New macro style with “master” objects

Stefano Spataro

UNIVERSITÀ DEGLI STUDI DI TORINO
ALMA UNIVERSITAS TAURINENSIS

ISTITUTO NAZIONALE DI FISICA NUCLEARE
Sezione di Torino

Tuesday, 7th June 2016

The Goal

Single centralized management of “default” settings

- ✓ Geometry definitions
- ✓ Task list
- ✓ Parameter settings
- ✓ Flags, name style, field settings, diagnostic, …

Problems which are going to be solved:

- ✧ A lot of qa macros to modify each time something changes
- ✧ Developers/users never updating their macros, using obsolete settings (and realizing when too late)

- ➢ Backward compatible
- ➢ Easy to implement “dedicated” settings for developers

Standard simulation macro

```
sim_complete(Int_t nEvents = 100, TString  SimEngine ="TGeant3", Float_t mom = 6.231552)
{
  //-----User Settings:-----------------------------------------------
  TString  OutputFile     ="sim_complete.root";
  TString  ParOutputfile  ="simparams.root";
  TString  MediaFile      ="media_pnd.geo";
  gDebug                  = 0;
  TString digiFile        = "all.par"; //The emc run the hit producer directly
                            // choose your event generator
  Bool_t UseEvtGenDirect      =kTRUE;
  Bool_t UseDpm                =kFALSE;
  Bool_t UseFtf                =kFALSE;
  Bool_t UseBoxGenerator      =kFALSE;

  Double_t BeamMomentum = 0.; // beam momentum ONLY for the scaling of the dipole field.
  if (UseBoxGenerator)
  {
    BeamMomentum   =15.0; // ** change HERE if you run Box generator
  }
  else
  {
    BeamMomentum = mom;  // for DPM/EvtGen BeamMomentum is always = mom
  }
  //-----------------------------------------------------------------
  TLorentzVector flni(0, 0, mom, sqrt(mom*mom+9.3827203e-01*9.3827203e-01)+9.3827203e-01);
  TDatabasePDG::Instance()->AddParticle("pbarpSystem","pbarpSystem",flni.M(),kFALSE,0.1,0, "",88888);
  //-----------------------------------------------------------------
  TStopwatch timer;
  timer.Start();
  gRandom->SetSeed();

  // Create the Simulation run manager----------------------------------
  FairRunSim *fRun = new FairRunSim();
  fRun->SetName(SimEngine.Data() );
  fRun->SetOutputFile(OutputFile.Data());
  fRun->SetGenerateRunInfo(kFALSE);
  fRun->SetBeamMom(BeamMomentum);
  fRun->SetMaterials(MediaFile.Data());
  fRun->SetUseFairLinks(kTRUE);
  FairRuntimeDb *rtdb=fRun->GetRuntimeDb();

  // Set the parameters
  //--------------------------------
  TString allDigiFile = gSystem->Getenv("VMCWORKDIR");
  allDigiFile += "/macro/params/";
  allDigiFile += digiFile;


  //-------Set the parameter output --------------------
  FairParAsciiFileIo* parIo1 = new FairParAsciiFileIo();
  parIo1->open(allDigiFile.Data(),"in");
  rtdb->setFirstInput(parIo1);

  //---------------------Set Parameter output      ----------
  Bool_t kParameterMerged=kTRUE;
  FairParRootFileIo* output=new FairParRootFileIo(kParameterMerged);
  output->open(ParOutputfile.Data());
  rtdb->setOutput(output);

  // Create and add detectors

  //-------------------------  CAVE      ----------------
  FairModule *Cave= new PndCave("CAVE");
  Cave->SetGeometryFileName("pndcave.geo");
  fRun->AddModule(Cave);
  //-------------------------  Magnet   ----------------
  //FairModule *Magnet= new PndMagnet("MAGNET");
  //Magnet->SetGeometryFileName("FullSolenoid_V842.root");
  //Magnet->SetGeometryFileName("FullSuperconductingSolenoid_v831.root");
  //fRun->AddModule(Magnet);
  FairModule *Dipole= new PndMagnet("MAGNET");
  Dipole->SetGeometryFileName("dipole.geo");
  fRun->AddModule(Dipole);
  //-------------------------  Pipe     ----------------
  FairModule *Pipe= new PndPipe("PIPE");
  Pipe->SetGeometryFileName("beampipe_201309.root");
  fRun->AddModule(Pipe);
  //-------------------------  STT      ----------------
  FairDetector *Stt= new PndStt("STT", kTRUE);
  Stt->SetGeometryFileName("straws_skewed_blocks_35cm_pipe.geo");
  fRun->AddModule(Stt);
  //-------------------------  MVD      ----------------
  FairDetector *Mvd = new PndMvdDetector("MVD", kTRUE);
  Mvd->SetGeometryFileName("Mvd-2.1_FullVersion.root");
  fRun->AddModule(Mvd);
  //-------------------------  GEM      ----------------
  FairDetector *Gem = new PndGemDetector("GEM", kTRUE);
  Gem->SetGeometryFileName("gem_3Stations_Tube.root");
  fRun->AddModule(Gem);
  //-------------------------  EMC      ----------------
  PndEmc *Emc = new PndEmc("EMC",kTRUE);
  Emc->SetGeometryVersion(1);
  Emc->SetStorageOfData(kFALSE);
  fRun->AddModule(Emc);
  //-------------------------  SCITIL   ----------------
  FairDetector *SciT = new PndSciT("SCIT",kTRUE);
  SciT->SetGeometryFileName("SciTil_201504.root");
  fRun->AddModule(SciT);
  //-------------------------  DRC      ----------------
  PndDrc *Drc = new PndDrc("DIRC", kTRUE);
  Drc->SetGeometryFileName("dirc_l0_p0_updated.root");
  Drc->SetRunCherenkov(kFALSE);
  fRun->AddModule(Drc);
  //-------------------------  DISC     ----------------
  PndDsk* Dsk = new PndDsk("DSK", kTRUE);
  Dsk->SetStoreCerenkovs(kFALSE);
  Dsk->SetStoreTrackPoints(kFALSE);
  fRun->AddModule(Dsk);
  //-------------------------  MDT      ----------------
  PndMdt *Muo = new PndMdt("MDT",kTRUE);
  Muo->SetBarrel("fast");
  Muo->SetEndcap("fast");
  Muo->SetMuonFilter("fast");
  Muo->SetForward("fast");
  Muo->SetMdtMagnet(kTRUE);
  Muo->SetMdtCoil(kTRUE);
  Muo->SetMdtMFIron(kTRUE);
  fRun->AddModule(Muo);
  //-------------------------  FTS      ----------------
  FairDetector *Fts= new PndFts("FTS", kTRUE);
  Fts->SetGeometryFileName("fts.geo");
  fRun->AddModule(Fts);
  //-------------------------  FTOF     ----------------
  FairDetector *FTof = new PndFtof("FTOF",kTRUE);
  FTof->SetGeometryFileName("ftofwall.root");
  fRun->AddModule(FTof);
  //-------------------------  RICH     ----------------
  FairDetector *Rich= new PndRich("RICH",kFALSE);
  Rich->SetGeometryFileName("rich_v2_shift.geo");
  fRun->AddModule(Rich);

  //Create and Set Event Generator
  //-------------------------------
  FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
  fRun->SetGenerator(primGen);

  if(UseBoxGenerator){      // Box Generator
    FairBoxGenerator* boxGen = new FairBoxGenerator(22, 5); // 13 = muon; 1 = multipl.
    boxGen->SetPRange(mom,mom); // GeV/c
    boxGen->SetPhiRange(0., 360.); // Azimuth angle range [degree]
    boxGen->SetThetaRange(0., 90.); // Polar angle in lab system range [degree]
    boxGen->SetXYZ(0., 0., 0.); // cm
    primGen->AddGenerator(boxGen);
  }
  if(UseDpm){
    PndDpmDirect *Dpm= new PndDpmDirect(mom,1);
    primGen->AddGenerator(Dpm);
  }
  if(UseFtf){
    //        TString macfile = gSystem->Getenv("VMCWORKDIR");
    //                          macfile += "/pgenerators/FtfEvtGen/PbarP.mac";
    //                          PndFtfDirect *Ftf = new PndFtfDirect(macfile.Data());
    PndFtfDirect *Ftf = new PndFtfDirect("anti_proton", "G4_H", 1, "ftfp", mom, 123456);
    primGen->AddGenerator(Ftf);
  }
  if(UseEvtGenDirect){
    TString  EvtInput =gSystem->Getenv("VMCWORKDIR");
    EvtInput+="/macro/run/psi2s_Jpsi2pi_Jpsi_mumu.dec";
    PndEvtGenDirect *EvtGen = new PndEvtGenDirect("pbarpSystem", EvtInput.Data(), mom);
    EvtGen->SetStoreTree(kTRUE);
    primGen->AddGenerator(EvtGen);
  }

  //---------------------Create and Set the Field(s)----------
  PndMultiField *fField= new PndMultiField("AUTO");
  fRun->SetField(fField);

  // EMC Hit producer
  //-----------------------------
  PndEmcHitProducer* emcHitProd = new PndEmcHitProducer();
  fRun->AddTask(emcHitProd);

  //-------------------------  Initialize the RUN  ----------------
  fRun->Init();
  //-------------------------  Run the Simulation  ----------------
  fRun->Run(nEvents);
  //-------------------------  Save the parameters ----------------
  rtdb->saveOutput();
  //------------------------Print some info and exit---------------
  timer.Stop();
  Double_t rtime = timer.RealTime();
  Double_t ctime = timer.CpuTime();
  printf("RealTime=%f seconds, CpuTime=%f seconds\n",rtime,ctime);

  cout << " Test passed" << endl;
  cout << " All ok " << endl;

  //exit(0);
```

199 code lines

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/run/sim_complete.C

Standard digitization macro

```
void digi_complete()
{
  // Macro created 20/09/2006 by S.Spataro
  // It loads a simulation file and digitize hits

  // Verbosity level (0=quiet, 1=event level, 2=track level, 3=debug)
  Int_t iVerbose = 0; // just forget about it, for the moment

  // Input file (MC events)
  TString inFile = "sim_complete.root";

  // Parameter file
  TString parFile = "simparams.root"; // at the moment you do not need it

  // Digitisation file (ascii)
  TString digiFile = "all.par";

  // Output file
  TString outFile = "digi_complete.root";

  // -----   Timer   --------------------------------------------------
  TStopwatch timer;

  // -----   Reconstruction run   -------------------------------------
  FairRunAna *fRun= new FairRunAna();
  fRun->SetInputFile(inFile);
  fRun->SetOutputFile(outFile);
  fRun->SetGenerateRunInfo(kFALSE);
  fRun->SetUseFairLinks(kTRUE);
  // -----  Parameter database   --------------------------------------
  TString allDigiFile = gSystem->Getenv("VMCWORKDIR");
  allDigiFile += "/macro/params/";
  allDigiFile += digiFile;

  FairRuntimeDb* rtdb = fRun->GetRuntimeDb();
  FairParRootFileIo* parInput1 = new FairParRootFileIo();
  parInput1->open(parFile.Data());

  FairParAsciiFileIo* parIo1 = new FairParAsciiFileIo();
  parIo1->open(allDigiFile.Data(),"in");

  rtdb->setFirstInput(parInput1);
  rtdb->setSecondInput(parIo1);

  // -----   STT digi producers   --------------------------------
  PndSttHitProducerRealFast* sttHitProducer = new PndSttHitProducerRealFast();
  fRun->AddTask(sttHitProducer);

  // -----   MDV digi producers   --------------------------------
  PndMvdDigiTask* mvddigi = new PndMvdDigiTask();
  mvddigi->SetVerbose(iVerbose);
  fRun->AddTask(mvddigi);

  PndMvdClusterTask* mvdmccls = new PndMvdClusterTask();
  mvdmccls->SetVerbose(iVerbose);
  fRun->AddTask(mvdmccls);

  // -----   EMC hit producers   --------------------------------
  PndEmcHitsToWaveform* emcHitsToWaveform= new PndEmcHitsToWaveform(iVerbose);
  PndEmcWaveformToDigi* emcWaveformToDigi=new PndEmcWaveformToDigi(iVerbose);
  emcHitsToWaveform->SetStorageOfData(kFALSE);
  //emcWaveformToDigi->SetStorageOfData(kFALSE);
  fRun->AddTask(emcHitsToWaveform);  // full digitization
  fRun->AddTask(emcWaveformToDigi);  // full digitization

  PndEmcMakeCluster* emcMakeCluster= new PndEmcMakeCluster(iVerbose);
  fRun->AddTask(emcMakeCluster);

  PndEmcMakeBump* emcMakeBump= new PndEmcMakeBump();
  fRun->AddTask(emcMakeBump);

  //PndEmcHdrFiller* emcHdrFiller = new PndEmcHdrFiller();
  //fRun->AddTask(emcHdrFiller); // ECM header

  // -----   SciT hit producers   ------------------------
  PndSciTHitProducerIdeal* tofhit = new PndSciTHitProducerIdeal();
  tofhit->SetVerbose(iVerbose);
  fRun->AddTask(tofhit);

  // -----   MDT hit producers   --------------------------------
  PndMdtHitProducerIdeal* mdtHitProd = new PndMdtHitProducerIdeal();
  mdtHitProd->SetPositionSmearing(.3); // position smearing [cm]
  fRun->AddTask(mdtHitProd);

  PndMdtTrkProducer* mdtTrkProd = new PndMdtTrkProducer();
  fRun->AddTask(mdtTrkProd);

  // -----   DRC hit producers   --------------------------------
  PndDrcHitProducerReal* drchit = new PndDrcHitProducerReal();
  drchit->SetVerbose(iVerbose);
  fRun->AddTask(drchit);

  // -----   GEM hit producers   --------------------------------
  Int_t verboseLevel = 0;
  PndGemDigitize* gemDigitize = new PndGemDigitize("GEM Digitizer", verboseLevel);
  fRun->AddTask(gemDigitize);
  PndGemFindHits* gemFindHits = new PndGemFindHits("GEM Hit
  fRun->AddTask(gemFindHits);

  // -----   FTS hit producers   --------------------------------
  PndFtsHitProducerRealFast* ftsHitProducer = new PndFtsHitProdu
  fRun->AddTask(ftsHitProducer);

  // -----   Ftof hit producers   -------------------------
  PndFtofHitProducerIdeal* ftofhit = new PndFtofHitProducerIdeal();
  ftofhit->SetVerbose(iVerbose);
  fRun->AddTask(ftofhit);
  // -----   Intialise and run   --------------------------------------------
  fRun->Init();

  timer.Start();
  fRun->Run();
  // -----   Finish   -------------------------------------------------------
  timer.Stop();
  Double_t rtime = timer.RealTime();
  Double_t ctime = timer.CpuTime();
  cout << endl << endl;
  cout << "Macro finished successfully." << endl;
  cout << "Output file is "    << outFile << endl;
  cout << "Parameter file is " << parFile << endl;
  cout << "Real time " << rtime << " s, CPU time " << ctime << " s" <
  cout << endl;
  // ------------------------------------------------------------------------
  cout << " Test passed" << endl;
  cout << " All ok " << endl;
                                                                    exit(0);
}
```

131 code lines

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/run/digi_complete.C

Standard macros

Simulation macro/run/sim_complete.C 199 code lines	Changes in geometry Need to set EvtGen params, DPM flags, box parameters, proper field
Digitization macro/run/digi_complete.C 131 code lines	Almost untouched Very rare changes in digitization tasks
Reconstruction (tracking) macro/run/reco_complete.C 143 code lines	Sometimes changes in the reconstruction flags
Reconstruction (PID) macro/run/pid_complete.C 115 code lines	Sometimes new PID algorithm is added

The Idea

FairRunSim
FairRunAna

PndMasterRunSim : public FairRunSim
PndMasterRunAna : public FairRunAna

They derive all the FairRun functionalities

AND

They add new functionalities

- ✓ standard filename style definition
- ✓ handling of common settings
- ✓ creation of standard geometry volumes
- ✓ inclusion of standard tasks
- ✓ finalize and provide diagnostics for dashboard

New simulation macro

```
sim_complete(Int_t nEvents = 100, TString  SimEngine ="TGeant3", Double_t BeamMomentum = 6.231552)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  // TString inputGenerator =
  // EvtGen -> "xxxxxxxx.dec"
  // DPM    -> "dpm_xxxxx"
  // FTF    -> "ftf_xxxxx"
  TString  inputGenerator = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  //----------------------------------------------------------------------
  // -----   Create the Simulation run manager ----------------------------
  PndMasterRunSim *fRun = new PndMasterRunSim();
  fRun->SetInput(inputGenerator);
  fRun->SetName(SimEngine);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->SetNumberOfEvents(nEvents);
  fRun->SetBeamMom(BeamMomentum);
  // -----   Initialization   ---------------------------------------------
  fRun->Setup();
  // -----   Geometry   ---------------------------------------------------
  fRun->CreateGeometry();
  // -----   Event generator   --------------------------------------------
  fRun->SetGenerator();
  // -----   Add tasks   --------------------------------------------------
  fRun->AddSimTasks();
  // -----   Intialise and run   ------------------------------------------
  fRun->Init();
  fRun->Run(nEvents);
  fRun->Finish();
};
```

User settings

I-do-whatever-you-need functions

39 code lines

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/sim_complete.C

PndMasterRunSim:Setup()

```
// -----   Setup   ----------------------------------------------------------
Bool_t PndMasterRunSim::Setup()
{

  TString inputName = fInput;
  inputName.ToLower();
  if (inputName.EndsWith(".dec")) inputName.Remove(inputName.Length()-4,4);

  PndFileNameCreator creator(inputName.Data());
  SetOutputFile(creator.GetSimFileName().data());
  fOutFile = creator.GetSimFileName().data();
  SetParamRootFile(creator.GetParFileName().data());
  SetMaterials("media_pnd.geo");
  SetGenerateRunInfo(kFALSE);
  SetUseFairLinks(kTRUE);

  // -----  Parameter database   --------------------------------------------
  TString allDigiFile = gSystem->Getenv("VMCWORKDIR");
  allDigiFile += "/macro/params/";
  allDigiFile += fParamAsciiFile;

  fRtdb = this->GetRuntimeDb();
  Bool_t kParameterMerged=kFALSE; // No use until now
  FairParRootFileIo* parOutput = new FairParRootFileIo(kParameterMerged);
  parOutput->open(fParamRootFile,"RECREATE");

  FairParAsciiFileIo* parIo1 = new FairParAsciiFileIo();
  parIo1->open(allDigiFile.Data(),"in");

  fRtdb->setFirstInput(parIo1);
  fRtdb->setOutput(parOutput);

  // -----  Create and Set the Field(s) ---------------------------------------
  PndMultiField *field= new PndMultiField("AUTO");
  SetField(field);

  // ---- Defining PANDA particles ------------------------------------------
  Double_t mom = GetBeamMom();
  TLorentzVector fIni(0, 0, mom, sqrt(mom*mom+9.3827203e-01*9.3827203e-01)+9.3827203e-01);
  TDatabasePDG::Instance()->AddParticle("pbarpSystem" ,"pbarpSystem",  fIni.M(), kFALSE, 0.1, 0, "", 88888);
  TDatabasePDG::Instance()->AddParticle("pbarpSystem0","pbarpSystem0", fIni.M(), kFALSE, 0.1, 0, "", 88880);
  TDatabasePDG::Instance()->AddParticle("pbarpSystem1","pbarpSystem1", fIni.M(), kFALSE, 0.1, 0, "", 88881);
  TDatabasePDG::Instance()->AddParticle("pbarpSystem2","pbarpSystem2", fIni.M(), kFALSE, 0.1, 0, "", 88882);

  return kTRUE;
}
```

It does all the boring things
which you don't need to take care

Yes I know
pbard is missing (still)

PndMasterRunSim:CreateGeometry()

```
void PndMasterRunSim::CreateGeometry()
{
  //------------------------  CAVE      -----------------
  FairModule *Cave= new PndCave("CAVE");
  Cave->SetGeometryFileName("pndcave.geo");
  AddModule(Cave);
  //------------------------  Magnet   -----------------
  //FairModule *Magnet= new PndMagnet("MAGNET");
  //Magnet->SetGeometryFileName("FullSolenoid_V842.root");
  //Magnet->SetGeometryFileName("FullSuperconductingSolenoid_v831.root");
  //AddModule(Magnet);
  FairModule *Dipole= new PndMagnet("MAGNET");
  Dipole->SetGeometryFileName("dipole.geo");
  AddModule(Dipole);
  //------------------------  Pipe     -----------------
  FairModule *Pipe= new PndPipe("PIPE");
  Pipe->SetGeometryFileName("beampipe_201309.root");
  AddModule(Pipe);
  //------------------------  STT       -----------------
  FairDetector *Stt= new PndStt("STT", kTRUE);
  Stt->SetGeometryFileName("straws_skewed_blocks_35cm_pipe.geo");
  AddModule(Stt);
  //------------------------  MVD       -----------------
  FairDetector *Mvd = new PndMvdDetector("MVD", kTRUE);
  Mvd->SetGeometryFileName("Mvd-2.1_FullVersion.root");
  AddModule(Mvd);
  //------------------------  GEM       -----------------
  FairDetector *Gem = new PndGemDetector("GEM", kTRUE);
  Gem->SetGeometryFileName("gem_3Stations_Tube.root");
  AddModule(Gem);
  //------------------------  EMC       -----------------
  PndEmc *Emc = new PndEmc("EMC",kTRUE);
  Emc->SetGeometryVersion(1);
  Emc->SetStorageOfData(kFALSE);
  AddModule(Emc);
  //------------------------  SCITIL    -----------------
  FairDetector *SciT = new PndSciT("SCIT",kTRUE);
  SciT->SetGeometryFileName("SciTil_201504.root");
  AddModule(SciT);
  //------------------------  DRC       -----------------
  PndDrc *Drc = new PndDrc("DIRC", kTRUE);
  Drc->SetGeometryFileName("dirc_l0_p0_updated.root");
  Drc->SetRunCherenkov(kFALSE);
  AddModule(Drc);
  //------------------------  DISC      -----------------
  PndDsk* Dsk = new PndDsk("DSK", kTRUE);
  Dsk->SetStoreCerenkovs(kFALSE);
  Dsk->SetStoreTrackPoints(kFALSE);
  AddModule(Dsk);
  //------------------------  MDT       -----------------
  PndMdt *Muo = new PndMdt("MDT",kTRUE);
  Muo->SetBarrel("fast");
  Muo->SetEndcap("fast");
  Muo->SetMuonFilter("fast");
  Muo->SetForward("fast");
  Muo->SetMdtMagnet(kTRUE);
  Muo->SetMdtCoil(kTRUE);
  Muo->SetMdtMFIron(kTRUE);
  AddModule(Muo);
  //------------------------  FTS       -----------------
  FairDetector *Fts= new PndFts("FTS", kTRUE);
  Fts->SetGeometryFileName("fts.geo");
  AddModule(Fts);
  //------------------------  FTOF      -----------------
  FairDetector *FTof = new PndFtof("FTOF",kTRUE);
  FTof->SetGeometryFileName("ftofwall.root");
  AddModule(FTof);
  //------------------------  RICH      ----------------
  FairDetector *Rich= new PndRich("RICH",kFALSE);
  Rich->SetGeometryFileName("rich_v2_shift.geo");
  AddModule(Rich);
}
```

Centralized geometry definition

Setting the input

```
//-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  // TString inputGenerator =
  // EvtGen -> "xxxxxxxx.dec"
  // DPM    -> "dpm_xxxxx"
  // FTF    -> "ftf_xxxxx"
  TString  inputGenerator = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  …
  fRun->SetGenerator();
  …
```

If inputGenerator contains:

“.dec” -> EvtGen

“dpm” -> DPM

“ftf” -> FTF

not case-sensitive

Output files:

- XXX_sim.root (simulation)
- XXX_par.root (parameters)

Klaus is implementing prefix to add

EvtGen: initial state particle automatically found from .dec file

Box generator?

```
sim_box(Int_t nEvents = 100, TString  SimEngine ="TGeant3", Double_t BeamMomentum = 6.231552)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  inputGenerator = "box_1pi_1GeV_theta10-120";
  //--------------------------------------------------------------------------
  // -----   Create the Simulation run manager ----------------------------
  PndMasterRunSim *fRun = new PndMasterRunSim();
  fRun->SetInput(inputGenerator);
  fRun->SetName(SimEngine);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->SetNumberOfEvents(nEvents);
  fRun->SetBeamMom(BeamMomentum);
  // -----   Initialization   ---------------------------------------------
  fRun->Setup();
  // -----   Geometry   ---------------------------------------------------
  fRun->CreateGeometry();
  // -----   Event generator   --------------------------------------------
  FairBoxGenerator* boxGen = new FairBoxGenerator(13, 1); // 13 = muon; 1 = multipl.
  boxGen->SetPRange(1.,1.); // GeV/c
  boxGen->SetPhiRange(0., 360.); // Azimuth angle range [degree]
  boxGen->SetThetaRange(10., 120.); // Polar angle in lab system range [degree]
  boxGen->SetXYZ(0., 0., 0.); // cm
  fRun->SetGenerator(boxGen);
  // -----   Add tasks   --------------------------------------------------
  fRun->AddSimTasks();
  // -----   Intialise and run   ------------------------------------------
  fRun->Init();
  fRun->Run(nEvents);
  fRun->Finish();
};
```

Klaus is implementing a single line parser

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/sim_box.C

New digitization macro

```
void digi_complete(Int_t nEvents = 0)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  input          = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  TString  output         = "digi";
  TString  friend1        = "";
  TString  friend2        = "";
  TString  friend3        = "";
  TString  friend4        = "";

  // -----   Initial Settings   ----------------------------------------
  PndMasterRunAna *fRun= new PndMasterRunAna();
  fRun->SetInput(input);
  fRun->SetOutput(output);
  fRun->SetFriend1(friend1);
  fRun->SetFriend2(friend2);
  fRun->SetFriend3(friend3);
  fRun->SetFriend4(friend4);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->Setup();

  // -----   Add tasks   -----------------------------------------------
  fRun->AddDigiTasks();

  // -----   Intialise and run   ---------------------------------------
  fRun->Init();
  fRun->Run(0, nEvents);
  fRun->Finish();
  exit(0);
}
```

Fixed style name

Input -> original inputGenerator

Output -> only suffix

Complete list of digi tasks

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/digi_complete.C

New reconstruction macro

```
void reco_complete(Int_t nEvents = 0)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  input          = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  TString  output         = "reco";
  TString  friend1        = "digi";
  TString  friend2        = "";
  TString  friend3        = "";
  TString  friend4        = ""
  // -----   Initial Settings   ----------------------------------------
  PndMasterRunAna *fRun= new PndMasterRunAna();
  fRun->SetInput(input);
  fRun->SetOutput(output);
  fRun->SetFriend1(friend1);
  fRun->SetFriend2(friend2);
  fRun->SetFriend3(friend3);
  fRun->SetFriend4(friend4);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->Setup();

  // -----   Add tasks   -----------------------------------------------
  fRun->AddRecoTasks();

  // -----   Intialise and run   ---------------------------------------
  PndEmcMapper::Init(1);
  fRun->Init();
  fRun->Run(0, nEvents);
  fRun->Finish();
  exit(0);
}
```

Fixed style name

Input -> original inputGenerator

Output -> only suffix

Friends -> only suffix

Complete list of reco tasks

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/reco_complete.C

New pid macro

```
void pid_complete(Int_t nEvents = 0)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  input          = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  TString  output         = "pid";
  TString  friend1        = "digi";
  TString  friend2        = "reco";
  TString  friend3        = "";
  TString  friend4        = "";
  // -----   Initial Settings   ----------------------------------------
  PndMasterRunAna *fRun= new PndMasterRunAna();
  fRun->SetInput(input);
  fRun->SetOutput(output);
  fRun->SetFriend1(friend1);
  fRun->SetFriend2(friend2);
  fRun->SetFriend3(friend3);
  fRun->SetFriend4(friend4);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->Setup();

  // -----   Add tasks   -----------------------------------------------
  fRun->AddPidTasks();

  // -----   Intialise and run   ---------------------------------------
  PndEmcMapper::Init(1);
  fRun->Init();
  fRun->Run(0, nEvents);
  fRun->Finish();
  exit(0);
}
```

Fixed style name

Input -> original inputGenerator

Output -> only suffix

Friends -> only suffix

Complete list of pid tasks

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/pid_complete.C

All-in-one macro

```
void full_complete(Int_t nEvents = 0)
{
  //-----User Settings:-----------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  input          = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
  TString  output         = "pid";
  TString  friend1        = "";
  TString  friend2        = "";
  TString  friend3        = "";
  TString  friend4        = "";
  // -----   Initial Settings   ----------------------------------------
  PndMasterRunAna *fRun= new PndMasterRunAna();
  fRun->SetInput(input);
  fRun->SetOutput(output);
  fRun->SetFriend1(friend1);
  fRun->SetFriend2(friend2);
  fRun->SetFriend3(friend3);
  fRun->SetFriend4(friend4);
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->Setup();
  // -----   Add tasks   -----------------------------------------------
  fRun->AddDigiTasks(kFALSE);
  fRun->AddRecoTasks(kFALSE);
  fRun->AddPidTasks();
  // -----   Intialise and run   ---------------------------------------
  fRun->Init();
  fRun->Run(0, nEvents);
  fRun->Finish();
  exit(0);
}
```

Complete list of digi+reco+pid tasks

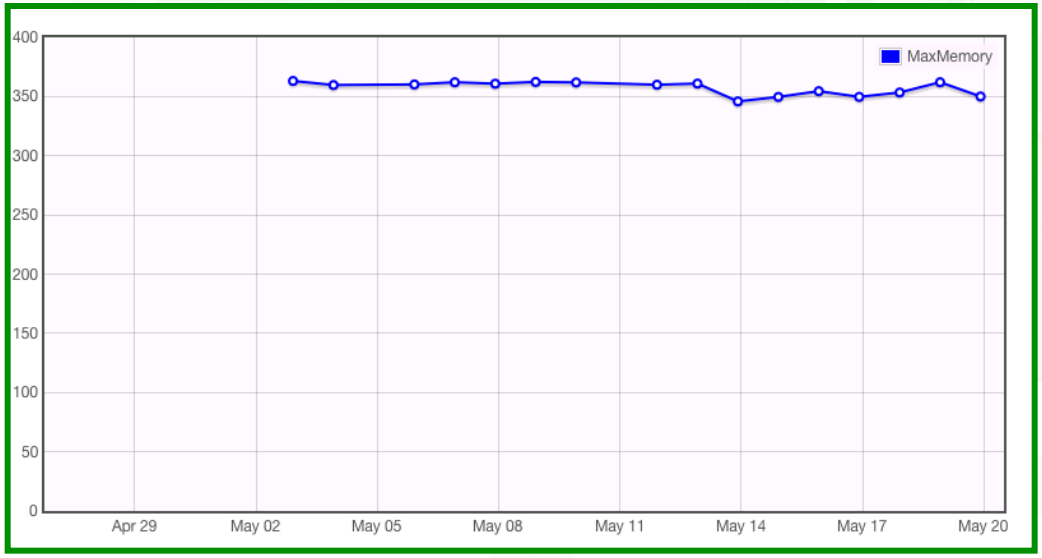
Persistency of digi+reco objects OFF

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/master/full_complete.C

PndMasterRunXxx:Finish()

```
<DartMeasurement name="MaxMemory" type="numeric/double">345.84</DartMeasurement>
<DartMeasurement name="CpuLoad" type="numeric/double">0.947874</DartMeasurement>

[INFO   ] Output file is                    psi2s_jpsi2pi_jpsi_mumu_pid.root
[INFO   ] Friend file is                    psi2s_jpsi2pi_jpsi_mumu_digi.root
[INFO   ] Friend file is                    psi2s_jpsi2pi_jpsi_mumu_reco.root
[INFO   ] Parameter ROOT file is            psi2s_jpsi2pi_jpsi_mumu_par.root
[INFO   ] Parameter ASCII file is           all.par
[INFO   ] Real time 24.3598 s, CPU time 23.09s
[INFO   ] CPU usage 94.7874%
[INFO   ] Max Memory 345.84 MB
[INFO   ] Macro finished successfully.
```



Dashboard

reco_complete3.C

Max Memory

Alternative tasks/geometry?

```
void recoqa_complete(Int_t nEvents = 0)
{
  //-----User Settings:---------------------------------------------------
  TString  parAsciiFile   = "all.par";
  TString  input          = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
…
  // -----   Initial Settings   ---------------------------------------------
  PndMasterRunAna *fRun= new PndMasterRunAna();
  fRun->SetInput(input);
…
  fRun->SetParamAsciiFile(parAsciiFile);
  fRun->Setup();

  // -----   Add tasks   ----------------------------------------------------
  /// Ideal Track finder
  PndMCIdealTrackFinderNewLinks* idealTracking = new PndMCIdealTrackFinderNewLinks();
  idealTracking->AddBranchName("MVDHitsPixel");
  idealTracking->AddBranchName("MVDHitsStrip");
  idealTracking->AddBranchName("STTHit");
  idealTracking->AddBranchName("GEMHit");
  fRun->AddTask(idealTracking);

  /// QA task
  PndTrackingQualityTaskNewLinks* trackingQA = new PndTrackingQualityTaskNewLinks("SttMvdGemGenTrack", "IdealTrack");
  fRun->AddTask(trackingQA);

  // -----   Intialise and run   --------------------------------------------
  PndEmcMapper::Init(1);
  fRun->Init();
…
```

PndMasterRunXxx has all the FairRunXxx functionalities

Instead of fRun->AddXxxTasks()

https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/macro/qa/run3/recoqa_complete.C

PndMasterRunXxx::SetEventCountRate()

```
PndMasterRunAna *fRun= new PndMasterRunAna();
…
fRun->SetEventCountRate(xxx);
…
```

xxx = 100 by default

```
#####Mapper:filltubearray######
fGeoType=1
[INFO   ] Object FtofPoint describing the branch in the folder structure was found
-I- PndFtofHitProducerIdeal: Intialisation successfull
[INFO   ] FairRunAna::Run() After checking, the run will run from event 0 to 2000.
Event 100/2000 : time   15.9 sec, (   178 sec remaining)
Event 200/2000 : time   24.0 sec, (   157 sec remaining)
Event 300/2000 : time   33.6 sec, (   153 sec remaining)
Event 400/2000 : time   43.1 sec, (   146 sec remaining)
Event 500/2000 : time   54.2 sec, (   143 sec remaining)
Event 600/2000 : time   62.7 sec, (   131 sec remaining)
Event 700/2000 : time   71.7 sec, (   121 sec remaining)
Event 800/2000 : time   81.7 sec, (   113 sec remaining)
Event 900/2000 : time   90.2 sec, (   102 sec remaining)
Event 1000/2000 : time   98.5 sec, (    92 sec remaining)
Event 1100/2000 : time  107.4 sec, (    82 sec remaining)
Event 1200/2000 : time  116.3 sec, (    73 sec remaining)
Event 1300/2000 : time  126.7 sec, (    65 sec remaining)
Event 1400/2000 : time  135.9 sec, (    55 sec remaining)
Event 1500/2000 : time  147.6 sec, (    47 sec remaining)
Event 1600/2000 : time  158.1 sec, (    38 sec remaining)
Event 1700/2000 : time  167.2 sec, (    28 sec remaining)
Event 1800/2000 : time  175.1 sec, (    19 sec remaining)
Event 1900/2000 : time  183.3 sec, (     9 sec remaining)
Event 2000/2000 : time  192.6 sec, (     0 sec remaining)
================================================
```

PndEventCounterTask (Ralf)

Everything documented!

FairRoot/PandaRoot

Main Page | Related Pages | Modules | Namespaces | Classes | Files | Examples

Class List | Class Index | Class Hierarchy | Class Members

Public Member Functions | Private Attributes | List of all members

PndMasterRunSim Class Reference

Class for the master simulation chain. More...

```
#include <PndMasterRunSim.h>
```

Documentation in Doxygen

Public Member Functions

	PndMasterRunSim ()
	Default constructor.
Bool_t	**Setup** ()
	Initial setup.
void	**Finish** ()
	Final diagnostics.
void	**CreateGeometry** ()
	It creates all the standard geometry volumes.
void	**AddSimTasks** ()
	Add simulation tasks.
void	**SetGenerator** ()
	Set the event generator.
void	**SetGenerator** (FairBoxGenerator *boxGen)
	Set the event generator for FairBoxGenerator.
void	**SetGenerator** (**PndBoxGenerator** *boxGen)
	Set the event generator for **PndBoxGenerator**.
void	**SetDpmFlag** (Int_t Mode)
	Set the DPM flag.
void	**UseDpmGenerator** ()
	Use DPM as event generator.
void	**UseFtfGenerator** ()
	Use FTF as event generator.
void	**UseEvtGenGenerator** (**TString** fEvtGenFile)
	Use EvtGen as event generator.
void	**SetInput** (**TString par**)
	Input of the simulation This string can be: a) the name of the dec file for EvtGen, w/ or w/o .dec b) "dpm" if you want to use dpm c) "ftf" if you want to use ftf.
void	**SetInputDir** (**TString par**)
	Input directory of the simulation.

Summary

New style for simulation, digitization and reconstruction macros
Enhancement of FairRun -> PndMasterRun
Compatible with “old style”

Centralized initialization – less error prone
Centralized geometry definition – always updated
Centralized task list – always updated

Check of max memory and CPU usage, also in dashboard

Examples: macro/master and tutorial!!
QA: run3, run4, dpm3, dpm4, ftf3, ca already modified
Please update detector macros!!!