE_DIR}/physics/anaclass
)

Set(SYSTEM_INCLUDE_DIRECTORIES
  ${ROOT_INCLUDE_DIR}
)

include_directories(${INCLUDE_DIRECTORIES})
Include_Directories(SYSTEM ${SYSTEM_INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories(${LINK_DIRECTORIES})

set(SRCS
  MpdPtTask.cxx
)

Set(HEADERS)
Set(LINKDEF AnaClassLinkDef.h)
Set(LIBRARY_NAME MpdAnaClass)
Set(DEPENDENCIES)

GENERATE_LIBRARY()
```

The structure is similar to the other CMakeList.txt files

# 1. Content of CMakeList.txt in anaclass folder

```
[maldonado@ncx103 anaclass]$ more CMakeLists.txt
set(INCLUDE_DIRECTORIES
  ${BASE_INCLUDE_DIRECTORIES}
  ${CMAKE_SOURCE_DIR}/mpdbase
  ${CMAKE_SOURCE_DIR}/kalman
  ${CMAKE_SOURCE_DIR}/mcstack
  ${CMAKE_SOURCE_DIR}/physics
  ${CMAKE_SOURCE_DIR}/physics/anaclass
)

Set(SYSTEM_INCLUDE_DIRECTORIES
  ${ROOT_INCLUDE_DIR}
)

include_directories(${INCLUDE_DIRECTORIES})
Include_Directories(SYSTEM ${SYSTEM_INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories(${LINK_DIRECTORIES})

set(SRCS
  MpdPtTask.cxx
)

Set(HEADERS)
Set(LINKDEF AnaClassLinkDef.h)
Set(LIBRARY_NAME MpdAnaClass)
Set(DEPENDENCIES)

GENERATE_LIBRARY()
```

Include all the folders with classes required for your analysis and the folder containing your task

## 2. Content of CMakeList.txt in anaclass folder

```
[maldonado@ncx103 anaclass]$ more CMakeLists.txt
set(INCLUDE_DIRECTORIES
  ${BASE_INCLUDE_DIRECTORIES}
  ${CMAKE_SOURCE_DIR}/mpdbase
  ${CMAKE_SOURCE_DIR}/kalman
  ${CMAKE_SOURCE_DIR}/mcstack
  ${CMAKE_SOURCE_DIR}/physics
  ${CMAKE_SOURCE_DIR}/physics/anaclass
)

Set(SYSTEM_INCLUDE_DIRECTORIES
  ${ROOT_INCLUDE_DIR}
)

include_directories(${INCLUDE_DIRECTORIES})
Include_Directories(SYSTEM ${SYSTEM_INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories(${LINK_DIRECTORIES})

set(SRCS
  MpdPtTask.cxx
)

Set(HEADERS)
Set(LINKDEF AnaClassLinkDef.h)
Set(LIBRARY_NAME MpdAnaClass)
Set(DEPENDENCIES)

GENERATE_LIBRARY()
[maldonado@ncx103 anaclass]$
```

Add the name of your class, in this case MpdPtTask.cxx

### 3. Content of CMakeList.txt in anaclass folder

```
[maldonado@ncx103 anaclass]$ more CMakeLists.txt
set(INCLUDE_DIRECTORIES
  ${BASE_INCLUDE_DIRECTORIES}
  ${CMAKE_SOURCE_DIR}/mpdbase
  ${CMAKE_SOURCE_DIR}/kalman
  ${CMAKE_SOURCE_DIR}/mcstack
  ${CMAKE_SOURCE_DIR}/physics
  ${CMAKE_SOURCE_DIR}/physics/anaclass
)

set(SYSTEM_INCLUDE_DIRECTORIES
  ${ROOT_INCLUDE_DIR}
)

include_directories(${INCLUDE_DIRECTORIES})
include_directories(SYSTEM ${SYSTEM_INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories(${LINK_DIRECTORIES})

set(SRCS
  MpdPtTask.cxx
)

set(HEADERS)
set(LINKDEF AnaClassLinkDef.h)
set(LIBRARY_NAME MpdAnaClass)
set(DEPENDENCIES)

generate_library()
```

Write the name of your LinkDef.h file  
and the name of the library


# The LinkDef.h file

The LinkDef.h file tells which classes should be added to the dictionary.

```
#ifndef __CINT__  
  
#pragma link off all globals;  
#pragma link off all classes;  
#pragma link off all functions;  
  
#pragma link C++ class MpdPtTask+;  
  
#endif
```

We add the class MpdPtTask





## Section 3

# **Macro to run the analysis**

# Run Analysis: runana.C

```
void runana(TString inFile = "mpdst.root", TString outFile = "outputana.root", Int_t nStartEvent = 0,
Int_t nEvents = 500){
// ----- Timer -----
TStopwatch timer;
timer.Start();

FairRunAna* fRun;
if (!CheckFileExist(inFile)) return;
fRun = new FairRunAna();

FairSource* fFileSource = new FairFileSource(inFile);
fRun->SetSource(fFileSource);
fRun->SetOutputFile(outFile); // [WARN] FairRun::SetOutputFile() deprecated. Use FairRootFileSink.
fRun->SetGenerateRunInfo(false);
fRun->SetUseFairLinks(true);

TString parFile = inFile;


FairTask* physics = new MpdPtTask("MpdPhysicsTask", "task");
fRun->AddTask(physics);

fRun->Init();
fRun->Run(nStartEvent, nStartEvent + nEvents);

// ----- Finish -----
timer.Stop();
Double_t rtime = timer.RealTime();
Double_t ctime = timer.CpuTime();
cout << endl << endl;
cout << "Macro finished successfully." << endl; // marker of successful execution for CD
ASH
cout << "Output file is " << outFile << endl;
cout << "Parameter file is " << parFile << endl;
cout << "Real time " << rtime << " s, CPU time " << ctime << " s" << endl;
cout << endl;
}
-- INSERT -- 25,73 Bot
```

Call the class FairRunAna allows data analysis, and call the file to be readed with FairSource.

Call your Task with FairTask and init the analysis



Section 4

## **Analysis in the cluster**

# Processing several files

To run in the nica cluster, use the the macro runanalysis.C together with files many\_jobs.sh, analysis.sh and mergefiles.C

- change the path of your mpddst.root file in the macro runanalysis.C
- change the name of your file in macro many\_jobs.sh
- change the paths to your own folders and to your mpdroot installation in macro analysis.sh
- run the analysis with:

```
source many_jobs.sh
```

- Wait for the jobs
- Change the path to your output file in the macro mergefiles.C to get only one file

Presentation with more details is in preparation