A Tutorial of BES III Physics Analysis

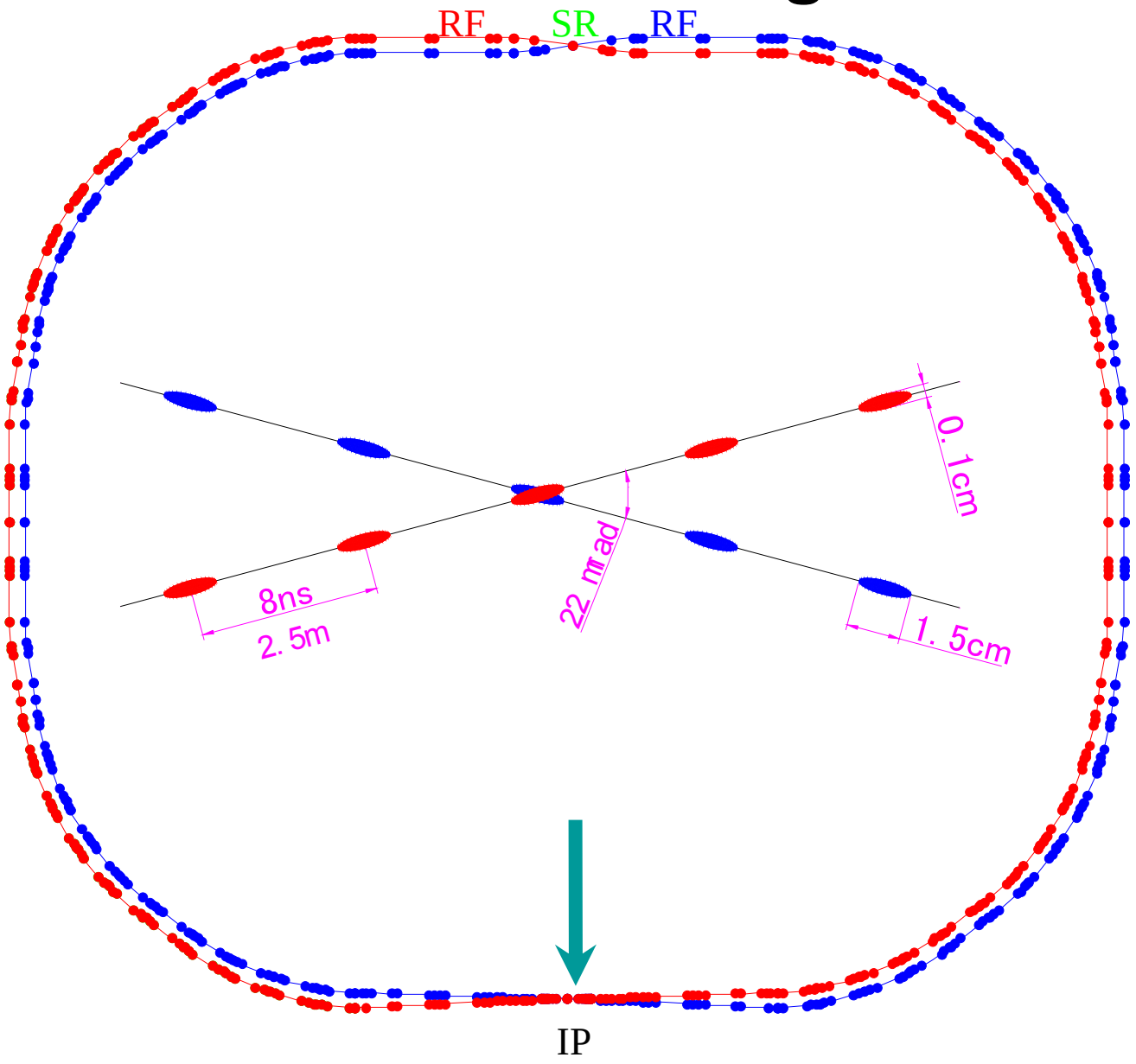
Chengping Shen
shencp@phys.hawaii.edu

Phys. Dept., University of Hawaii

Contents

- Preparation
- Generation, simulation, and reconstruction
- CLHEP, Histogram, N-Tuple
- Event data model
- Analysis tools
  - Vertex fit
  - Kinematic fit
  - PID
- BParticle

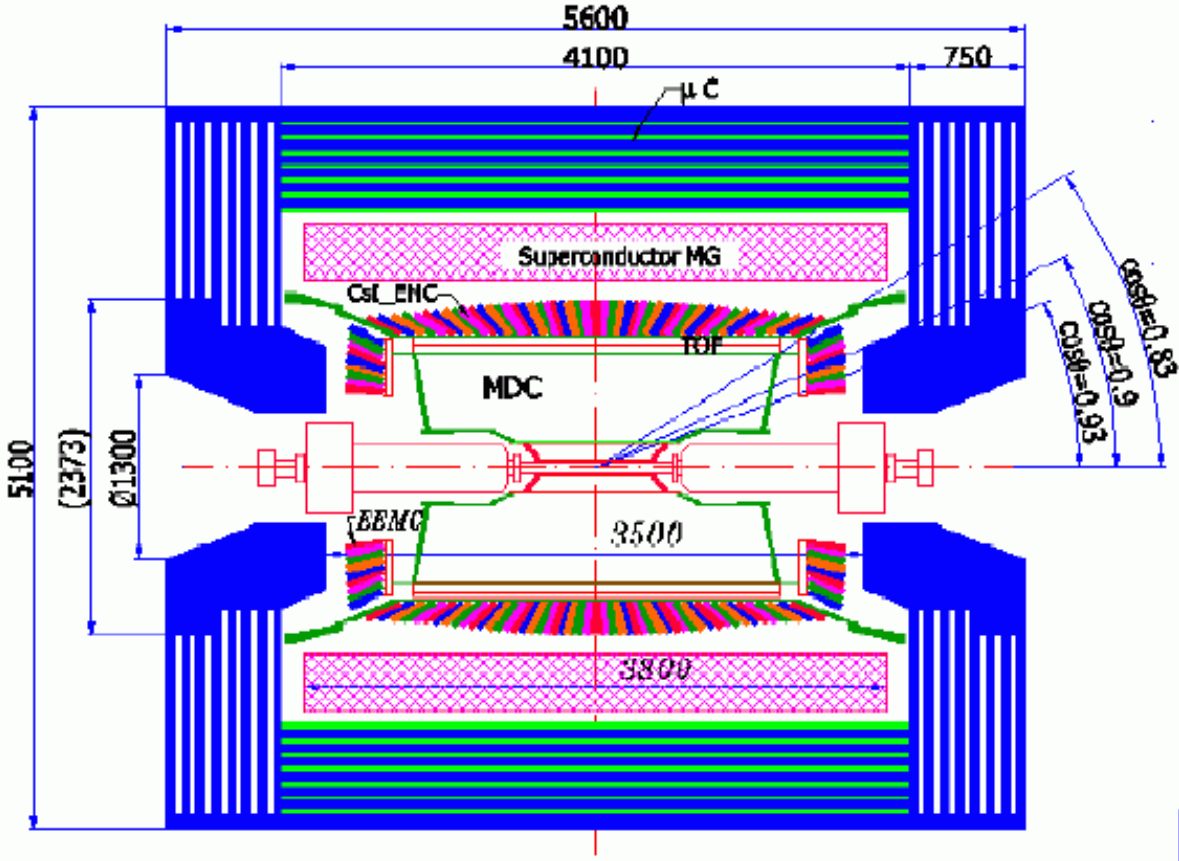
BEPC II Storage ring: Large angle, double-ring



Beam energy:
1.0-2.3 GeV
Luminosity:
1×10^{33} $cm^{-2}s^{-1}$
Optimum energy:
1.89 GeV
Energy spread:
5.16×10^{-4}
No. of bunches:
93
Bunch length:
1.5 cm
Total current:
0.91 A
SR mode:
0.25A @ 2.5 GeV

BESIII detector

Magnet: 1 T Super conducting



MDC: small cell & He gas
σ_{xy}=130 μm
s_p/p = 0.5% @1GeV
dE/dx=6%

TOF:
σT = 100 ps Barrel
110 ps Endcap

EMCAL: CsI crystal
ΔE/E = 2.2% @1 GeV
σz = 0.5 cm/√E

Muon ID: 9 layer RPC

Trigger: Tracks & Showers
Pipelined; Latency = 2.4 ms

Data Acquisition:
Event rate = 3 kHz
Thruput ~ 50 MB/s

- Adapt to high event rate of BEPCII:
  $10^{33}cm^{-2}\ s^{-1}$ and bunch spacing 8ns
- Reduce sys. errors to match high statistics
  photon measurement, PID…
- Increase acceptance

- Drift chamber and its electronics (IHEP, Sichuan, Tsinghua)
- CsI(Tl) calorimeter and its electronics(IHEP, Tsinghua)
- TOF（IHEP, USTC，Tokyo，Hawaii）
- TOF electronics （USTC）
- RPC（IHEP, Uni. of Washington）
- RPC electronics（USTC）
- Trigger（IHEP, USTC）
- DAQ & online software (IHEP, Tsinghua)
- Offline software （IHEP, Peking，Shangdong，Nanjing）
- Superconducting magnet （IHEP，Wang NMR）
- Mechanics (IHEP)
- Technical support (IHEP, Tsinghua）

Physics at BEPCII/BESIII

- Precision measurement of CKM matrix elements
- Precision test of Standard Model
- QCD and hadron production
- Light hadron spectroscopy
- Charmonium physics
- Search for new physics/new particles

A review book on
tau-charm physics at BESIII
~ 800 pages, to be completed this year

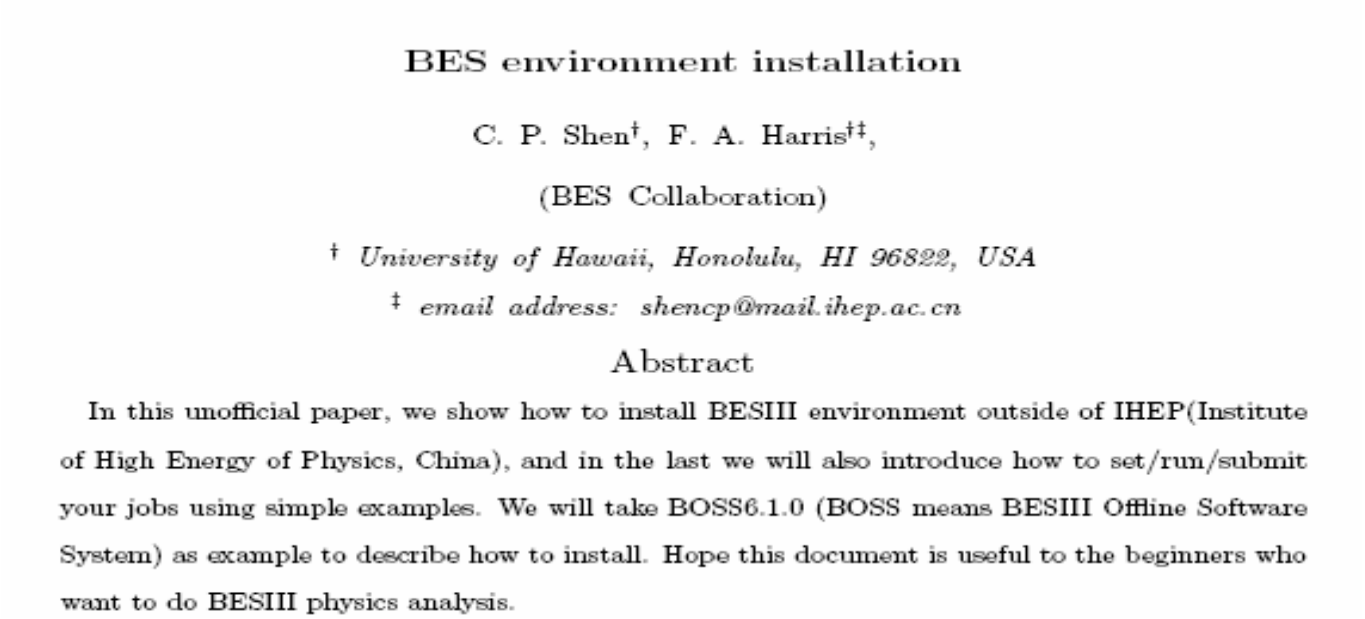
Physics Channel	Energy (GeV)	Luminosity (10^{33} cm^{-2}s^{-1})	Events/year
J/ψ	3.097	0.6	1.0×10^{10}
τ	3.67	1.0	1.2×10^{7}
ψ'	3.686	1.0	3.0×10^{9}
D*	3.77	1.0	2.5×10^{7}
Ds	4.03	0.6	1.0×10^{6}
Ds	4.14	0.6	2.0×10^{6}

Preparation

BOSS--BESIII Offline Software System

- There are two ways to install BOSS

1. Using “pacman” method: easy to install, save time. But if you have sth. wrong in stalling, you must remove everything, and start again. You also need to install database by yourself.
2. Using regular method: we have written a simple manual based BOSS6.3.0. The BOSS6.3.1, the method should be similar. But this method needs much time, and you need to copy and correct many files by yourself.



BES environment installation

C. P. Shen[†], F. A. Harris[†‡],

(BES Collaboration)

[†] *University of Hawaii, Honolulu, HI 96822, USA*

[‡] *email address: shencp@mail.ihep.ac.cn*

Abstract

In this unofficial paper, we show how to install BESIII environment outside of IHEP(Institute of High Energy of Physics, China), and in the last we will also introduce how to set/run/submit your jobs using simple examples. We will take BOSS6.1.0 (BOSS means BESIII Offline Software System) as example to describe how to install. Hope this document is useful to the beginners who want to do BESIII physics analysis.

Use pacman to install

Installation of BOSS 6.3.1

- The software is installed from the network. Currently the only cache in Dubna at http://bes3.jinr.ru is available. It is easy to make mirrors.
- To install the software, you need
  - fresh version of Pacman
  - Linux SLC4 (gcc-3.4)
  - 7 Gb of disk space (5Gb for the software + 2Gb temporary)
  - good network connection
  - time and patience :)
- Installation manual can be found at http://bes3.jinr.ru

Step 1. Install Pacman

- Download Pacman from *http://physics.bu.edu/pacman/*
- Then type

tar xzvf pacman-latest.tar.gz
cd pacman-3.20
source setup.sh (or source setup.csh)
cd ../

- You can check Pacman is working and list available BESIII software versions by

pacman -lc http://bes3.jinr.ru/cache

Step 2. Install the BESIII software

- Create directory to contain the software

mkdir /opt/bes3/

cd /opt/bes3

- Install the software

pacman -get http://bes3.jinr.ru/cache/:6.3.1

- It will take some time, depending on your network connection (about 1.5 Gb should be downloaded and unpacked). For IHEP it takes ~2-3 hours.

For the moment, installation requires partial compilation of packages. Do not get surprised :)

```
[localhost] /home/jemtchou/sw > pacman -get http://bes3.jinr.ru/cache:6.3.1
Do you want to add [http://bes3.jinr.ru/cache] to [trusted.caches]? (y or n): y
Package [6.3.1] found in [http://bes3.jinr.ru/cache]...
Package [Ready] found in [http://bes3.jinr.ru/cache]...
Package [Gaudi_v19r4-i686_slc4_gcc34_opt] found in [http://bes3.jinr.ru/cache]...
....
Downloading [dist_6.3.1-i686_slc4_gcc34_opt.tar.gz] from
[http://bes3.jinr.ru/cache/../kits]...
Untarring [dist_6.3.1-i686_slc4_gcc34_opt.tar.gz]...
Downloading [BES3_DataArea.tar.gz] from [http://bes3.jinr.ru/cache/../kits]...
Untarring [BES3_DataArea.tar.gz]...
Downloading [scripts.tar.gz] from [http://bes3.jinr.ru/cache/../kits]...
Untarring [scripts.tar.gz]...
Converting GDML files ...
Creating InstallArea ...
[localhost]
```

Step 3. Use it!

- Generally, no ~/cmthome/requirements is needed

*source /opt/bes3/setup.sh (*or *.csh)*

source /opt/bes3/scripts/setup.sh (or .csh)

*cmt co TestRelease (*or *cp -r dist/*/TestRelease . Followed by cmt config)*

cd TestRelease//run*

*source ../cmt/setup.sh (*or *.csh)*

boss.exe jobOptions_sim.txt (for example)

- By defaults, the database replica at bes3.jinr.ru is used

Database replica

No database replica (Postgresql) is installed automatically. You should take care of it yourself. The relevant commands are (as superuser):

```
yum install postgresql-server
createdb -U postgres besdb
pg_restore -F t -Upostgres -d besdb besdb_dump.sql.tar
```

where besdb_dump.sql.tar is a database dump, which can be produced at lxplus.ihep.ac.cn by

```
pg_dump -F t -h koala.ihep.ac.cn -Upostgres besdb >
besdb_dump.sql.tar
```

After that you need to change IP address of the database server in the code of Database package, and recompile.

We should invent a good way to synchronize database replicas with the central DB at IHEP!

Boss development environment(BOSS6.3.1)

- Machine: lxslc01(-03).ihep.ac.cn(SLC3), lxslc04(-06)).ihep.ac.cn(SLC4)
- OS : Scientific Linux Cern
- Compiler : gcc-3.4.6
- Framework :Based on GAUDI-v1r14p20031120
- External Libs: CERNLIB□ROOT□CLHEP□Geant4□and etc.
- Developing language : C++□some FORTRAN
- Database : PostgreSql, MySql
- Version management tool : CVS
- Browsing files in CVS repository : ViewCVS
- Configuration management tool : CMT

BOSS Structure

Analysis -- Analysis software and some analysis tools
BesExamples -- Examples for using some packages
BesPolicy -- define some standard patterns of Bes
BesCxxPolicy -- define some standard patterns only for c++ compilier
BesFortranPolicy -- define some standard patterns only for fortran compilier
BesRelease -- for all the packages release
Calibration -- for calibration framework and common tools
Control -- Common tools and framework of boss
Database -- the geometry data IO

DetectorDescription	-- Detector Description packages
Emc	-- for emc
Event	-- different event model
EventDisplay	-- the interface for event display
EventFilter	-- for event filter
External	-- interfaces of external libs
Generator	-- for generators
MagneticField	--
Mdc	-- for mdc
Muc	-- for MuonCounter
Reconstruction	-- all reconsrtuction algorithms
RootPolicy	--
Simulation	-- simulation
TestRelease	-- for some test release
Tof	-- tof
Utilities	-- some common tools

CVS and CMT

- The BESIII Offline Software sources are stored in CVS and can be accessed using the CVS command.
- The BESIII Offline Software libraries have been built using the CMT. Therefore, using the CMT tool is the recommended way to modify existing packages or re-build the examples included in the release.

ViewCVS

File	Rev.	Age	Author	Last log entry
Parent Directory				
Analysis/				
BesCxxPolicy/				
BesExamples/				
BesFortranPolicy/				
BesPolicy/				
BesRelease/				
BesVis/				
BossJob/				
Calibration/				
Control/				
Database/				
DetectorDescription/				
Emc/				
Event/				
EventDisplay/				

http://docbes3.ihep.ac.cn/viewvc/cgi-bin/viewvc.cgi/BossCvs/

What is CMT Configuration management tool

- A set of tools and conventions
  - structures software development or production
    - concepts of areas, packages, versions, constituents
  - organises software into packages
  - describes package properties
  - describes package constituents
  - operates the software production (management, build, import/export, etc...)
    - by transparently configuring and driving the various conventional tools (CVS, make, MSDev, Web, tar, compilers, linkers, archivers, etc...)

Useful variables and commands

- Basic syntax

```
cmt <verb> [ -<option>=<value> <parameters...> ]
```

- Main commands

`cmt config`	re-configure a package
`cmt create ...`	create a new package
`cmt show ...`	monitors configuration parameters
`cmt broadcast ...`	launch shell command over several packages
`cmt co ...`	interface to CVS' checkout operation
`cmt help`	get the list of possible commands
`source cleanup.sh`	undo the source setup
`source setup.sh`	do the source setup

Structure of a package

- /src source files (*.cxx)
- /<package-name> header files (*.h)
- /cmt ***requirements*** *file and some scripts*
- /share jobOptions and other running scripts
- /i386_linux24 *Automatically created at build time*

(libs. Applications and etc. under this directory)

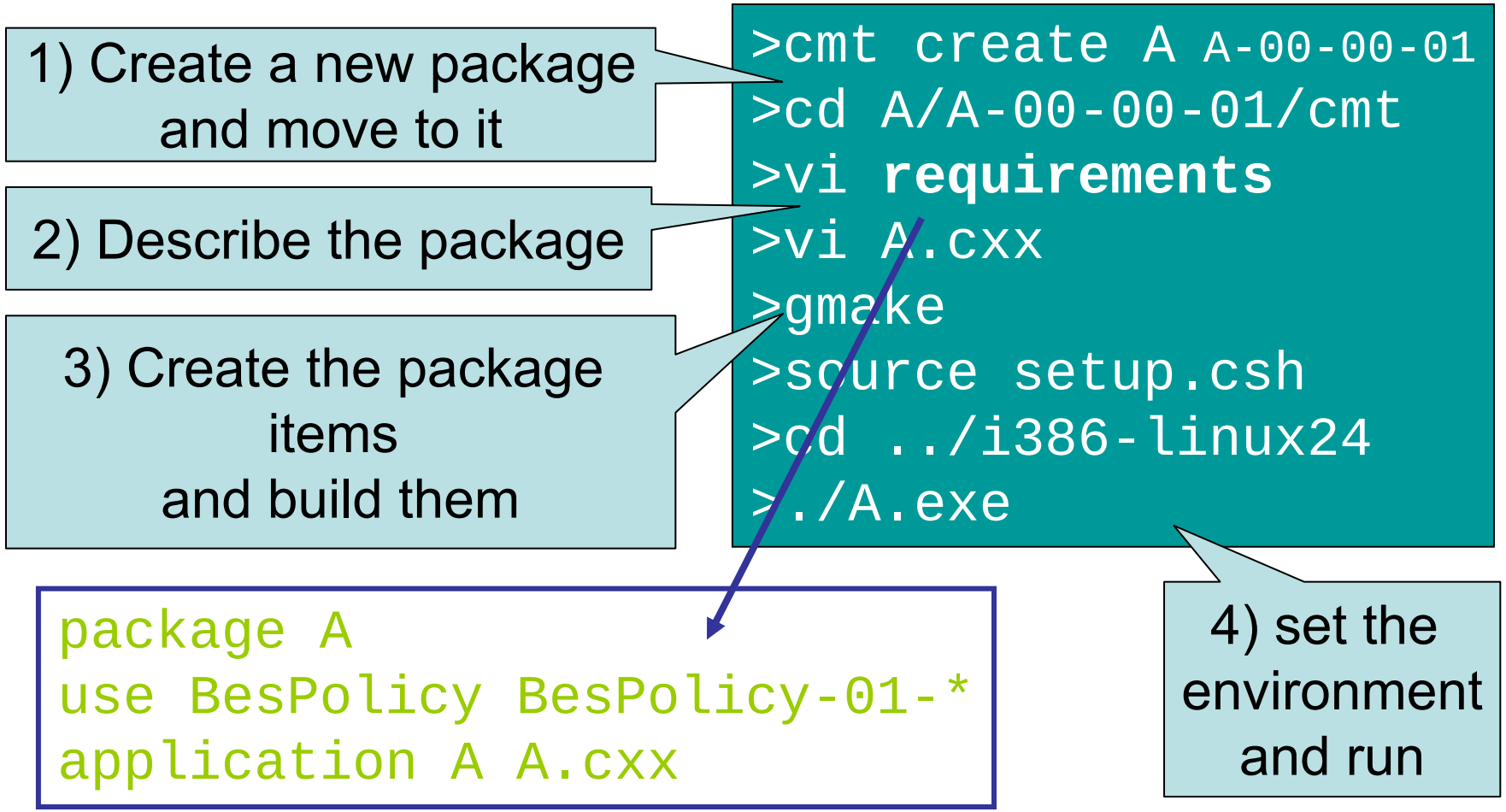
Creating a new package

1. Choosing a location
2. Choosing a name
   - Must be a unique package name in boss (should not either just differentiate by case of letters)
3. Choosing a version
4. The command

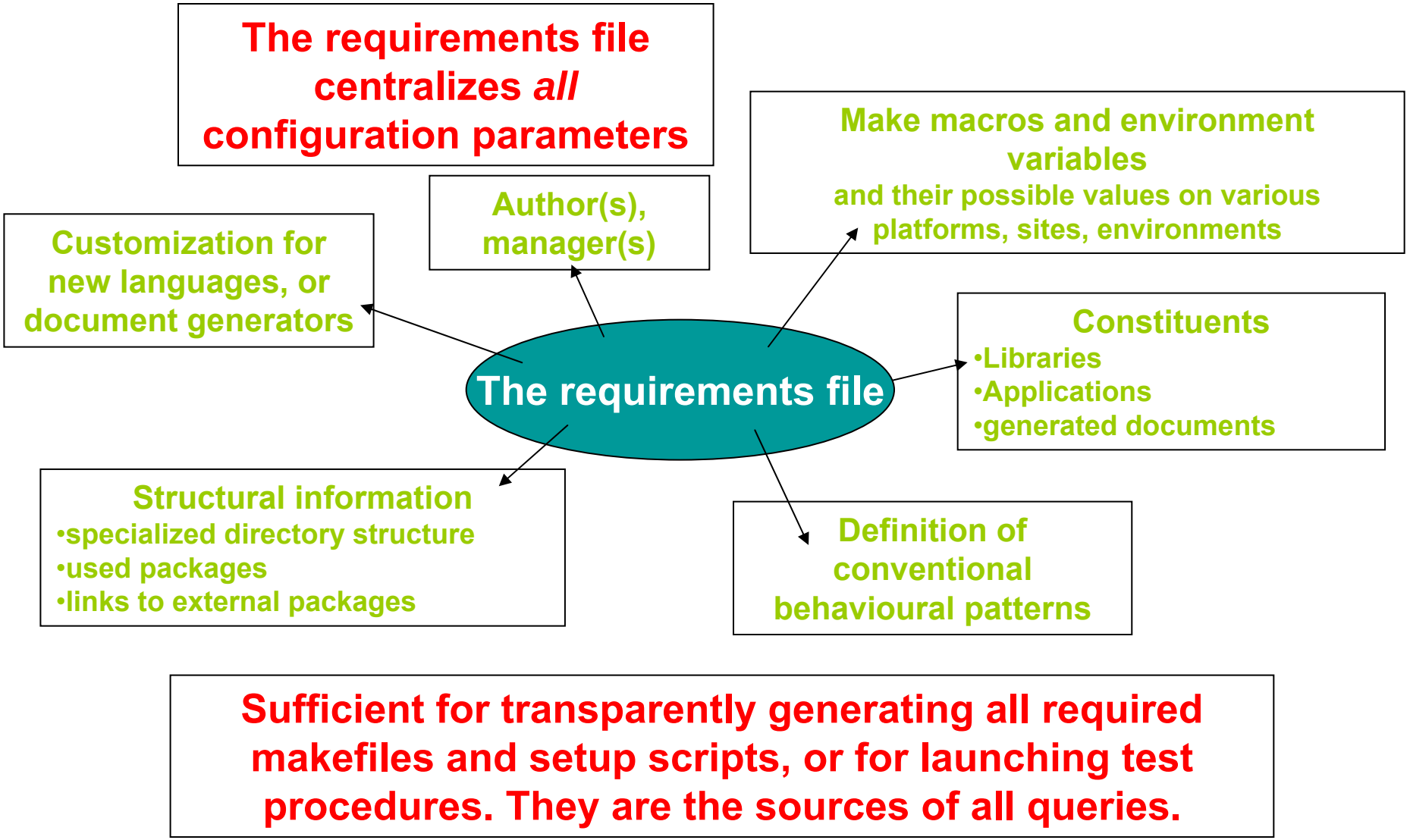
```
cmt  create BesGeoDB BesGeoDB-00-00-01
```

Examples…

- The simplest one
  - One new application with (some) source file(s)



The requirements file



Example of requirements(package)

```
package getacAlg
author shencp@ihep.ac.cn

use BesPolicy BesPolicy-01-*
use GaudiInterface GaudiInterface-01-* External
use DstEvent     DstEvent-*    Event
use EventModel   EventModel-*  Event
use MdcRawEvent  MdcRawEvent-* Mdc
use ParticleID   ParticleID-*  Analysis
use VertexFit    VertexFit-*   Analysis
use BesROOT      BesROOT-00-*     External
use getacAlg  TemplateAlg-* Analysis/Physics
use McTruth      McTruth-*     Event
use AnalEvent    AnalEvent-*   Event


apply_pattern component_library
library getacAlg  *.cxx  components/*.cxx
```

TestRelease/cmt/requirements

```
#======== setup gaudi env      ======
use GaudiInterface   GaudiInterface-*   External
……
#========= for Analysis  ==========
use Analysis          Analysis-*
use AnalysisDemoAlg   AnalysisDemoAlg-*   Analysis
……
#========= for calibration    =======
use CalibSvc      CalibSvc-*      Calibration
……
#========= for reconstruction  ======
use Reconstruction Reconstruction-*
use MdcDedxAlg    MdcDedxAlg-*   Reconstruction
……
#========= for simulation      =======
use BesSim        BesSim-*        Simulation/BOOST
```

```
#======== for Trigger        ===
use Trigger       Trigger-*
……
#======== for DST            ===
use RootIO        RootIO-*        Event
……
#======== for  RawData       ===
use RawDataCnv    RawDataCnv-*    Event
……
#======== for  MagneticField   ==
use MagneticField MagneticField-*
……
#======== for  Generator      ===
use PartPropSvc   PartPropSvc-*  Generator
use KKMC   KKMC-*     Generator
……
```

Checking out a package

- Getting a package out from CVS to modify it
  - Go to your private development area
  - Check out the package (installing the proper CMT directory structure)

cvs login

cmt co TestRelease (or)

cmt co –r TestRelease-00-00-15 TestRelease

  - Get the required revision, and make use of that tag

--Applying a command to all checked out packages

cmt broadcast cmt config

cmt broadcast gmake

Running boss

#running a simple example -- ExHelloWorld

- cd <yourworkarea>
- cmt co TestRelease
- cd Testrelease/*/cmt
- jed requirements

  Add “use ExHelloWorld ExHelloWorld-* BesExamples” to requirements
- cmt broadcast cmt config
- cmt broadcast gmake
- source setup.sh
- cd ../run
- source ../share/TestRelease_links.csh
- boss.exe HelloWorldOptions.txt

Simulation

#include "$KKMCROOT/share/jobOptions_KKMC.txt“

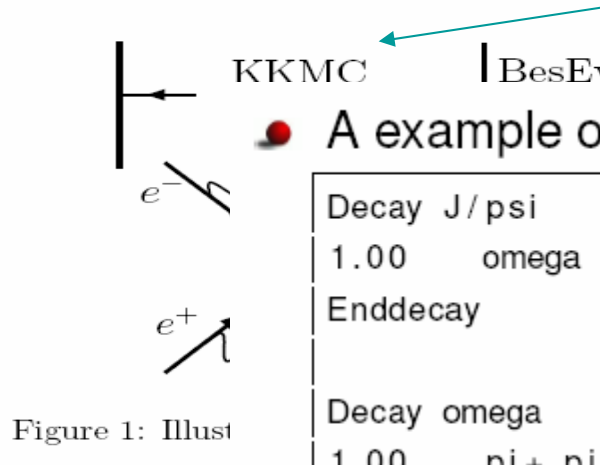


Figure 1: Illust

- A example of decay file (EvtGen)

```
Decay J/psi
1.00    omega pi+ pi-   PHSP;
Enddecay

Decay omega
1.00    pi+ pi- pi0     PHSP;
Enddecay

End
```

BESIII used KKMC+BesEvtGen to
y events.
ate e+e-
ction
ner with the

KKMC.CMS
KKMC.CMS
KKMC.InitializedSeed={400206,1,0};
KKMC.NumberOfEventPrinted=100;
KKMC.GenerateJPsi=true;

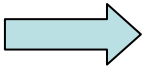
spread,
initialized seed and the number of printed events(output topology)

EvtDecay.userDecayTableName = "jpsi.dec";

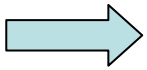
User define decay table, here jpsi.dec define all the decays of jpsi (inclusive sample)

KKMC.ParticleDecayThroughEvtGen=true;

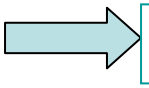
```
ApplicationMgr.DLLs += { "BesEvtGen","BesServices","BesSim"};
ApplicationMgr.TopAlg += { "EvtDecay","BesSim"};

#include "$BESSIMROOT/share/Bes_Gen.txt"
#include "$BESSIMROOT/share/G4Svc_BesSim.txt"

#include "$CALIBSVCROOT/share/job-CalibData.txt"
#include "$MDCCALIBFUNSVCROOT/share/job_MdcCalibFunSvc.txt"
CalibRootCnvSvc.Mdcrootfile="/afs/ihep.ac.cn/bes3/offline/sw/data/con
stant/MdcCalibConst.root";


G4Svc.RunID = -10206;
G4Svc.FADSMacro = "run.mac";
// collision with beam angle
G4Svc.BoostLab = true;

// beam bunch position, unit (mm)
G4Svc.BeamPosX=0;
G4Svc.BeamPosY=0;
G4Svc.BeamPosZ=0;
```



Run Number



Geant4 run macro file, control MDC noise



Define the angle between the e- beam and e+ beam

```
// beam bunch size , unit (mm)
G4Svc.BeamSizeX=0.38;
G4Svc.BeamSizeY=0.0057;
G4Svc.BeamSizeZ=15;

// geometry construction(0:not construct, 1:construct with G4 code)
//                       (2:construct with gdml files)
BesSim.Mdc = 2;
BesSim.Tof = 2;
BesSim.Emc = 2;
BesSim.Muc = 2;
BesSim.PipeSCM = 2;

#include "Sim2Root_Options.txt"
RootCnvSvc.digiRootOutputFile = "./mc10206-2.rtraw";

// OUTPUT PRINTOUT LEVEL
// Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING,
5=ERROR, 6=FATAL )
MessageSvc.OutputLevel  = 6;

ApplicationMgr.EvtMax = 500;
```

Output file name, not ASCII file any more. (new to BOSS6.3.1)

Total Number need to be produced

run.mac

```
//control of smear drift distance in MdcDigitizer
/mdc/digi/smearflag    0 : no smear
                       1 : get spatial resolution from CalFunSvc (default)
                       2 : fixed resolution
// The fixed resolution value (micrometer) when /mdc/digi/smearflag 2
/mdc/digi/spacers 130

//by default, we use noise type 3, which the noise is randomly got from a BESII run.

//noise flag:      0: no noise; 1: add noise; Default 0
/mdc/noise/flag 1
/mdc/noise/type 3
/mdc/noise/level 0.2


#0: get wire efficiency from CalFunSvc, "/mdc/eff/eff" command will be ignored;
#1: If you want to set eff. by hand.    Default: 0
#/mdc/eff/flag 1

#First parameter is layer no. , range -1,0-42.       "-1" means all layers.
#Second is eff. value, default is 1.
#/mdc/eff/eff -1 1.
```

Generator

Monte Carlo Generators for Tau-Charm-Physics at BESIII

Rong -Gang Ping (pingrg@ihep.ac.cn)
Cai -Ying Pang (pangcy@ihep.ac.cn)
July 3, 2007

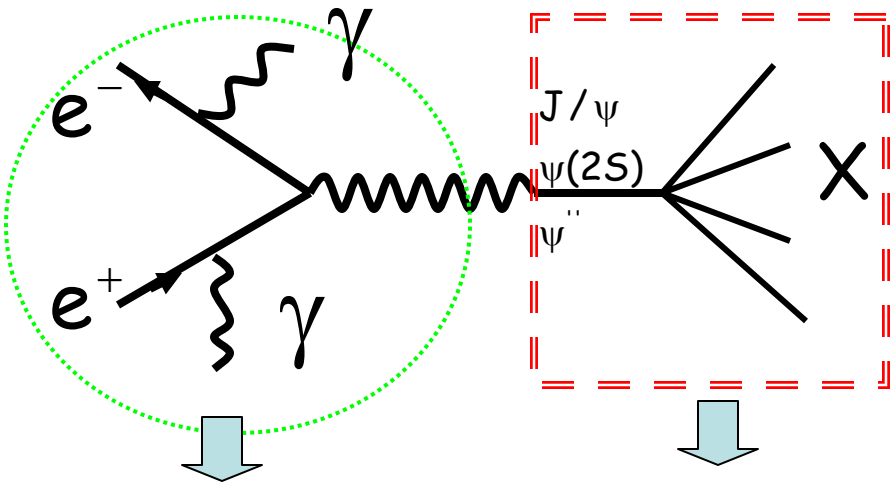
Abstract

This note describes two kinds of Mont-Carlo generator at BESIII. One is inherited from BESII and the other is that based on the amplitude information developed in the EvtGen framework. We focus on the description on the model for the latter, especially for the construction of the amplitude. The models cover the hadronic decays, radiative decays and decays for investigating some physical quantities in charmonium physics. The user is suggested to refer the guide of the EvtGen for some generators applicable to both charm and B physics.

BESIII Generator Framework

- The framework of BESIII event generator is approved to use KKMC+BesEvtGen to generate events.

KKMC(ISR) + BesEvtGen(FSR)



CMS➔Lab. Syst. (Yan Ye) → G4

Both for exclusive and inclusive decays

$e^+e^- \to (n\gamma)\gamma^*$

$(n\gamma)J/\psi$

$(n\gamma)\psi''$

$J/\psi \to X$

$\psi(2S) \to X$

$\psi'' \to X$

GenBes: obsolete

Not recommend to use

Reconstruction

```
#include "ReadRoot_Options.txt"
#include "$CALIBSVCROOT/share/job-CalibData.txt"
#include "$ESTIMEALGROOT/share/job_EsTimeAlg.txt"
//zhangy
#include "$MDCTRKRECONROOT/share/jobOptions_MdcPatRec.txt"
//zangsl
//#include "$TRKRECOROOT/share/jobOptions_TrkReco.txt"

#include "$KALFITALGROOT/share/job_kalfit_all.txt"
#include "$MDCDEDXALGROOT/share/job_dedx_all.txt"
#include "$TRKEXTALGROOT/share/TrkExtAlgOption.txt"
#include "$EMCRECROOT/share/EmcRecOptions.txt"
#include "$TOFRECROOT/share/jobOptions_TofRec.txt"
#include "$MUCRECALGROOT/share/jobOptions_MucRecTrkExt.txt"
#include
"$RECEVENTASSEMBLYALGROOT/share/RecEventAssemblyAlg_Options.txt"

//in order to save MC Truth information
#include "Save2Root_Options.txt"
```

```
// Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
ApplicationMgr.DLLs += {"EmcRecGeoSvc","RootHistCnv","DedxCorrecSvc",
    "TofGeomSvc","TofCaliSvc","MucGeomSvc" };
 MessageSvc.OutputLevel      = 5;
 // Number of events to be processed (default is 10)
  ApplicationMgr.EvtMax = 50000;
  EventCnvSvc.digiRootInputFile = {"./jpsi_getac_etapipi.rtraw" };
  ApplicationMgr.HistogramPersistency = "ROOT";
  EventCnvSvc.digiRootOutputFile ="./jpsi_getac_etapipi.rec";

NTupleSvc.Output = { "FILE101 DATAFILE='/ihepbatch/d03/shencp/tmp/Mdc.root'
                          OPT='NEW' TYP='ROOT'",
          "FILE102 DATAFILE='/ihepbatch/d03/shencp/tmp/fzisan.root' OPT='NEW' TYP='ROOT'",
          "FILE104 DATAFILE='/ihepbatch/d03/shencp/tmp/kalfit.root' OPT='NEW' TYP='ROOT'",
          "FILE103 DATAFILE='/ihepbatch/d03/shencp/tmp/DeDx.root' OPT='NEW' TYP='ROOT'",
          "FILE105 DATAFILE='/ihepbatch/d03/shencp/tmp/Estime.root' OPT='NEW' TYP='ROOT'",
          "FILE201 DATAFILE='/ihepbatch/d03/shencp/tmp/Tof.root' OPT='NEW' TYP='ROOT'",
          "FILE301 DATAFILE='/ihepbatch/d03/shencp/tmp/Emc.root' OPT='NEW' TYP='ROOT'",
          "FILE401 DATAFILE='/ihepbatch/d03/shencp/tmp/Muc.root' OPT='NEW' TYP='ROOT'",
          "FILE998 DATAFILE='/ihepbatch/d03/shencp/tmp/RecAssembly.root' OPT='NEW' TYP='ROOT'"};
```

CLHEP Vector Package

The CLHEP Vector (manual) package implements 3-vectors, 4-vectors, rotations, Lorentz transformations and related concepts.

Avaliable Classes

- Hep3Vector – Vector of real quantities in 3-space
- HepLorentzVector – Vector of real quantities in 4-space
- LorentzVector – Typedefed from HepLorentzVector
- Hep2Vector
- HepRotation
- ... ...

Hep3Vector

Hep3Vector v1(v1x, v1y, v1z), v2(v2x,v2y,v2z)

- dot product: HepDouble v12 = v1.dot(v2);
- cross product: Hep3Vector v12c = v1.cross(v2);
- magnitude: HepDouble Magv1 = v1.mag();
- angle: HepDouble Angv12 = v1.angle(v2);
- perpendicular component: HepDouble perpv = v1.perp();
- polar angle: HepDouble pang = v1.polarAngle(v2);
- azimuth angle: HepDouble aang = v1.azimAngle(v2);

Histograms & N-tuples

- One of the key tools in HEP
- HBOOK was one of the best packages in CERNLIB
- Usage and function is obvious
- In Gaudi it's the same concept
  - First book then fill, requires explicit use of histogram pointer (c.f. HFF1)
  - Simplification in GaudiHistoAlg, combine in a single call, and hide pointer handling in base class

Histograms - Good To Know...

- Histograms are kept in memory
- If not saved - they are lost
- Like all other data - they reside in a Data Store
  - Same access mechanism
- Persistency is configurable
  - HBOOK, ROOT

Book and Fill Histograms

```
#include "AIDA/IHistogram1D.h"
...
// Book 1D histogram in the histogram data store
IHistogram1D* m_hTrackCount= histoSvc()->
     book( "simple", 1, "TrackCount", 100, 0., 3000. );
SmartDataPtr<MCParticleVector> particles( eventSvc(), "/Event/
    MCParticles" );
if ( 0 != particles ) { // Filling the track count histogram
m_hTrackCount->fill(particles->size(), 1.);}
```

Histogram Persistency(ROOT)

CMT requirements file

```
use RootHistCnv v*
```

Job options file (In your analysis job option)

```
ApplicationMgr.DLLs += {“RootHistCnv” };
ApplicationMgr.HistogramPersistency = “ROOT”;
RootHistSvc.OutputFile = "histo.root";
```

N-tuples - Good To Know...

- Cannot be kept in memory
  - Grow and grow and grow...
- Like all other data - reside in a Data Store
  - Same access mechanism
  - Usage simplified by GaudiTupleAlg

Book and fill an N-tuple

- Defining N-tuple tags

```
NTuple::Item<long>                m_ntrk;
NTuple::Item<float>               m_energy;
// Items for the column wise n-tuple
NTuple::Array<long>               m_iNumbers;
NTuple::Array<float>              m_fNumbers;
NTuple::Item<long>                m_n;
```

Book and fill an N-tuple (cont.)

- Booking and Declaring Tags to the N-tuple

```
NTuplePtr nt1(ntupleSvc(), "MyTuples/1");
if ( nt1 ) m_tuple1 = nt1;
else {
  m_tuple1 = ntupleSvc()->book ("MyTuples/1",
       CLID_RowWiseTuple, "example");
  if ( m_tuple1 )    {
    status = m_tuple1->addItem ("Ntrack",    m_ntrk);
    status = m_tuple1->addItem ("Energy",    m_energy);
    status = m_tuple1->addItem ("N", m_n, 0, 100);
    status = m_tuple1->addIndexItem ("FNumbers", m_n,
         m_fNumbers);
  }
  else    {   // did not manage to book the N tuple....
    log << MSG::ERROR << "Cannot book N-tuple:" << long(
         m_tuple1) << endmsg;
    return StatusCode::FAILURE;
```

Book and fill an N-tuple(cont.)

```
  static int n = 0;
  m_ntrk    = long(sin(double(n)) * 52. + 50.);
  m_energy = sin(double(n)) * 52. + 50.;
  m_n       = abs((1234567*(n+1))%100);
  for( int i = 0; i < m_n; i++ ) {
    m_fNumbers[i] = cos(double(2*n)) * 52. + 50.;
  }
  n++;
  status = m_tuple1->write();
  if( status.isFailure() ){
    log << MSG::ERROR << "Cannot fill N-tuple:" << long(m_tuple1
        ) << endmsg;
    return StatusCode::FAILURE;
}
```

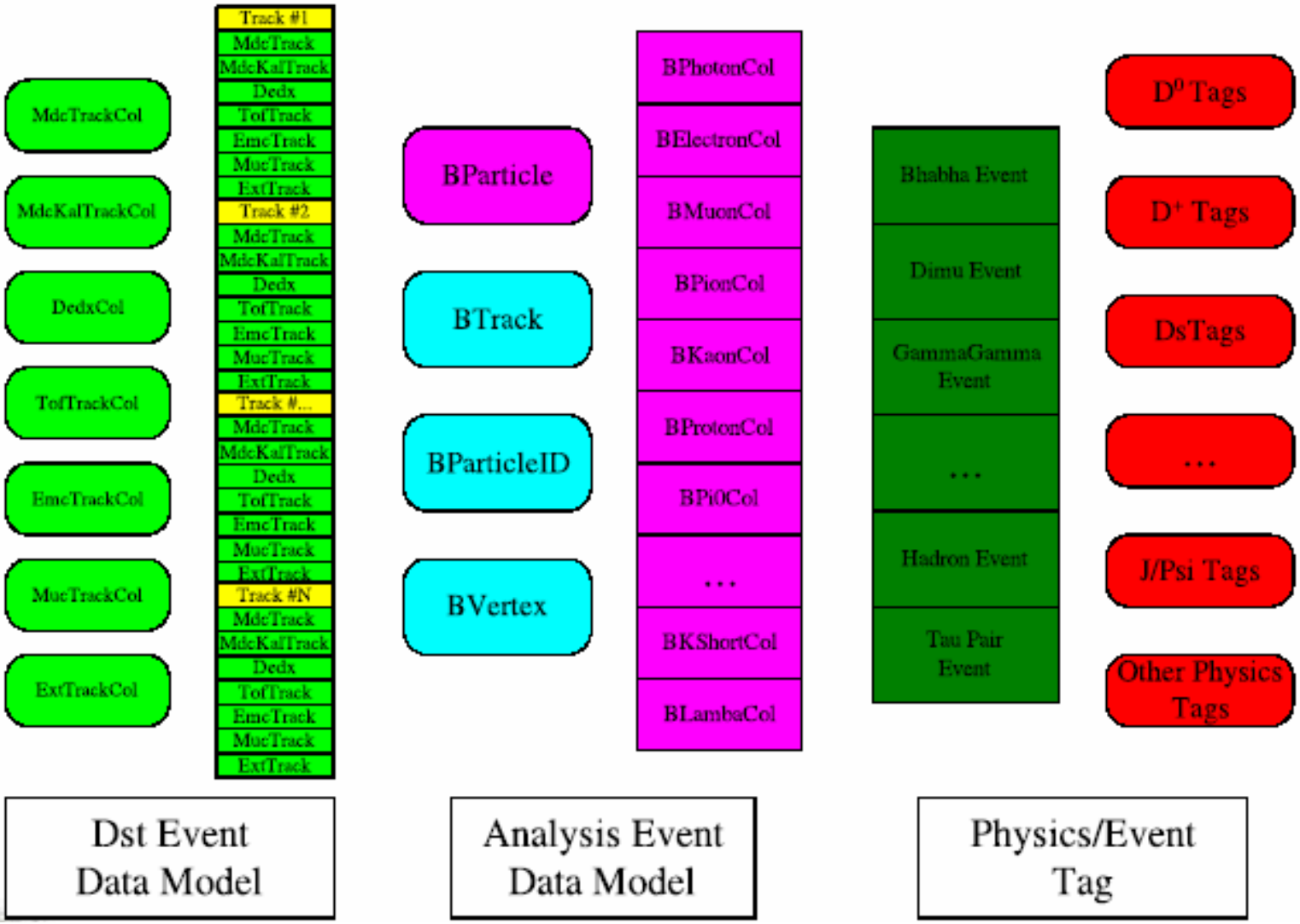
It is up to the user to ensure the arrays do not overflow

N-tuple Persistency

Job options

```
NTupleSvc.Output = {"MyTuples DATAFILE='ntuple.root' OPT='NEW'
TYP='ROOT'"}
```

Analysis Event Data Flow



Example: $J/\psi \rightarrow \rho\pi$

The source program can be found at RhopiAlg
or
In your workarea, type the command:

```
cmt co Analysis/Physics/RhopiAlg
```

Package structure

```
RhopiAlg
'-- RhopiAlg-00-00-06
    |-- RhopiAlg
    |   '-- Rhopi.h
    |-- cmt
    |   |-- Makefile
    |   |-- cleanup.csh
    |   |-- cleanup.sh
    |   |-- requirements
    |   |-- setup.csh
    |   '-- setup.sh
    |-- share
    |   '-- jobOptions_Rhopi.txt
    '-- src
        |-- Rhopi.cxx
        '-- components
            |-- Rhopi_entries.cxx
            '-- Rhopi_load.cxx
```

requirements file

```
package RhopiAlg
author He Kanglin <hekl@ihep.ac.cn>

use BesPolicy BesPolicy-01-*
use GaudiInterface GaudiInterface-01-* External
use DstEvent      DstEvent-*     Event
use EventModel    EventModel-*   Event
use MdcRawEvent   MdcRawEvent-*  Mdc
use ParticleID    ParticleID-*   Analysis
use VertexFit     VertexFit-*    Analysis
use BesROOT       BesROOT-00-*       External

apply_pattern component_library
library RhopiAlg  *.cxx  components/*.cxx
```

Head file *(Rhopi.h)*

```
#include "GaudiKernel/AlgFactory.h"
#include "GaudiKernel/Algorithm.h"
#include "GaudiKernel/NTuple.h"
class Rhopi : public Algorithm {
public:
  Rhopi(const std::string& name, ISvcLocator* pSvcLocator);
  StatusCode initialize();
  StatusCode execute();
  StatusCode finalize();
private:
  double m_vr0cut;
  // define Ntuples here
  NTuple::Tuple*  m_tuple1;       // charged track vertex
  NTuple::Item<double>  m_vx0;
  NTuple::Item<double>  m_vy0;
... ...
```

Access to Event Data

```
#include "EventModel/EventModel.h"
#include "EventModel/Event.h"
#include "EventModel/EventList.h"
#include "DstEvent/DstTrkList.h"
// Get the event header (event and run number)
SmartDataPtr<Event::EventHeader> eventHeader(eventSvc(),"/Event");
// total charged, total neutral, ...
SmartDataPtr<EventList> dstEvent(eventSvc(), EventModel::Dst::
    EventList);
SmartDataPtr<DstTrkListCol> dstTrkListCol(eventSvc(),  EventModel::
    Dst::DstTrkListCol);
```

Dst Track List Collection

Access sub-detector's information through a set of pointers

trackID()		
MdcTrk()	→	DstMdcTrack
MdcKalTrk()	→	DstMdcKalTrack
Dedx()	→	DstDedx
TofTrk()	→	DstTofTrack
EmcTrk()	→	DstEmcTrack
MucTrk()	→	DstMucTrack
ExtTrk()	→	DstExtTrack

Access to Dst Event Data

```
for(int i = 0; i < dstEvent->totalCharged(); i++){
  DstTrkIterator itTrk=dstTrkListCol->begin() + i;
  if(!(*itTrk)->IsMdcTrkValid()) continue;
  DstMdcTrack *mdcTrk = (*itTrk)->MdcTrk();
  m_vx0 = mdcTrk->x();
  m_vy0 = mdcTrk->y();
  m_vz0 = mdcTrk->z();
  m_vr0 = mdcTrk->r();
  m_tuple1->write();
  if(fabs(m_vz0) >= m_vz0cut) continue; // 10 cm
  if(m_vr0 >= m_vr0cut) continue; // 1 cm
  iGood.push_back((*itTrk)->trackID());
  nCharge += mdcTrk->charge();
}
```

Access to Dst Event Data

```
for(int i = dstEvent->totalCharged(); i< dstEvent->totalTracks();
    i++) {
  DstTrkIterator itTrk=dstTrkListCol->begin() + i;
  if(!(*itTrk)->IsEmcTrkValid()) continue;
  DstEmcTrack *emcTrk = (*itTrk)->EmcTrk();
  Hep3Vector emcpos(emcTrk->x(), emcTrk->y(), emcTrk->z());
  ... ...
```

Analysis Tools

- PID
- Vertex fit
- Second vertex fit
- Kinematic fit

References:

- Vertex/Kinematic fitting at BESIII
- Particle Identification
- BESIII Analysis Software and Event Data Model

PID

```
#include "ParticleID/ParticleID.h"
  ParticleID *pid = ParticleID::instance();
  for(int i = 0; i < nGood; i++) {
    DstTrkIterator itTrk = dstTrkListCol->begin() + iGood[i];
    pid->init();
    pid->setMethod(pid->methodLikelihood());
    pid->setChiMinCut(4);
    pid->setDstTrack(*itTrk);
    pid->usePidSys(pid->useDedx() | pid->useTof1() | pid->useTof2
        () | pid->useTofE() | pid->useTofQ() | pid->useEmc()); //
        use PID sub-system
    pid->identify(pid->onlyPion() | pid->onlyKaon());     //
        seperater Pion/Kaon
    pid->calculate();
    if(!(pid->IsPidInfoValid())) continue;
    m_prob_pid = pid->probPion();
```

Another method on PID

Chinese Physics C 1,32 (2008)

The ANN method is contained in the Analysis/PsrticleID package.

To use the ANN method, these codes are suggested:

```
pid->init();
pid->setDstTrack(*itTrk);
 pid->usePidSys(pid->useDedx() | pid->useTof1() | pid->useTof2() | pid->useEmc()
        | pid->useMuc());
 pid->identify(pid->all()|pid->onlyMuon());
 pid->setNeuronPidCuts();
 pid->setMethod(pid->methodNeuronNetwork());
 pid->calculate();
 if(!(pid->neuronStat()> 0)) continue;
  // the detectors' information is not correct if (pid->neuronStat()< 0)
  if( pid->neuronVal()>0.5 &&  pid->neuronVal()<1.5 )
   //the particle is electron if 0.5<pid->neuronVal()<1.5
 if( pid->neuronVal()>1.5 &&  pid->neuronVal()<3.5 )
   //the particle is pion if 1.5<pid->neuronVal()<3.5
 if( pid->neuronVal()>3.5 &&  pid->neuronVal()<4.5 )
   //the particle is kaon if 3.5<pid->neuronVal()<4.5
  if( pid->neuronVal()>4.5 &&  pid->neuronVal()<5.5 )
   //the particle is proton if 4.5<pid->neuronVal()<5.5
  if( pid->neuronVal()>6)
  //the particle is muon if pid->neuronVal()>6
```

AddTrack Utility

```
//                    event data model dependent
void AddTrack(const int number, const double mass, const
    DstMdcTrack *trk);
void AddTrack(const int number, const double mass, const
    DstMdcKalTrack *trk);
void AddTrack(const int number, const double mass, const
    DstEmcTrack *trk);
//                    event data model independent
void AddTrack(const int number, const WTrackParameter wtrk);
```

Vertex Fit

```
#include "VertexFit/VertexFit.h"
  DstMdcTrack *pipTrk = (*(dstTrkListCol->begin()+ipip[0]))->MdcTrk
      ();
  DstMdcTrack *pimTrk = (*(dstTrkListCol->begin()+ipim[0]))->MdcTrk
      ();
  HepPoint3D vx(0., 0., 0.);
  HepSymMatrix Evx(3, 0);
  double bx = 1E+6;
  double by = 1E+6;
  double bz = 1E+6;
  Evx[0][0] = bx*bx;
  Evx[1][1] = by*by;
  Evx[2][2] = bz*bz;
```

Vertex Fit (cont.)

```
VertexParameter vxpar;
vxpar.setVx(vx);
vxpar.setEvx(Evx);

VertexFit* vtxfit = VertexFit::instance();
vtxfit->init();
vtxfit->AddTrack(0, mpi, pipTrk);
vtxfit->AddTrack(1, mpi, pimTrk);
vtxfit->AddVertex(0, vxpar,0, 1);
if(!vtxfit->Fit(0)) return SUCCESS;
```

Kinematic Constraints

```
//  Resonance Constraints
void AddResonance(int number, double mres, int n1, int n2, ...);
//  Total Energy Constraints
void AddTotalEnergy(int number, double etot, int n1, int n2, ...);
//  Total Momentum Constraints
void AddTotalMomentum(int number, double ptot, int n1, int n2
    , ...);
//  Three Momentum Constraints
void AddThreeMomentum(int number, Hep3Vector p3);
//  Four Momentum Constraints
void AddFourMomentum(int number, HepLorentzVector p4);
void AddFourMomentum(int number, double etot);
//  Equal Mass Constraints
void AddEqualMass(int number, std::vector<int> lis1, std::vector<
    int> lis2);
```

Kinematic Fit (4C)

```
#include "VertexFit/KinematicFit.h"
  WTrackParameter wpip = vtxfit->wtrk(0);
  WTrackParameter wpim = vtxfit->wtrk(1);
  KinematicFit * kmfit = KinematicFit::instance();
  ...
      DstEmcTrack *g1Trk = (*(dstTrkListCol->begin()+ig1))->EmcTrk
          ();
      DstEmcTrack *g2Trk = (*(dstTrkListCol->begin()+ig2))->EmcTrk
          ();
      kmfit->init();
      kmfit->AddTrack(0, wpip);
      kmfit->AddTrack(1, wpim);
      kmfit->AddTrack(2, 0.0, g1Trk);
      kmfit->AddTrack(3, 0.0, g2Trk);
      kmfit->AddFourMomentum(0, ecms);
      bool oksq = kmfit->Fit();
```

Kinematic Fit (5C)

```
kmfit->init();
kmfit->AddTrack(0, wpip);
kmfit->AddTrack(1, wpim);
kmfit->AddTrack(2, 0.0, g1Trk);
kmfit->AddTrack(3, 0.0, g2Trk);
kmfit->AddResonance(0, 0.135, 2, 3);
kmfit->AddFourMomentum(1, ecms);
bool oksq = kmfit->Fit();
if(oksq){
  HepLorentzVector ppi0 = kmfit->pfit(2) + kmfit->pfit(3);
  HepLorentzVector prho0 = kmfit->pfit(0) + kmfit->pfit(1);
  HepLorentzVector prhop = ppi0 + kmfit->pfit(0);
  HepLorentzVector prhom = ppi0 + kmfit->pfit(1);
  m_chi2  = kmfit->chisq();
  ... ...
```

Using Kalmanfit

During vertex fit and kinematic fit, it is better to use the track parameters from kalmanfit.

```
for(int i = 0; i < dstEvent->totalCharged(); i++){
```

Application of Kalman Filter Technique to the Kinematic Fitting at BESIII

Kanglin He*
Institute of High Energy Physics
Beijing, 100049

Liang Yan †
Institute of High Energy Physics
Beijing, 100049


```
WTra
    getZError());
WTrackParameter wpim(mpi, pimKalTrk->getZHelix(),pimKalTrk->
    getZError());
WTrackParameter wkp(mk, kpKalTrk->getZHelixK(), kpKalTrk->getZError
    ());
WTrackParameter wproton(mproton, protonKalTrk->getZHelixP(),
    protonKalTrk->getZErrorP());
```

Analysis Framework

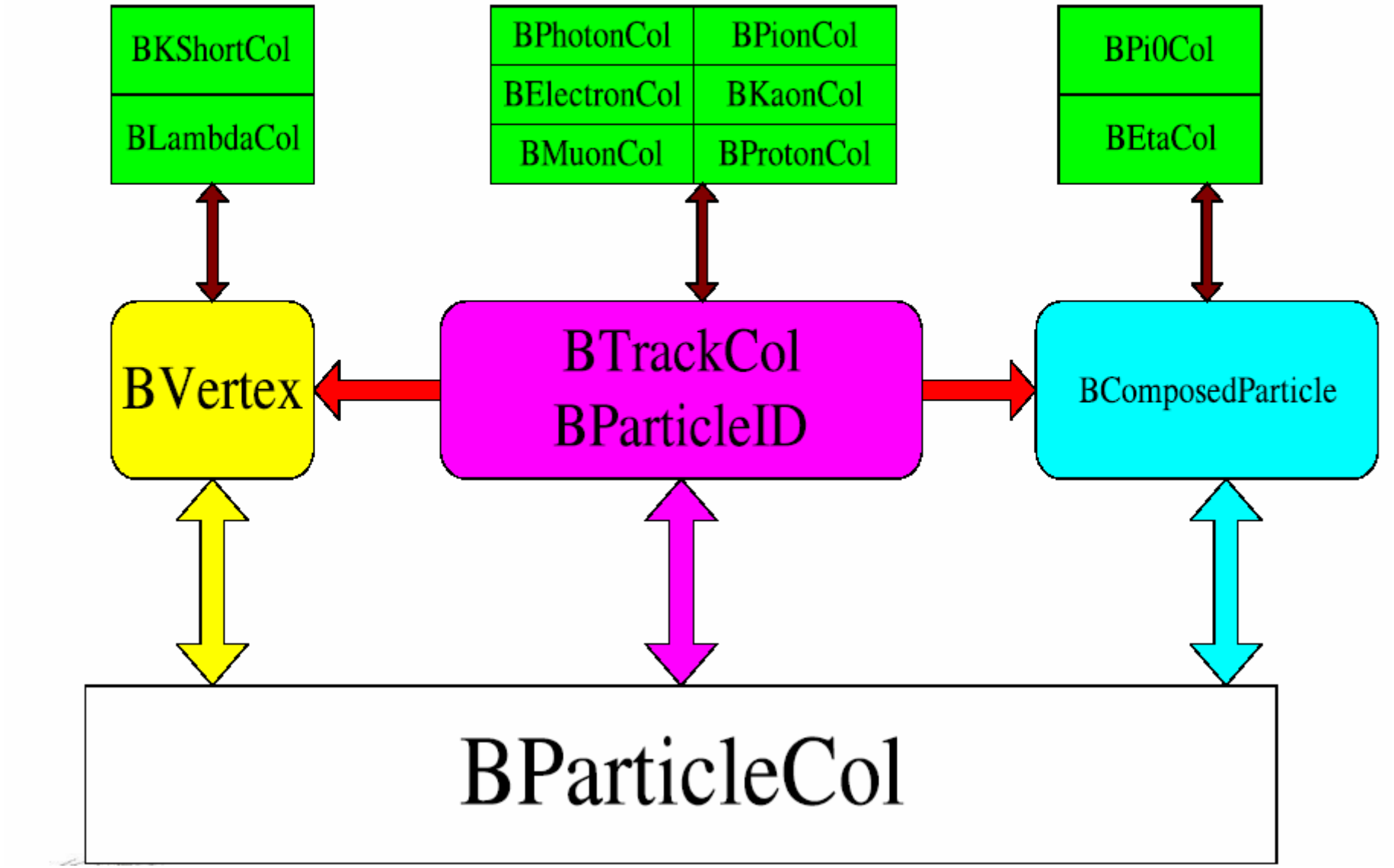
B(es)Particle Project

- Developed by He Kanglin
- Write Particle (Charged tracks, neutrals, vertexes and composed particle) information into TDS. Analysis Algorithm can access these info.
- May be developed as a common standard platform, making analysis job easily and friendly
- Allow Physics groups to design Physics Data model easily

BParticle Contents

- BParticle Collection
  - Event Analysis Event Data Model
- BParticleID Collection
  - Particle identification data
- BTrack Collection
  - Charged and Neutral Track information data
- BVertex Collection
  - (2nd)Vertex Reconstruction data
- BStableParticle Collection
  - Photon, Electron, Muon, Pion, Kaon, Proton
- BComposedParticle Collection
  - π^0, η, K_S, Λ, etc.

BParticle Structure



Index of /BossCvs/Analysis/BParticleAlg/BParticleAlg

File	Rev.	Age	Author	Last log entry
Parent Directory				
BParticles.h	1.3	21 months	hekl	Add Eta, EtaPrime Collection
EtaPrimeReconstruction.h	1.1	21 months	hekl	Add Eta, EtaPrime Collection
EtaReconstruction.h	1.1	21 months	hekl	Add Eta, EtaPrime Collection
GammaConversionReconstruction.h	1.1	18 months	hekl	GAmma conversion vertex was first imported
GoodElectronSelection.h	1.2	18 months	hekl	support neural network PID
GoodHadronSelection.h	1.2	18 months	hekl	support neural network PID
GoodKShortSelection.h	1.1.1.1	22 months	hekl	First Import
GoodLambdaSelection.h	1.1.1.1	22 months	hekl	First Import
GoodMuonSelection.h	1.2	18 months	hekl	support neural network PID
GoodPhotonSelection.h	1.1.1.1	22 months	hekl	First Import
GoodTrackSelection.h	1.1.1.1	22 months	hekl	First Import
KShortReconstruction.h	1.1.1.1	22 months	hekl	First Import

BParticle Property

- Selection criteria controlled by job option file
- Fine tuning of the criteria is needed
- User can change the criteria easily

BParticle Property

- Selection criteria controlled by job option file
- Fine tuning of the criteria is needed
- User can change the criteria easily

```
//
// cuts for good photon selection
//
GoodPhotonSelection.DeltaThetaCut = 20.0;
GoodPhotonSelection.DeltaPhiCut = 20.0;
GoodPhotonSelection.DeltaAngleCut = -18.0;
GoodPhotonSelection.AngleTwoPhoton = 7.0;
GoodPhotonSelection.EnergyThreshold = 0.03;
GoodPhotonSelection.UseBarrelEmc = true;
GoodPhotonSelection.UseEndcapEmc = true;
```

BParticle in Analysis

```
#include "AnalEvent/BParticle.h"
// get pion Lists
SmartDataPtr<BPionCol> pionCol(eventSvc(), EventModel::Anal::
    BPionCol);
// get Kaon Lists
SmartDataPtr<BKaonCol> kaonCol(eventSvc(), EventModel::Anal::
    BKaonCol);
// get Photon Lists
SmartDataPtr<BPhotonCol> photonCol(eventSvc(), EventModel::Anal::
    BPhotonCol);
//get KShort Lists
SmartDataPtr<BKShortCol> ksCol(eventSvc(),EventModel::Anal::
    BKShortCol);
//get pi0 Lists
SmartDataPtr<BPi0Col> pi0Col(eventSvc(), EventModel::Anal::BPi0Col);
```

BTrack in Analysis

```
#include "AnalEvent/BTrack.h"
SmartDataPtr<BTrackCol> trkCol(eventSvc(),EventModel::Anal::
    BTrackCol);
if(!trkCol) return  StatusCode::SUCCESS;
```

BTrack in Analysis (cont.)

```
for(unsigned int i = 0; i < trkCol->size(); i++) {
  BTrackCol::iterator itTrk = trkCol->begin() + i;
  if(!(*itTrk)->isGoodTrack()) continue;
  if((*itTrk)->isCharged()) {
    nCharge += (*itTrk)->charge();
    if((*itTrk)->isPion()) {
      if((*itTrk)->charge() == 1)
        ipip.push_back(i);
      else
        ipim.push_back(i);
    }
  }
  if((*itTrk)->isPhoton())
    iGam.push_back(i);
}
```

BTrack in Analysis (cont.)

```
BTrack *pipTrk = *(trkCol->begin() + ipip[0]);
BTrack *pimTrk = *(trkCol->begin() + ipim[0]);

vtxfit->init();
vtxfit->AddTrack(0, (pipTrk->pionTrack())->wTrack());
vtxfit->AddTrack(1, (pimTrk->pionTrack())->wTrack());
vtxfit->AddVertex(0, vxpar,0, 1);
```

BTrack in Analysis (cont.)

```
BTrack *g1Trk = *(trkCol->begin() + ig1);
BTrack *g2Trk = *(trkCol->begin() + ig2);
kmfit->init();
kmfit->AddTrack(0, wpip);
kmfit->AddTrack(1, wpim);
kmfit->AddTrack(2, (g1Trk->photonTrack())->wTrack());
kmfit->AddTrack(3, (g2Trk->photonTrack())->wTrack());
kmfit->AddFourMomentum(0, ecms);
```

Kalmanfit in BParticle

Controled by the line in the file

`jobOptions_BParticles.txt`

/afs/ihep.ac.cn/bes3/***/6.1.0/Analysis/BParticleAlg/**/share/

```
TrackListDuplication.UseMdcKalTrack = true;
```

Track parameters from Kalmanfit will be used.

Read MC Truth Information

Requirement File:

```
use McTruth      McTruth-*     Event
```

***.cxx file:

```
#include "McTruth/McParticle.h"

//*** MCMADE information
  SmartDataPtr<Event::McParticleCol> mcParticleCol(eventSvc(),
                        "/Event/MC/McParticleCol");
   if (!mcParticleCol){
    log<< MSG::FATAL << "Could not find McParticle" << endreq;
    return( StatusCode::FAILURE);
  }
     Event::McParticleCol::iterator iiter_mc = mcParticleCol->begin();
  for(;iiter_mc != mcParticleCol->end(); iiter_mc++){
      double px=(*iiter_mc)->initialFourMomentum().px()/1000.;
      double py=(*iiter_mc)->initialFourMomentum().py()/1000.;
      double pz=(*iiter_mc)->initialFourMomentum().pz()/1000.;
      double ee=(*iiter_mc)->initialFourMomentum().e()/1000.;
      double mm=sqrt(abs(ee*ee-px*px-py*py-pz*pz));
```

```
pdg=(*iiter_mc)->particleProperty();

      if(pdg==441) //etac
      {
       p4etac.setPx(px);
       p4etac.setPy(py);
       p4etac.setPz(pz);
       p4etac.setE(ee);
      }

  } //for
```

Build and run your package

- Build

```
cd cmt
vi requirement
cmt config
source setup.csh
gmake
```

- Run

```
boss.exe joboption.txt
```

Analysis results

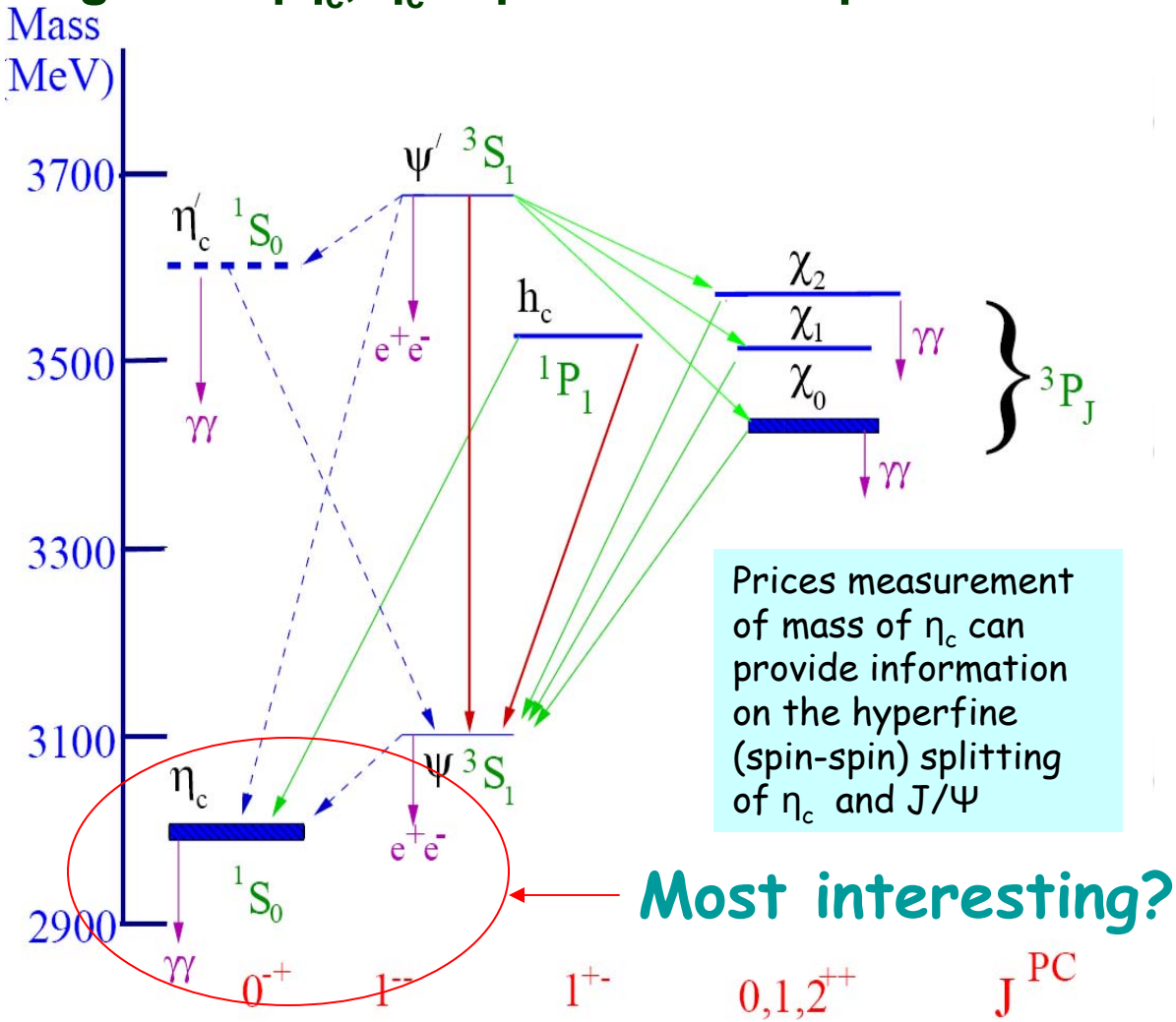
A example of root macro file

```
{
gROOT->Reset();
gStyle->SetOptStat(0); // no statistics display
TFile ff("p2pi2.root");
TH1F *mmass = new TH1F("mmass", "missing mass", 80, 0.8, 1.2);
TCanvas *c1 = new TCanvas("c1", "p pbar pi pi", 200, 10, 600, 400);
pp.Draw("msm>>mmass", "msm>0.8 && msm<1.2");
mmass->GetXaxis()->SetTitle("m (GeV/c^{2})");
mmass.Draw();
c1->Print("msm.eps");
}
```

More examples, see ROOT Tutorials

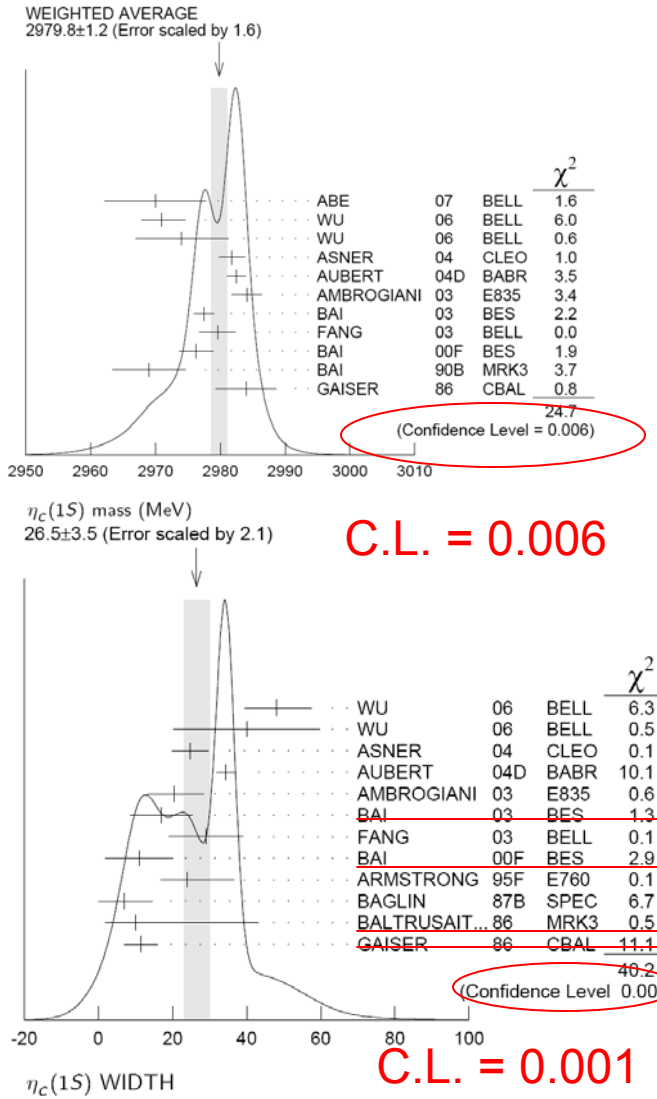
Known problems in BOSS6.3.1

Using $J/\Psi \to \gamma\eta_c$, $\eta_c \to \eta\pi^+\pi^-$ as example to check. Why study ηc decays?



Prices measurement of mass of η_c can provide information on the hyperfine (spin-spin) splitting of η_c and J/Ψ

Most interesting?



C.L. = 0.006

C.L. = 0.001

Br(J/ψ→γηc) is off too!

Our Goal:

Measure the inclusive properties of the η_c with > 10x improved precision

Priority order

1) $\delta M_{\eta c}$: **± 1.2 MeV**
   → **≈± 0.1 MeV**
2) $\delta \Gamma_{tot}$: **± 3.5 MeV**
   → **≈± 0.5 MeV**
3) $\delta Br_{J/\psi \to \gamma \eta_c}$: **± .36%**
   → **≈± 0.05%**

The expected statistical error of η_c mass measurement is about 0.05 MeV at BESIII.

General selection:

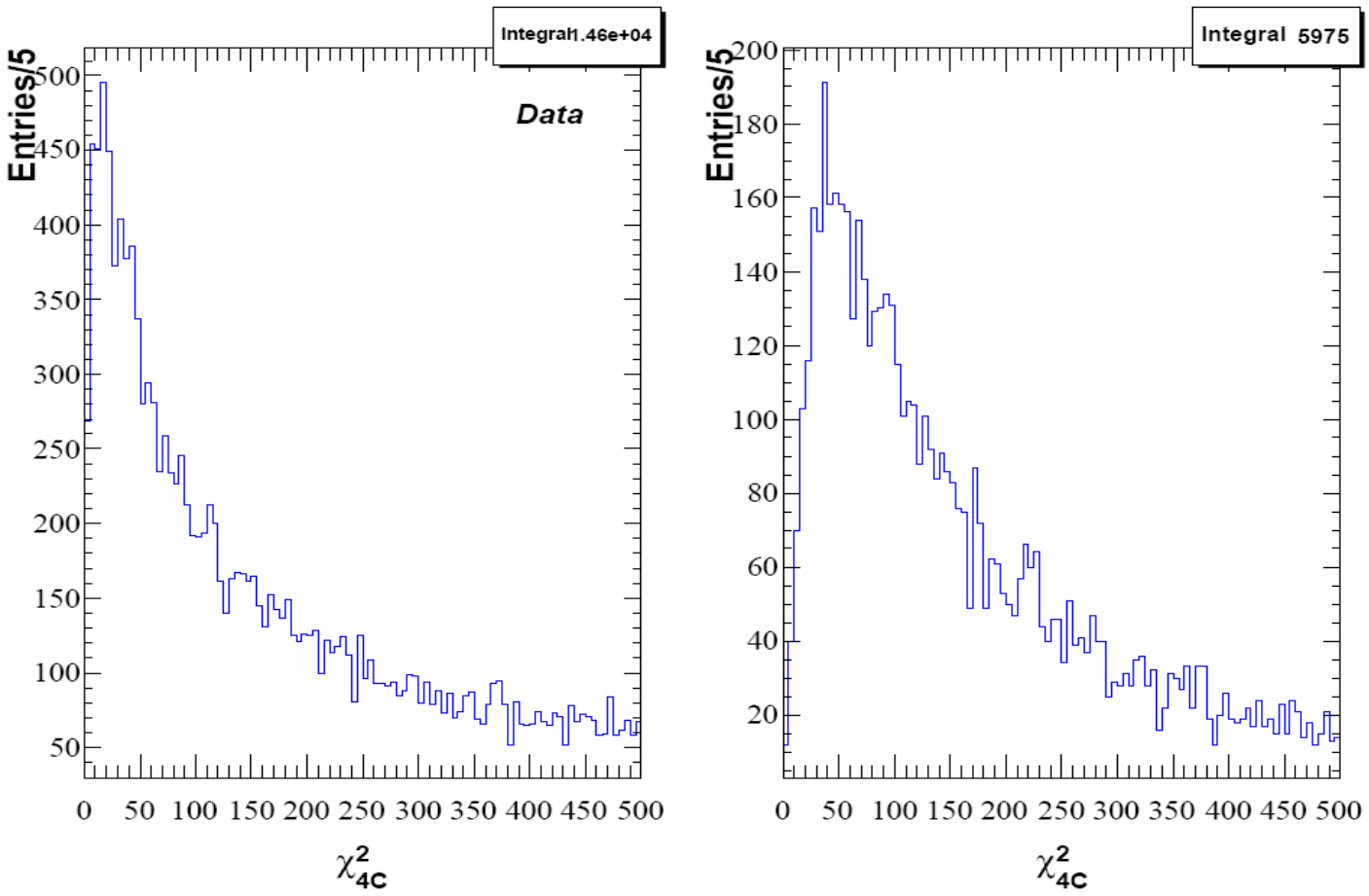
Good charged tracks select: Vr < 1.0cm, |Vz|<5.0cm, |cos**θ**|<0.93

Good photon select: Eraw >= 40MeV, $|\theta_n - \theta_c| \geq 20^\circ$ or $|\phi_n - \phi_c| \geq 20^\circ$

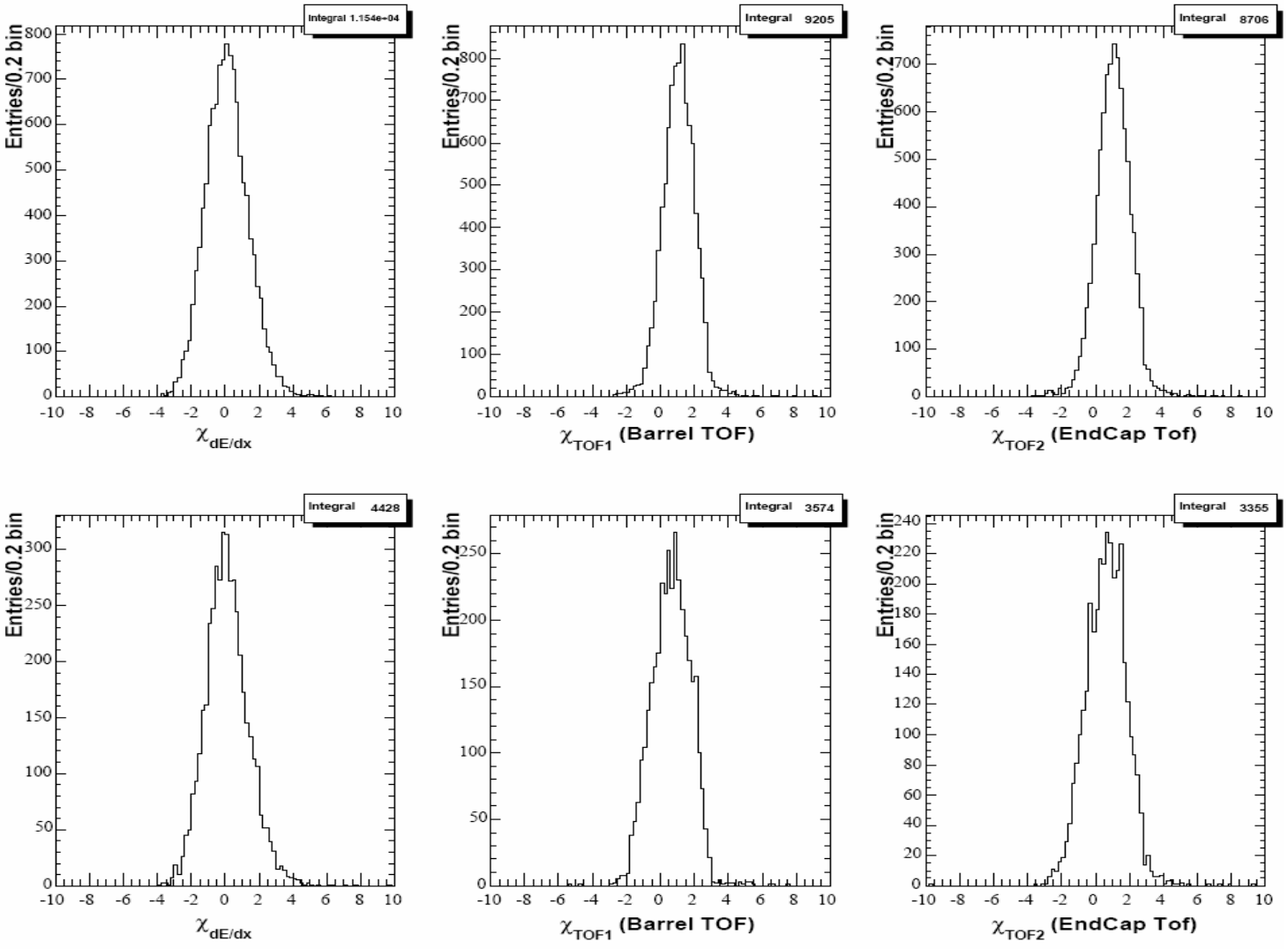
N_{trk}=2 and Σ_Q=0

PID: pi: Prob(K) <Prob(pi)

4C Kinematic Fit, looping over all good photon candidates, select the combination with the minimum χ^2.



- $<\chi^2>\neq 4$. So the χ^2 value problem not solved yet in Boss 6.3.1
- We do not add the χ^2 cut, the default value is 500.



χ distributions are not correct, so the PID efficiency is not reliable.

Inclusive MC

2.0M

PYTHIA

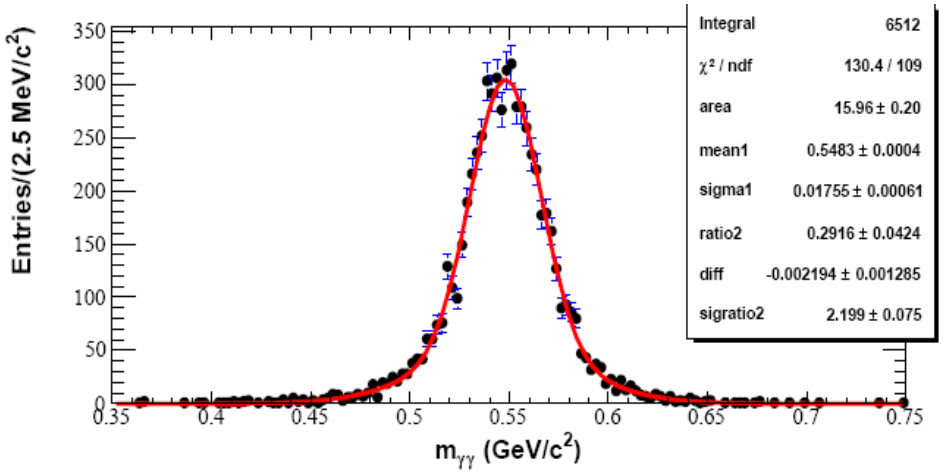
MC

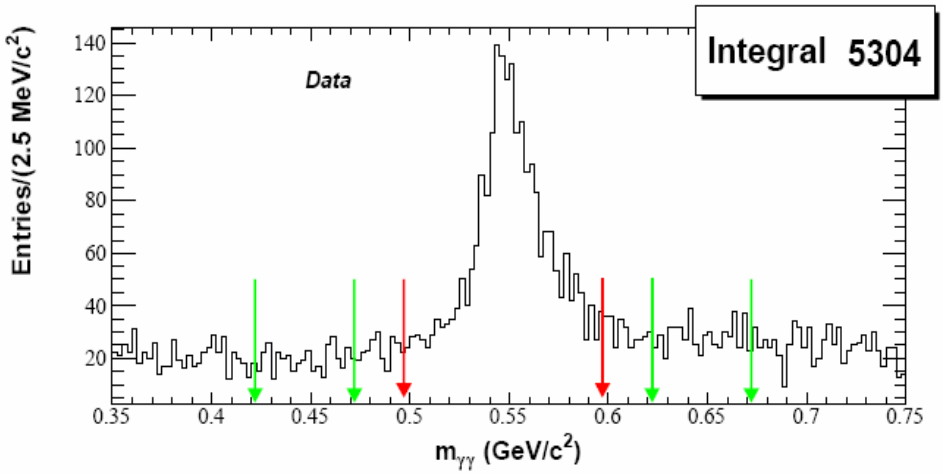
Cut on χ^2 is not used!

Chi distribution in PID

Boss6.3.2

χ distributions are correct!





$\overline{\sigma}_{\gamma\gamma} \approx 16.2\text{MeV}$

Boss6.3.2

$\overline{\sigma}_{\gamma\gamma} \approx 25.5\text{MeV}$

Boss6.3.1

BESIII Physics Yellow Book

τ-Charm Physics at *BES-III*

Editor-in-Chief:
Kuang-Ta Chao and Yifang Wang

Editors:
Shan Jin, Hai-Bo Li, Jian-Ping Ma,
Ya-Jun Mao, Cong-Feng Qiao, Bing-Song Zou