Hall C User Howto

Experiment: HKS

Using and Modifying HKS Ntuples

T. Miyoshi

June 15, 2004

Abstract

This document describes how to play with analysis code. (1) how to play with ntuple, (2) how to examine subroutine code. I describes only the standard Hall C way to create ntuple routine by users.

1 Ntuple

1.1 Caution

Ntuple can be huge file size. If you make ntuple, make sure your disk space is enough to store your file. if disk space is not enough, avoid disk full. Just copy your ntuple file to other local disk where has more disk space.

Do not commit personal ntuple file on CVS.

Do not commit your own hks, enge and coin ntuple file. These should be maintained by autholized person (analysis expert). Especially we should share coin ntuple to avoid multiple results from each university group.

1.2 filename

There are several names of ntuple prepared:

```
beam_ntuple        = 'paw/beam%d.rzdat' for beam
hks_ntuple         = 'paw/hks%d.rzdat'  for hks summary
hks_DCntuple       = 'paw/hdc%d.rzdat'  for HDC
hks_SCINntuple     = 'paw/hscin%d.rzdat' for HKS scintillator
hks_CHntuple       = 'paw/hch%d.rzdat' for HKS Cherenkov
enge_ntuple        = 'paw/enge%d.rzdat' for ENGE summary
enge_DCntuple      = 'paw/edc%d.rzdat' for EDC
enge_SCINntuple    = 'paw/escin%d.rzdat' for Enge scintillator
enge_FISNtuple     = 'paw/efis%d.rzdat' for fission chamber
coin_Ntuple        = 'paw/coin%d.rzdat' for coincidence (the most important!)
```

These can be on/off in REPLAY.PARM. (i.e.,';beam_ntuple' is off)
All the ntuple files are in e01011src/ENGINE:

```
e_scin_ntuple_keep.f for paw/escin%d.rzdat
e_dc_ntuple_keep.f   for paw/hdc%d.rzdat
e_fis_ntuple_keep.f  for paw/efis%d.rzdat
h_scin_ntuple_keep.f for paw/hscin%d.rzdat
h_ch_ntuple_keep.f   for paw/hch%d.rzdat
h_dc_ntuple_keep.f   for paw/hdc%d.rzdat
e_ntuple_keep.f      for paw/enge%d.rzdat
h_ntuple_keep.f      for paw/hks%d.rzdat
c_ntuple_keep.f      for paw/coin%d.rzdat

e_scin_ntuple_init.f for paw/escin%d.rzdat
e_dc_ntuple_init.f   for paw/hdc%d.rzdat
e_fis_ntuple_init.f  for paw/efis%d.rzdat
h_scin_ntuple_init.f for paw/hscin%d.rzdat
h_ch_ntuple_init.f   for paw/hch%d.rzdat
h_dc_ntuple_init.f   for paw/hdc%d.rzdat
e_ntuple_init.f      for paw/enge%d.rzdat
h_ntuple_init.f      for paw/hks%d.rzdat
c_ntuple_init.f      for paw/coin%d.rzdat
```

hks_ntuple, enge_ntuple and coin_ntuple are special files to be shared in the entire HKS group. you can have the other ntuple in your own directory but the ntuple is not maintained in CVS.

1.3 Example of creating your own ntuple

If you don't e01011-replay and e01011src, first

```
cvs checkout e01011src
cvs checkout e01011-replay
```

Then, go to e01011-replay/SRC.

For example, if you want to have your own Cherenkov ntuple in you own directory, copy h_ch_ntuple_init.f and h_ch_ntuple_keep.f to your e01011-replay/SRC directory. Then add the filenames in Makefile,

```
OBJS= engine.o h_ch_ntuple_init.o h_ch_ntuple_keep.o
```

Where engine.o is a default file. If you do make, the file on the same directory becomes up-to-date one. Then, edit the two files. For example...

```
(all '!' is comment)

in h_ch_ntuple_init.f
...
m=0
m=m+1
h_ch_ntuple_tag(m)= 'ac11+'    ! 1. up to 8 characters
m=m+1
h_ch_ntuple_tag(m)= 'ac11-'    ! 1. up to 8 characters

h_ch_ntuple_size= m          !total size

...

-----------------

in h_ch_ntuple_keep.f

integer*4 temp_vector1(3,7),temp_vector2(3,7) ! vector definition
integer*4 la,co
...
```

```
Do i=1,3 ! initialize
  Do j=1,7
    temp_vector1(i,j)=0
    temp_vector2(i,j)=0
  EndDo
EndDo

Do i=1,haer_raw_tot_hits ! fill adc in vector
  la=haer_raw_layer_num(i) ! AC layer number
  co=haer_raw_counter_num(i) !AC segment number
  temp_vector1(la,co)=haer_rawadc_pos(i) !ADC+
  temp_vector2(la,co)=haer_rawadc_neg(i) !ADC-
EndDo


m=0   ! fill contents in ntuple vector
m=m+1
h_ch_ntuple_contents(m)= temp_vector1(1,1) ! ac11+
m=m+1
h_ch_ntuple_contents(m)= temp_vector2(1,1) ! ac11-
...
```

Make sure h_ch_ntuple_size= m is the same as the number of contents in the ntuple keep routine.

After editing, do make. Then, edit REPLAY.PARM in e01011-replay/, turn on hks_CHntuple then start replay. After finishing replay, you may find hch[run number].rzdat in paw directory. To analyze ntuple, read paw manual, please.

2 Examine source code

Editing and committing source code are basically autholized by only analysis expert. However, if you want to examine some subroutine, copy files from e01011src/ENGINE, HTRACKING or ETRACKING to e01011-replay/SRC directory. For example, if you look at h_aero.f and add test_aceff.f, copy e01011src/HTRACKING/h_aero.f to e01011-replay/SRC/, create test_aceff.f, add sentence 'call test_aceff' into h_aero.f then, edit Makefile to add object file like OBJS=engine.o h_aero.o test_aero.o, then do make. Go back to e01011-replay, then replay.

Do not commit your code without checking if (1) compile is passed, (2) no core dump, (3) histogram/ntuple contents are fine, and (4) do not conflict filename.