COMPACT 7.3 USER GUIDE

C. Biino, G. Bocquet, M. Clemencic, B. Hay, V. Kozhuharov,
A. Maier, M. Martini, R. Moore, I. Wingerter

C version

March 14, 2011

Contents

1 Installing COmPACT

The COmPACT installation files are all stored in the NA48 AFS offline directory called */afs/cern.ch/na48/offline2/compact/compact-7.3*. If the computer you work on has access to afs, you only need the reader file: *reader-7.3.tar.gz* which allows you to read COmPACT files. The description of the installation is in section 1.2. The user only needs the user routines, the analysis routines and the main program; these files are included in the *reader-7.3.tar.gz*. If you do not have access to afs, then you need the full compact file: *compact-7.3.tar.gz*. The description of the installation is in section 1.3.
When there is a compact update, the entire tar file is rebuild.

1.1 Compact directories on public afs

The compact public afs area is */afs/cern.ch/na48/offline2/compact/*. In this directory there are subdirectories which can be useful to the user, in particular since by default most of the COmPACT code is not copied to the user area.

- compact-n.m for previous compact versions (before 7.3).
- compact-7.3: contains the sources of the version 7.3. There are subdirectories:

  compact/lib : The COmPACT library (with 2 sub-directories *inc* and *src*).

  compact/rlib : The COmPACT reader library (with 4 sub-directories *inc*, *userinc*, *src* and *anasrc*).

  compact/zlib : The compression library.

  compact/doc : The documentation files.

  compact/tools: The compact tools directory (relevant to build the compact libraries).

- lib directory: contains the compact libraries.
- rlib directory: contains the compact reader libraries.
- There is one directory per operating system containing links the COmPACT libraries (libcompact and libreader).
  - .sun4m_53/lib
  - .alpha_osf20/lib
  - .hp700_ux90/lib
  - .linux/lib
  - .aix/lib

By default, the Makefile in the reader directory points to these directories on public afs.

1.2 Installing COmPACT for Reading with access to afs

Copy the file *reader-7.3.tar.gz* to your home directory then *gunzip* the file before running it through *tar*. To do this you type the following two commands:

```
gunzip reader-7.3.tar.gz
tar xvf reader-7.3.tar
```

or

```
tar xvzf reader-7.3.tar.gz
```

These commands uncompress the distribution file and create the directory structure for COmPACT. The result is a single directory called *reader* which contains the COmPACT code usefull to the user. An example of the output generated by these commands is shown below:

```
> cp /afs/cern.ch/na48/offline2/compact/compact-7.2/reader-7.2.tar.gz .
> gunzip reader-7.2.tar.gz
> tar xvf reader-7.2.tar
x reader, 0 bytes, 0 tape blocks
x reader/userinc, 0 bytes, 0 tape blocks
x reader/userinc/user.h, 12 bytes, 1 tape blocks
x reader/userinc/constants.h, 474 bytes, 1 tape blocks
x reader/userinc/F77ana_define.h, 269 bytes, 1 tape blocks
x reader/userinc/F77_ana.h, 1858 bytes, 4 tape blocks
x reader/userinc/user_cuts.h, 2470 bytes, 5 tape blocks
x reader/usersrc, 0 bytes, 0 tape blocks
x reader/usersrc/fuser_burst.F, 1554 bytes, 4 tape blocks
x reader/usersrc/fuser_cmpEvent.F, 1189 bytes, 3 tape blocks
x reader/usersrc/fuser_cmpEvent.example.F, 11167 bytes, 22 tape blocks
x reader/usersrc/fuser_cmpFilter.F, 1104 bytes, 3 tape blocks
.
.
.
.
x reader/obj, 0 bytes, 0 tape blocks
x reader/depends, 0 bytes, 0 tape blocks
x reader/.last.f77.compile.h, 0 bytes, 0 tape blocks
x reader/.last.c.compile.h, 0 bytes, 0 tape blocks
x reader/compact.job, 5320 bytes, 11 tape blocks
x reader/runcompact symbolic link to
/afs/cern.ch/na48/offline2/compact/tools/compact.pl
x reader/CompactUG-7.2-F.ps symbolic link to
/afs/cern.ch/na48/offline2/compact/compact-7.2/doc/CompactUG-7.2-F.ps
x reader/src, 0 bytes, 0 tape blocks
x reader/src/compact_main.c, 42856 bytes, 84 tape blocks
x reader/.__templist, 138 bytes, 1 tape blocks
x reader/CompactUG-7.2-C.ps symbolic link to
/afs/cern.ch/na48/offline2/compact/compact-7.2/doc/CompactUG-7.2-C.ps
x reader/.compact-status, 3 bytes, 1 tape blocks
x reader/test.list, 472 bytes, 1 tape blocks
x reader/Makefile, 10708 bytes, 21 tape blocks
x reader/userana, 0 bytes, 0 tape blocks
x reader/last.kumac, 152 bytes, 1 tape blocks
>
```

To perform an initial test of COmPACT change to the new *reader* directory and run the GNU make program. The commands to do this are:

```
cd reader
gmake
```

COmPACT will now compile with the default C user routines. These will print out a text dump of the first 20 events of a burst to a file called *compact.txt*. If COmPACT fails to compile ensure that you are running on a supported machine (Linux, DEC Unix, SunOS or HP-UX9) and then that you have the GNU C compiler and CERN libraries installed in the standard places, if not you may need to edit the Makefile.
To use rfio (i.e. to have the possiblity to access data remotely and data stored inside the *CASTOR* system) one needs to modify the Makefile (around line 14):
change

```
USE_RFIO = no
```

to

```
USE_RFIO = yes
```

IMPORTANT

Only the C or F77 user routines are compiled in to COmPACT, not both. The default is to use the C routines, however uncommenting the line containing "F77DEF=...." in *Makefile* by removing the "#" will cause the F77 routines to be called instead.

```
> cd reader
> gmake
Makefile:319: depends/compact_main.d: No such file or directory
Makefile:319: depends/user_init.d: No such file or directory
Makefile:319: depends/user_burst.d: No such file or directory
Makefile:319: depends/user_superBurst.d: No such file or directory
Makefile:319: depends/user_cmpEvent.d: No such file or directory
Makefile:319: depends/user_ke3Event.d: No such file or directory
Makefile:319: depends/user_kmu3Event.d: No such file or directory
Makefile:319: depends/user_mcEvent.d: No such file or directory
.
.
.
.
gmake compact
gmake[1]: Entering directory `/u/vl/wingerte/testcmp/reader'
gcc -O
> -I. -I/afs/cern.ch/na48/offline2/compact/compact-7.2/compact/lib/inc
-I/afs/cern.ch/na48/offline2/compact/compact-7.2/compact/rlib/inc
-I/afs/cern.ch/na48/offline2/compact/compact-7.2/compact/rlib/anainc
-I./userinc
> -I/afs/cern.ch/na48/offline2/compact/compact-7.2/compact/zlib
 -c src/compact_main.c -o ./obj/compact_main.o
.
.
.
.
>
```

1.3 Installing COmPACT for Reading without afs

Copy the file *compact-7.3.tar.gz* to your home directory then *gunzip* the file before running it through *tar*. To do this you type the following two commands:

`gunzip compact-7.3.tar.gz`

`tar xvf compact-7.3.tar`

or

`tar xvzf compact-7.3.tar.gz`

These commands uncompress the distribution file and then create the directory structure for COmPACT. The result is a single directory called *compact* which contains all the COmPACT code. An example of the output generated by these commands is shown below:

```
blocksize = 128
x compact/
x compact/lib/
x compact/lib/Makefile, 4907 bytes, 10 tape blocks
x compact/lib/compact-7.2.x, 38680 bytes, 76 tape blocks
x compact/lib/inc/
x compact/lib/inc/F77common.h, 4886 bytes, 10 tape blocks
x compact/lib/inc/geometry.h, 595 bytes, 2 tape blocks
x compact/lib/inc/physics.h, 593 bytes, 2 tape blocks
x compact/lib/inc/compact-2.2.h symbolic link to compact.h
x compact/lib/inc/compact.h, 23712 bytes, 47 tape blocks
x compact/lib/inc/offsets.h, 18579 bytes, 37 tape blocks
x compact/lib/inc/compact-3.0.h symbolic link to compact.h
x compact/lib/inc/compact-3.1.h symbolic link to compact.h
x compact/lib/inc/compact-3.2.h symbolic link to compact.h
x compact/lib/inc/compact-3.3.h, 23712 bytes, 47 tape blocks
.
.
.
.
x compact/reader/.compact-status, 3 bytes, 1 tape blocks
x compact/reader/test.list, 472 bytes, 1 tape blocks
x compact/reader/Makefile, 10708 bytes, 21 tape blocks
x compact/reader/userana, 0 bytes, 0 tape blocks
x compact/reader/last.kumac, 152 bytes, 1 tape blocks
```

You will then need to compile the two COmPACT libraries:

- libcompact: go into compact/lib directory.
  - Change in the Makefile USE_RFIO =yes to USE_RFIO =no (line18) if your computer dont support rfio.
  - `gmake`
  - When the Makefile has finished, you should have a file called: libcompact.7.3.a
- rlib: go into compact/rlib directory.
  - Change PUBLIC_AFS =yes to PUBLIC_AFS =no in the Makefile (line 16).
  - `gmake`
  - When the Makefile has finished, you should have a file called: libreader.7.3.a

When the two COmPACT libraries are done, you need to go to the directory compact/reader and edit the Makefile to ensure that the libraries are taken from your account and not from afs. You simply need to change the line (17):

```
PUBLIC_AFS = yes
```

to

```
PUBLIC_AFS = no
```

Then the procedure is as explained in the previous section 1.2. The tar file can then be removed.

2 Running COmPACT

2.1 The COmPACT executable

If all went well in the installation phase you now have and executable program called “compact” sitting in the directory. To have a look at the list of command line options available just type:

```
compact -h
```

or just

```
compact
```

To run COmPACT on a data file the most useful options are “-i”, which allows you to specify a single input file, and “-l” which specifies a text file containing the name of a data file on every line.
In the directories /afs/cern.ch/na48/offline2/compact/FilterProd/summaryXX one can find lists of compact and super-compact files available on disk (.list) or through the *CASTOR* system (.castor).
To make compact read one of these files and produce a print out of the first 20 events type of a file:

```
compact -i /castor/cern.ch/na48/na48prod/2004/run/SC/gcmp2416702_1090132191.scmp.1
```

This produces the text file *compact.txt* which contains the event print out.
The online “help” (generated when you type `compact` or `compact -h` shows:

```
user@lxplus007 1:50pm [reader]compact -h
*COmPACT*  - Compact command arguments:
*COmPACT*  -    -i   <file>     declare input file name (replaces s/e/n options)
*COmPACT*  -    -l   <file>     give input file list (replaces s/e/n/i options)
*COmPACT*  -    -co  <file>     declare COmPACT output file name
*COmPACT*  -    -ko  <file>     declare Ke3 output file name
*COmPACT*  -    -kmo <file>     declare Kmu3 output file name
*COmPACT*  -    -mo  <file>     declare MC output file name
*COmPACT*  -    -so  <file>     declare SuperCOmPACT output file name (user filter)
*COmPACT*  -    -soc <file>     declare SuperCOmPACT output file name (charged filte
*COmPACT*  -    -son <file>     declare SuperCOmPACT output file name (neutral filte
```

```
*COmPACT*  -    -ho  <file>      declare HyperCOmPACT output file name (user filter)
*COmPACT*  -    -hoc <file>      declare HyperCOmPACT output file name (charged filte
*COmPACT*  -    -hon <file>      declare HyperCOmPACT output file name (neutral filte
*COmPACT*  -    -scprod-cuts     set default for looser filter cuts (SCProd)
*COmPACT*  -    -nokl2           disable kl2 filter in the neutral split (enabled by
*COmPACT*  -    -smo <file>      declare MC output file name
*COmPACT*  -    -sl  #           set SuperCOmPACT output level
*COmPACT*  -                     default (no option): epsilon standard super-compact
*COmPACT*  -                     #=16+iflag (iflag=1,2,4) for 2pi0, 3pi0, 2 gamma res
*COmPACT*  -                     #=32 for Dalitz summary
*COmPACT*  -                     #=64 for rare decay summary
*COmPACT*  -                     #=128 for four-trackssummary
*COmPACT*  -                     #=a combination of 16+iflag, 32, 64, 128 is also acc
*COmPACT*  -    -srw             rewrite SuperCOmPACT integer event structure
*COmPACT*  -                     from float version
*COmPACT*  -    -db              access cdb (compact database)
*COmPACT*  -    -ndb             do not access cdb (compact database) This is the def
*COmPACT*  -    -cheat           transform compact events into super-compact
*COmPACT*  -    -skip-kab-rec    turn off Kabes reconstruction (when reading cmp even
*COmPACT*  -    -rmin <RMIN>     Only analysis bursts for which run>=RMIN
*COmPACT*  -    -rmax <RMAX>     Only analysis bursts for which run<=RMAX
*COmPACT*  -    -nevt <NEVT>     Only analysis NEVT events and then stops
*COmPACT*  -    -empty #         Empty Burst, EoB, evt lists structures:
*COmPACT*  -                     to save disk space when filtering events
*COmPACT*  -                     #=0: empty burst, evt lists, EoB
*COmPACT*  -                     #=1: empty burst
*COmPACT*  -                     #=2: empty evt lists
*COmPACT*  -                     #=4: empty EoB
*COmPACT*  -                     #=6,3,.. a combination of above is also accepted
*COmPACT*  -   -string <string>  Pass an arbitrary string. This string is available a
*COmPACT*  -   -h                print this help message
```

2.2 Description of the command line options

FIX-ME: must be updated

- **-i** Selection of one compact input file:
  compact -i /castor/cern.ch/na48/na48prod/2004/run/SC/gcmp2416702_1090132191.scmp.1

- **-l** Run compact with a list of input files:
  compact -l /afs/cern.ch/na48/offline2/compact/FilterProd/summary2003/SC_SS1-SS2-SS3_pass2.list: compact will read the SS1,SS2,SS3 super-compact data.

- **-co** Output compact events into the file specified.
  compact -l input.list -co high-mass.cmp:
  compact will read files in *input.list* and write required events (selected from

user_cmpFilter routine) into the file *high-mass.cmp* inside the working directory.

- **-ko** Same as for option -co but for events with K_e^3 compact format. The selection should be done from user_ke3Filter routine.
- **-kmo** Same as for option -co but for events with K_μ^3 compact format. The selection should be done from user_kmu3Filter routine.
- **-mo** Same as for option -co but for monte-carlo events. The selection should be done from user_mcFilter routine.
- **-so** Same as for option -co but the output will have super-compact format. This option is used when producing super-compact samples. As for -co option, the user_cmpFilter routine is used to select events if input data have compact format; if input events have super-compact format, routine user_userCmpFilter should be used.
- **-soc** Output selected events into supercompact. Selection is done by user_superSel3pic() routine in the standard compact library (so-called charged split).
- **-son** The same as -soc option but the selection is done by user_superSel3pin() routine in the standard library (so-called neutral split).
- **-ho** This option is included for filling of user-defined hyperCompact. The selection is done by user_userCmpFilter routine where the hyperCmpEvent structure should be filled. It is the one written to the output file.
- **-hoc** Writing hyperCompact events to the output file starting from SuperCompact. Selection is done inside user_superSel3pic() routine (so-called charged split).
- **-hon** Same as -hon but the selection is done by user_superSel3pin() routine in the standard library (so-called neutral split).
- **-scprod-cuts** An initialization of the cuts for sel3pin and sel3pic routines is done. Cuts are reset to be the ones used during SuperCompact splits production.
- **-scprod-mode** Used during main production of SuperCompact, HyperCompact, Charged and Neutral splits.
- **-nokl2** Doesn't write the Kl2 events into the neutral split. By the default it is not set (i.e the neutral split contains Kl2 events)
- **-smo** Same as for option -so but for super-compact monte-carlo events. The selection should be done from user_superMcFilter routine if input events have super-compact format or from user_McFilter if input events have mc-compact format.
- **-rwts** Rewrite the time stamp of the previous event with the time stamp of the previous event in the input file. Value '-1' is used if a first event in a burst. This option is valid only for Compact and SuperCompact input files

- **-b** This option is used in conjonction with the previous six options to create output files with a fixed size. *compact -l input.list -co high-mass.cmp -b 200MB*: compact will create multiple consecutive output files with 200Mbytes size (the output files will be called high-mass.cmp.1, high-mass.cmp.2, ...).
- **-ndb** With this option (*compact -l input.list -ndb*) the compact (or burst) database is not read - default.
- **-cheat** This option allows the user to read input files with compact format (cmpEvent or mcEvent) but to analyse them as super-compact events (superCmpEvent or superMcEvent). This is usefull when one has a running super-compact program but super-compact have not be produced. Of course, the program takes longer than if the input data were super-compact. . . . . . .
- **-rmin** This option allows to specify a minimum run number. *compact -l input.list -rmin 10540*: only data with $RunNumber >= 10540$ will be analysed. If the input list contains data with $RunNumber < 10540$, data will be readin by compact but not given to analysis routines (i.e. the user routines will only *see* data for the requested runs.
- **-rmax** Same as *rmin* but for maximum run number.
- **-nevt** Limits the number of events to be analysed. When the limit is reached, compact stops.*compact -l input.list -nevt 9999*
- **-empty** This allows to suppress the content of some big compact structure which are known to be no use for the foreseen analysis. The above listing is clear to understand how to handle this option. *compact -l input.list -co rare.cmp -empty 0*: the output file will not contain any data inside the *Burst*, *EndofBurst* structures and the *event lists* will be empty. This is particularly useful when filtering compact data (like rare decays comapct filters) for which this history information is not useful.

2.3 Accessing the run database

FIX-ME: must be updated In the reader directory there is perl script called runcompact_pc. It invokes another perl script called sms_select.pl which connects to the run database. These scripts allow to find out where the data produced by L3 (goldraw,goldcmp,compact) are.
If one just types:

```
runcompact_pc
```

one gets

```
Usage: runcompact\_pc selections [options]
 selections: -f FirstRun[:FirstBurstInRun]
             -l LastRun[:LastBurstInRun]
             -s StartTime (format YYYY_MM_DD_HH_MM_SS or UNIX timestamp)
             -e EndTime (format YYYY_MM_DD_HH_MM_SS or UNIX timestamp)
```

```
            -t BeamType e.g KS, KSKL, test, ...
            -u file_name containing bursts to be processed
options:    -n Max_Number_of_Bursts
            -o Max_Number_of_Bursts with tape optimization
            -p stream_name change input stream (default goldcompact)
            -b send your jour to NQS batch system (qsub)
            -i run interactively
            -d dry run: generate compact.list and quit
            -D debugger program   run your favourite debugger

You have to make at least one selection,
if you specify more criteria, they are put in AND
```

3 Writing Code for COmPACT

The directory structure has been designed so that all the COmPACT library source code and header files are stored in the *lib* directory on /afs/cern.ch/na48/offline2/compact/compact-7.3.

The COmPACT reader code is kept in a library as well; the source code and header files are stored in the *rlib* directory on /afs/cern.ch/na48/offline2/compact/compa 7.3.

The user code and any user header files required are placed in the *usersrc* and *userinc* directories where they will be automatically compiled and linked by the make file. The user code consists of several routines:

Routine	Called when
user_init	program startup
user_burst	new burst header read
user_superBurst	new superCOmPACT burst header read
user_hyperBurst	new hyperCOmPACT burst header read
user_cmpEvent	COmPACT format event read
user_cmpFilter	COmPACT event read and COmPACT output file active
user_ke3Event	Ke3 format event read
user_ke3Filter	Ke3 event read and Ke3 output file active
user_mcEvent	MC format event read
user_mcFilter	MC event read and MC output file active
user_superCmpEvent	SuperCOmPACT format event read
user_superCmpFilter	SuperCOmPACT event read and SuperCOmPACT output file acti
user_hyperCmpEvent	HyperCOmPACT format event read
user_eob	end of burst structure read
user_superEob	SuperCOmPACT end of burst structure read
user_exit	end of program

All these routines must return an integer which is zero if no errors are encountered. For the filter routines a negative number indicates that the event should be written to the output file. Any other number will generate a warning message from COmPACT.

To add your own code modify the relevant dummy user routine supplied. These are carefully commented to show where to add your code and contain a small header needed to interface to COmPACT. The variable names of the various quantities stored for each event and burst are given in appendix I. (C and FORTRAN names are on different files: CompactUG-7.3-C.pdf and CompactUG-7.3-F.pdf)

> **IMPORTANT**
> Only the C or F77 user routines are compiled in to COmPACT, not both. The default is to use the C routines, however uncommenting the line containing "F77DEF=...." in *Makefile* by removing the "#" will cause the F77 routines to be called instead.

Each user routine has it's own file. For the C interface these are called "user_xxxx.c" and for the F77 interface "fuser_xxxx.F".

4 Analysis routines in COmPACT 7.3

From version 4.1 the analysis routines are in the reader library rlib, i.e. they are not anymore in the reader/userana directory. The user has the possibility to modify the routines:

- copy the relevant routine from /afs/cern.ch/na48/offline2/compact/compact-7.3/compact/rlib/anasrc to reader/userana (and the necessary include files from ...../rlib/anainc to reader/userinc
- add the copied file name in the Makefile in the section:

  `UCASRCS =`

  if its a C routine.

  `UFASRCS =`

  if its a fortran routine.
- gmake

IMPORTANT

In version 4.1, most of the analysis routines are called by default, some are called on conditions, *tagtime* is called by default ONLY for good charged and neutral events, because of its CPU use. A selection mechanism has been set-up; its description is given in 8.7 to alter the COmPACT default. The routine steering the analysis routines is in /afs/cern.ch/na48/offline2/compact/compact-7.3/compact/rlib/src/cmpAnalysis.c

4.1 user_stdcmpch.c (H. Fox)

This file contains routines:

- USER_STD_EP which computes quantities related to the tracks: $\frac{E}{p}$ stored in $evt->track[i].EovP$ [TRACK_EOVP(i)], Cluster associated to the track stored in $evt->track[i].LKRclu$ [TRACK_LKRCLU(i)].

4.2 fuser_hodotime.F (M. Lenti)

The USER_HODOTIME routine computes the event time measured with the charged hodoscope which is stored in $evt->achod.hodotime$ [ACHOD_HODOTIME] and other quantities which are stored in structure anacharghod [ACHOD_xxx].

4.3 fuser_lkrtime.F (Lydia Fayard)

The USER_LKRTIME routines computes the event time for neutral events, measured using at most 8 cells from the LKR clusters selected by the neutral selection routine (USER_SEL2PI0). Variables $evt->aneut.LKRtime$ [ANEUT_LKRTIME], $evt->aneut.LKRNHODtime$ [ANEUT_LKRNHODTIME] and $evt->aneut.ntUsed$ [ANEUT_NTUSED] (part of the ananeut structure) are filled by LKR_TIME. The routine is documented in the code.

4.4 fuser_badburst.F (Lydia Fayard)

The USER_BADBURST routine fills the $bur->BadB$ [BUR_BADB_xxx] structure with non zero values if the burst is declared *bad* for the analysis. This routine has to be called inside the USER_BURST routine and then events can be rejected, according to the flags, inside USER_CMPEVENT routine.

4.5 fuser_espy.F (M. Velasco)

This routine computes the energy behind the tracks, measured by the Neutral trigger; this quantity is stored in $evt- > track[itrack].Espy$ [TRACK_ESPY(itrack)] which is a volatile variable i.e. is not saved on disk. If there was no NUT data in at least one view -8888. is filled. If the distance between the track extrapolation to LKR and the column center is larger than 4cm for BOTH views -9999. is filled. If $|Ex-Ey|/Emax >$ 20%, the smallest energy is filled with negative sign.

4.6 fuser_bluefield.F (J.B. Cheze)

This routine has to be applied for 1997 data but not for the next years; for 1998 and later data, the correction from the field in the blue tube is already applied at the reconstruction level. This routine recomputes quantities related to vertex and tracks, taking into account the magnetic field in the blue tube; for the time being, the recomputed quantities are stored in compact free variables; they may, one day, replace the standard variables. c evt-¿vertex[ivertex].anavar[1] = corrected evt-¿vertex[ivertex].y

4.7 fuser_lkrpedcor.F (A. Ceccucci)

```
/***********************************************************/
/* Compact routines to correct 1997 data for LKR pedestal     */
/* variations.  (Augusto Ceccucci)                            */
/*                                                            */
/* The corrections have been measured on groups               */
/* of 40 bursts. Files containing the averages are stored in  */
/* the compact/GeomFiles directory.                           */
/* The correction is not applied to the compact event but     */
/* the correction are stored in a "volatile" structure        */
/*                                                            */
/* LKR_ANAVAR(iclu,1): ecorrke3 corrected for ped. shift (GeV) */
/*                     (evt-lkr[iclu].anavar[0])              */
/* LKR_ANAVAR(iclu,2): pedestal correction (in MeV)           */
/*                     (evt-lkr[iclu].anavar[1])              */
/***********************************************************/
```

From 23-06-98, use latest updated correction flags.

4.8 user_eobdec (G. Barr)

```
/*******************************************************************/
/* int errdvec_(int v[],int vlen);                                  */
```

```
/* v[]      input  Array containing the ProcError array                   */
/* vlen     input  Length of data in v[] (=NProcError)                    */
/* This routine fills the structure eob->aerrdec.decRes[idec]             */
/* (i.e. in fortran EOB_AERRDEC_DECRES_XXXX(idec+1)                       */
/*                                                                        */
/* eob->aerrdec.decRes[idec].code     : y    y    y  0=root,1=echan,2=log */
/* eob->aerrdec.decRes[idec].echan    : n    y    y  echan                */
/* eob->aerrdec.decRes[idec].log      : n    n    y  log                  */
/* eob->aerrdec.decRes[idec].Nerr     : y    y    y  #errors = #calls     */
/* eob->aerrdec.decRes[idec].StatFlag: y    y    n  sat flag             */
/* eob->aerrdec.decRes[idec].evtno    : y    y    n  evtno                */
/* eob->aerrdec.decRes[idec].ts       : y    y    n  ts                   */
/* eob->aerrdec.decRes[idec].x1       : y    y    n  ???                  */
/* eob->aerrdec.decRes[idec].x2       : y    y    n  ????                 */
/*                                                                        */
/* EOB_AERRDEC_DECRES_CODE(idec)      : y    y    y  0=root,1=echan,2=log */
/* EOB_AERRDEC_DECRES_ECHAN(idec)     : n    y    y  echan                */
/* EOB_AERRDEC_DECRES_LOG(idec)       : n    n    y  log                  */
/* EOB_AERRDEC_DECRES_NERR(idec)      : y    y    y  #errors = #calls     */
/* EOB_AERRDEC_DECRES_STATFLAG(idec) : y    y    n  sat flag             */
/* EOB_AERRDEC_DECRES_EVTNO(idec)     : y    y    n  evtno                */
/* EOB_AERRDEC_DECRES_TS(idec)        : y    y    n  ts                   */
/* EOB_AERRDEC_DECRES_X1(idec)        : y    y    n  ???                  */
/* EOB_AERRDEC_DECRES_X2(idec)        : y    y    n  ????                 */
/**************************************************************************/
```

4.9 fuser_nhodtime (M. Lenti)

4.10 fuser_aklflag (F. Marchetto)

```
C*** Routine to interface Compact output to Super-Compact
C*** It should be used also to over-write Variable EVT_AKLTIME
C*** and EVT_AKLERRFLAG in the COmPACT Structure cmpEvent
C***
C*** TimeAkl is the time of the Akl hit closest to Hodoscope
C*** time, i.e. EVT_HODTIME. If EVT_HODTIME is zero, then EVT_NHOTIME is
C*** used. With a proper and small adjustment one could use the event
C*** time as reference.
C*** FlagAkl is an overall flag which is built acording to the number of hit
C*** in a given window:
C*** FlagAkl = 100*Nhit1rms + 10*Nhit2rms + Nhit3rms
```

```
C*** where Nhit1rms is the number of hits within a 1 rms ( 1 ns),
C*** Nhit2rms is the number of hits in the 1-2 rms window (1 to 2 ns)
C*** Nhit3rms is the number of hits in the 2-3 rms window (2 to 3 ns)
C*** [There is a undetermination in this definition: in example 10 hits
C*** in the 1 to 2 rms range give a flag as a single hit in the 1 rms
C*** window. This is extremely unlikely]
```

4.11 fuser_lkrposcor (G. Unal)

This routine corrects the cluster position (applied to 1997, 1998 and 1999 data) using measurements obtained from Ke3 analysis. The corrected x and y positions are stored respectively into LKR_ANAVAR(i,3) and LKR_ANAVAR(i,4) ($lkr[iclu]->anavar[2]$ and $lkr[iclu]->anavar[3]$). Value used are in the files compact/GeomFiles/cpd_pos_97.txt, cpd_pos_98.txt., cpd_pos_99.txt.

4.12 user_TrackVertexCor (I. Wingerter)

This routine steers the correction routines for the spectrometer: geometry, magnet and field in blue field corrections. It is called for all events and calls the required correction routines. Vertices are recomputed after the corrections were applied

Please note that from the 19-11-98, the correction for the field in the blue tube is NOT available in the evt->trackCorr.xxx (TRACKCORR_XXX) structure; it is available for vertices in the evt->vertexCorr.xxx (VERTEXCORR_XXX) if the correction was required. This remark also applies to super-compact: the tracks dont have the correction due to the field in the blue tube applied; the charged summaries do. This is because this correction is applied to the tracks belonging to a given vertex; if a track belongs to more than one vertex, the correction would be applied twice; therefore to correct the vertex it is applied to tracks, then the vertex is computed, then the correction is removed from the track.

4.13 fuser_lkrcalcor (G. Unal)

```
      subroutine user_lkrcalcor_event(IB, QB, IE, QE)
c
c correct cluster energy for calibration effect
c      (1) eta factors for intercalibration 97
c      (2) Time variation of energy scale
c      (3) Overall energy scale factor=1.0029 (should be correct to =< 5*10*
c
```

```
This routine (valid only for 1997 data) works as the previous lkr
correction routines: it overwrites LKR_ANAVAR(i,1) (lkr[i]->anavar[0])
with the corrected value.
```

4.14 fuser_lkrcalcor98 (G. Unal)

This routine contains all the applied neutral corrections for 1998 and 1999 data. It also applies the necessary correction to monte-carlo events.

```
c
c correct cluster energy
c   (0) E vs radius close to beam tube
c   (1) Zero suppression effect
c   (2) Energy loss at low photon energy (from MC)
c   (3) Sharing correction, bias in reconstruction from MC
c   (4) Sharing correction, data/MC shower profile
c   (5) Sharing correction, beam tube effect
c   (6) Energy non linearity (E/p correction)
c   (7) Time variation of Escale (makes sense only if Pedestal correction do
c   (8) Overall Escale
c   (9) Space charge effect correction
c  (10) Pedestal shift variation
c  (11) Eta intercalibration for 99
c
c if icorr = 0, apply corrections on data/MC according to
c    best knowledge
c if icorr !=0 apply only specified corrections (according to
c    bits above)   !! You should know what you are doing !!
c
c  returns in function value word corresponding to corrections done
c
```

The variable LKR_ANAVAR(i,5) (lkr[i]-¿anavar[4]) is filled with the energy measured in the two samples before the pulse.

4.15 murec0999 (T. Gershon)

This routine is the updated version of the muon reconstruction in order to handle 1999 data and MC data.

4.16 user_lkrcalhi2k (A. Ceccucci)

This routine applies a correction to the cluster energy, following the change of calibration due to the change in Lkr temperature which occured during August 2000.

```
/*******************************************************/
/* COmPACT user routine: user_lkrcalhi2k_init()        */
/*                                                     */
/* User routine called to read the correction to the   */
/* LKr calibration for the data collected in 2000      */
/*                                          AC 30/1/01 */
/*******************************************************/
```

4.17 fuser_ghost.F (G.Unal)

This routine was developped in order to reject ghost track. This routine is called for super-compact events and fills sevt->summary[isum].Char[0].anaflag[1] with 1 if the vertex is made with one ghost track (from K_{e3} or $K_{\mu 3}$).

4.18 fuser_lkrsmear.F (G.Unal)

This routine smears the cluster energy (in evt-¿lkr[i].anavar[0] [LKR_ANAVAR(1)])for monte-carlo events.

4.19 mc_etail.F (G.Unal)

This routine compute s effect on energy from resolution tails; the modified energy is stored in sevt-¿cluster[iclu].spare[0] [SCLUSTER_SPARE(1)].

4.20 fuser_pedint.F (M.Scarpa)

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  Feb. 2002 Marcella S.                                           c
c  Ferrara pedestals and bad runs/bursts list (epsi,KS,KL runs)    c
c  4 channels ks/l : 1us 200ns 3us 15us                            c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

This routine subtract pedestal values and stores bad bursts/runs list. The 4 KS(L) channels' pedestal values are stored in pedks(l) burst by burst and subtracted to beam-monitor intragrator compact variables. The compact variables are over-written, so the old value must be stored before. The routine fill also FEBAD_KS(L): negative value (-1) means bad burst or run.

```
c ----------------------------------------------------------------------
c       FEBADR_KS/L: bad runs & burst ks/l channels
c                    -1 bad
c                     0 ok
c                     2 oscillatory behavior but peds computed anyway

        INTEGER FEBAD_KS,FEBAD_KL
        REAL pedks,pedkl
        COMMON/FEMON/pedks(4),pedkl(4),FEBAD_KS,FEBAD_KL
c ----------------------------------------------------------------------
```

IN COMPACT the routine (if wanted) must be called by the user in fuser_cmpEvent.F (user_cmpEvent.c) as

```
CALL USER_PEDINT(bur,evt)   (USER_PEDINT(bur,evt);)
```

IN SUPERCOMPACT the pedestals are always applyed ad the routine is called in cmp2scmp.c
If the routine is called a bit (27) is set in the evt->FlagCorr variable. This will be more usefull when we will make the new compact version and we will introduce a flag to call by default the routine.

4.21 fuser_dcholes.F (M.Scarpa)

From version 6.2.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  Sept. 2002 Marcella S.                                            c
c  catch the hole(s) in dch r/o wire maps through eob rate infos     c
c  1 connector = 16 wires = 1 card in old r/o picture                c
c  new r/o 1 cmc = 2 connectors                                      c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

In *fuserhole.h* the COMMON/HOLESdch/holes(24,16) is defined, used to house the hole-nothole definition for each card(16) in each plane(24: dch1-2-4). A card is a 16 wires group as was defined in the old DCH read-out picture.
The *fuserhole.F* routine detects the holes and fills the array holes(24,16). The strategy is to look at the rate per card(16), in each plane(24), and 'catch' the holes in two steps:

- 1) flag (flag1) the low rate cards w.r.t cut1
  Also the low rate cards at the edges are flagged
- 2) flag (flag2) the asymmetric cards w.r.t cut2
  The edges are excluded from this flag. An hole is found if both flag1 and flag2 are fired and the correspondig card index is filled in holes(plane,card)

The rate per card is computed in raw and stored in the DCHRATES2 common block according to the following relation:
100*(number of hits per card)/(number of hits per plane), where

- #hits per card is stored in eob->SGNeff (EOB_SGNEFF(384)) (16 cards X 24 planes=384)
- #hits per plane is stored in eob->SGNineff ($EOB\_SGNINEFF$(384)) (24 planes from DCH 1-2-4)

All these variables are end-of-burst information and they will be trasmitted to the SuperCompact *eob*-structure.

4.21.1 6.2 new compact variable based on DCH info

In order to allow multiplicity and accidental related study new compact structure *DCHmult* has been introduced which variables are taken from raw common blocks. The same structure is then passed to SuperCompact *SCDCHmult*.

1. The number of hits in the $[-25, 175]$ ns MBX time window, per plane for DCH1-2-4 are coded in:
   evt->DCHEFFmult.MBXPlaneEff[4] (DCHEFFMULT_MBXPLANEEFF(4))
   The maximum number of hit per plane is 64, i.e. 6 bits. We limit to 5 bits. The 24 numbers are coded in 4 words. The less significant bit of the first word is for the first plane od DCH 1, see 4.22. These information is elaborated in 4.22 to have an hint about the accidental activity (see evt->DCHEFFmult.MBXmult).
2. multiplicity in the tighter MBX time window $[-10, 155]$ ns, per plane for DCH 1:
   evt->DCHEFFmult.L1Trk24Eff[2] (DCHEFFMULT_L1TRK24EFF(2)) coded as before. These information are elaborated in 4.23 to determine which events should have been triggered by the L1=2trk+4trk and the result stored in a flag passed to SuperCompact evt->Trk24ON (DCHEFFMULT_TRK24ON).
3. DCH snow effect flagging evt->DCHEFFmult.DCHSnowErr[2] (DCHEFFMULT_DCHSNOWERR(2)) two bit words. The bit meaning is described in table 1
   First word: bit 0-14 dcherr1; bit 16-31 dcherr2.
   Second word: bit 0-14 dcherr3; bit 16-31 dcherr4. These words are in superCompact as well.

4.22 fuser_mboxeff_SC.F (F.Marchetto - M.Holder)

From version 6.2.

0	bit error in 9th time bit plane 8 dch2
1	first word is not a header
2	plane number not as expected
3	chamber number not as expected
4	wrong word count
5	header TimeStamp different from global TS
6	TS difference between headers
7	crate collect status word TS different from global TS
8	token time out
9	data ready time out
10	header-CSC time mismatch
11	less headers than expected
12	more headers than expected
13	hit time out of [-150,350] ns window
14	only for DCH2 or DCH4: snow flag (1—2—4—6—7—11—12).and.14

Table 1: evt->DCHEFFmult.DCHSnowErr description

```
      Function user_mboxeff_SC(ISB,QSB,ISE,QSE)
C***     The output of the routine is a flag stored in user_mboxeff.
C***     ONLY DCH1, DCH2, and DCH4 are considered
C***     The procedure is the following:
C***  a) get the number of hits per plane (hit -> drift time between
C***    -25 and 175)
C***  b) for each view is extracted the number of hits defined as
C***     the minimum number of hits between the two planes making a view
C***  c) the number of extra hits for each view is defined as
C***     the number of hits per view - 2 (where two is the number of hits
C***     expected for a decay channel with two charged particles in the
C***     final state)
C***  d) result is coded into user_mboxeff: for example
C***     user_mboxeff = 3 -> at least 3 views in DCH1 have .ge. Nextra_hit
C***     user_mboxeff = 15 -> at least 3 views in DCH2 have .ge. Nextra_hit
C***     user_mboxeff = 75 -> at least 3 views in DCH4 have .ge. Nextra_hit
C***     etc...
```

4.23 fuser_L1trk24Eff.F (N.Cartiglia)

From version 6.2.

```
      function is_it_l1ok(IB,QB,IE,QE)
C---
C--- Routine to determine which events should have been triggered
C--  by the L1=2trk+4trk
```

Table 2: Routines for the selection of 3π events

Event type		Charged	Neutral
cmpEvent	C	user_sel3pic(bur,evt)	user_sel3pin(bur,evt)
	F	USER_SEL3PIC(IB,IE)	USER_SEL3PIN(IB,IE)
superCmpEvent	C	user_superSel3pic(sbur,sevt)	user_superSel3pin(sbur,sevt)
	F	USER_SUPERSEL3PIC(ISB,ISE)	USER_SUPERSEL3PIN(ISB,ISE)
hyperCmpEvent	C	user_hyperSel3pic(hbur,hevt)	user_hyperSel3pin(hbur,hevt)
	F	USER_HYPERSEL3PIC(IHB,IHE)	USER_HYPERSEL3PIN(IHB,IHE)

```
C---
C     The output of the subroutine is in L1eff
C     l1eff = 1 then the event was trigger also by "ge2trk+ge4trk"
C
```

This routine uses the packed words DCHEFFMULT_MBXPLANEEFF(i) and the array DCHEFFMULT_L1TRK24EFF(i) as input.
See the code for the unpacking of DCHEFFMULT_MBXPLANEEFF.
The same routine exists in SuperCompact and it is called by syperCmpAnanlysis.c and it is filling the variable sevt$-$ >SCDCHEFFmult.Trk24ON (SCDCHEFFMULT_TRK24ON).

5 Analysis routines in COmPACT (versions $\geq$ 7.1)

5.1 Selection routines for 3π events (M. Sozzi)

For the standard selections of $K^\pm \to \pi^\pm\pi^+\pi^-$ and $K^\pm \to \pi^\pm\pi^0\pi^0$ a set of routines is provided (see Table 2).
Those reoutines fill the C structures (FORTRAN common blocks) ana3pic and ana3pin (ANA3PIC and ANA3PIN), see Table 3 and Table 4.

5.2 Filter routine for $Kl2$ decays (L. Fiorini)

The routine `KL2FILTER`[1], defined as

```
INTEGER FUNCTION KL2FILTER(ISB,QSB,ISE,QSE)
INTEGER ISB(2),ISE(2)
REAL*4  QSB(2),QSE(2)
```

inside the file `rlib/anasrc/kl2filter.F`, has been provided by Luca Fiorini to filter $Kl2$ events. It takes as input the superBurst and the

[1]C/C++ programmers should call it according to the prototype:
`int kl2filter_(superBurst*, superBurst*, superCmpEvent*, superCmpEvent*).`

Table 3: Sturucture filled by the 3π charged selection

C	FORTRAN	type	meaning
ana3pic.called	ANA3PIC_CALLED	integer	Non-zero if user_sel3pic was called
ana3pic.flag	ANA3PIC_FLAG	integer	Non-zero if event is selected
ana3pic.ghost	ANA3PIC_GHOST	integer	Non-zero if the first best vtx had a ghost track
ana3pic.ivertex	ANA3PIC_IVERTEX	integer	Index of the best vertex
ana3pic.ikabtrk	ANA3PIC_IKABTRK	integer	Index of the best KABES track
ana3pic.charge	ANA3PIC_CHARGE	integer	K charge
ana3pic.pk	ANA3PIC_PK	float	Reconstructed K momentum
ana3pic.pt2	ANA3PIC_PT2	float	Transverse momentum squared
ana3pic.mass	ANA3PIC_MASS	float	3pi mass
ana3pic.cog1x	ANA3PIC_COG1X	float	COG at DCH1 (x)
ana3pic.cog1y	ANA3PIC_COG1Y	float	COG at DCH1 (y)
ana3pic.cog4x	ANA3PIC_COG4X	float	COG at DCH4 (x)
ana3pic.cog4y	ANA3PIC_COG4Y	float	COG at DCH4 (y)
ana3pic.u	ANA3PIC_U	float	U (this is U0)
ana3pic.v	ANA3PIC_V	float	V (absolute value)
ana3pic.u1	ANA3PIC_U1	float	U from invariant mass of even pions
ana3pic.u2	ANA3PIC_U2	float	U from CM energy of odd pion

Table 4: Sturucture filled by the 3π neutral selection

C	FORTRAN	type	meaning
ana3pin.called	ANA3PIN_CALLED	integer	Non-zero if user_sel3pin was called
ana3pin.flag	ANA3PIN_FLAG	integer	Non-zero if event is selected
ana3pin.ghost	ANA3PIN_GHOST	integer	Non-zero if the first best vtx had a ghost track
ana3pin.ivertex	ANA3PIN_IVERTEX	integer	Index of the best vertex
ana3pin.ikabtrk	ANA3PIN_IKABTRK	integer	Index of the best KABES track
ana3pin.charge	ANA3PIN_CHARGE	integer	K charge
ana3pin.pk	ANA3PIN_PK	float	Reconstructed K momentum
ana3pin.pt2	ANA3PIN_PT2	float	Transverse momentum squared
ana3pin.mass	ANA3PIN_MASS	float	3pi mass
ana3pin.cog1x	ANA3PIN_COG1X	float	COG at DCH1 (x)
ana3pin.cog1y	ANA3PIN_COG1Y	float	COG at DCH1 (y)
ana3pin.cog4x	ANA3PIN_COG4X	float	COG at DCH4 (x)
ana3pin.cog4y	ANA3PIN_COG4Y	float	COG at DCH4 (y)
ana3pin.u	ANA3PIN_U	float	U
ana3pin.v	ANA3PIN_V	float	V (absolute value)

superCmpEvent structures and returns -1 for accepted events or 0 for rejected events.

Since COmPACT 7.1.1, the events selected by this filter are stored in the neutral scmp split.

6 Detailed description of super-compact and compact structure

This section aims at giving more details on the compact variables than the single line listed with the compact structure list. It is divided into three sub-sections: the first one 6.1 describes the super-compact structure; the second one 6.2 describes the compact structure; and the third one 6.3 describes routines that are used to fill the variables (either compact or super-compact or both).

6.1 Super-compact structure

The routine to fill the super-compact structure is *rlib/src/cmp2scmp.c.*

6.1.1 superCmpEvent structure

- struct rndm: for overlayed events.This structure contains information on the random used to overlay (overflows,....).
- nEvt: Event number (0 for overlayed events) .
- trigWord: Bit coded word for trigger, overlaying and filters.
  - bits 0-15.(trigger definition can vary year to year; available from na48 web information).
- timeStamp: The event timestamp in seconds since . . . .
- nTrigBef: coded word for the number of triggers in the previous 20ns (bits 0-7), 60ns (bits 8-15), 100 ns(bits 16-23) and 200ns (bits 24-32) (computed in *rlib/src/getTrigBef.c*).
- timeToPrev: time in timestamp units (25ns units) between the event and the previous trigger (bits 0-15) and the next to previous trigger (bits 16-31).
- SPSPhase: SPS frequency phase to $40MHz$ clock.
- MainsPhase: Phase inside the $50Hz$ cycle.
- QXdNdt: QX intensity between two L2 triggers in Hz.
- struct PUtslice : pattern-unit bits (cf sect. 6.1.9).
- LKRenergy: Total Energy in the Lkr calorimeter (Sum of cluster uncorrected energies).
- HACenergy: Total Energy in the HAC calorimeter

- struct proton: contains description of reconstructed protons from tagger (cf sect. 6.1.7).
- NprotLadder: Number of proton hits on ladder.
- DCHbz: track z position before magnet.
- DCHz: track z position after magnet.
- struct trak (cf sect. 6.1.5).
- struct SCvertex (cd sect. 6.1.6).
- struct FourVertex (cd sect. **??**).
- struct muon (cd sect. 6.1.3).
- struct cluster (cf sect. 6.1.8).
- struct chamber (cf sect. 6.1.10).
- struct DCHcluster: This structure is not filled.
- NovrflwSim: Number of overflows symetrized for charged and neutrals.
- ovrflwSimBef: Time of closest symetrized overflow before the event.
- ovrflwSimAft: Time of closest symetrized overflow after the event.
- tsPrev: Timestamp of previous (L2) trigger.
- tsNext: Timestamp of next (L2) trigger.
- twPrev: Trigger word of previous (L2) trigger.
- twNext: Trigger word of next (L2) trigger.
- nHACcut: Number of HAC clusters with $E > 3\ GeV$ (*getnHACcut in rlib/src/cmp2scmp.c*).
- spareInt[2]: spare variables (integer).
- spareFloat[2]: spare variables (float).

6.1.2 DETstatus sub-structure

This structure holds information describing the sub-detector status (table 5).

6.1.3 muon sub-structure

Valid from version 6.0. From version 6.2 on, this structure is filled at run time according to a new muon-reconstruction routine. The information used to fill the Super-Compact muon structure is taken from the pmuon structure.

6.1.4 pmuon sub-structure

From version 6.2. This structure contains the photo-multiplier hits of the muon veto detector.

TAG	Not filled
AKS	Not filled
AKL	Not filled
DCH	MBOX simulation word (evt->DCHstatus)
HOD	Not filled
HAC	Not filled
LKR	decoding error flag (evt->LKRerrflag)
NHO	Not filled
MUV	decoding error flag (evt->MUVerrflag)
MBX	dead time information: bit 0 ON: MBX alive. (rlib/src/getMBOXdeadTime(bur,evt))
NTR	0: OK; 1: empty; 2: decoding error (rlib/anasrc/nuterr.c))
L3	16 low bits (0-15): Golden Neutral filter bits (see table 2 of Na48 Note 98-28) 16 high bits (16-31): Golden Charged filter bits (see table 3 of Na48 Note 98-28)
LV3Trig	16 low bits(0-15):evt->L3trigword[0] 16 high bits (16-31): evt->L3FilterDownScale
LV3TrigRare	L3 rare filter bits (evt->L3trigword[1])
LV3ABTrig	16 low bits(0-15):evt->L3ONLINEtrigword[0] 16 high bits (16-31): evt->L3Btrigword[0]
LV3ATrigRare	L3 online rare filter bits (evt->L3ONLINEtrigword[1])
LV3BTrigRare	L3 B rare filter bits (evt->L3Btrigword[1])
ChTrEff[10]	Extract from PU info (see table 6) for 10 time slices around the event

Table 5: sevt->DETstatus[0].xxx description

6.1.5 Trak sub-structure

6.1.6 SCvertex sub-structure

Introduced with version 6.0.

6.1.7 Proton sub-structure

Valid up to version 5.2.

6.1.8 Cluster sub-structure

6.1.9 PUtslice sub-structure

From version 6.1 the pattern-unit in Super-Compact is a copy of the pattern-unit in Compact (for the meaning of the bit see the Web address http://na48.web.cern.ch/NA48/private/Trigger/Overview.html). However, in order to save space certain channels have been zeroed and only timeslices 3-5 (counting from zero) are kept. In version 6.2 the following bits are kept:

```
Channel  4 all bits
Channel  5 all bits
Channel  6 all bits
Channel  7 only bit 5 (TON*NAKL)
Channel 12 all bits
Channel 13 all bits
Channel 14 only bits 16-20
Channel 15 all bits
```

6.1.10 Chamber sub-structure

6.1.11 Overflow sub-structure

6.1.12 Accidental sub-structure

6.1.13 Random for overlay sub-structure

6.2 Compact structure

6.2.1 Beam integrator sub-structure

The routine /lib/src/getBINT.c fills the variables for beam monitors integrator. The meaning of the variables is changed in 2001 as listed in table 6.2.1. In particular 8 more words have been added in order to get the samples for the channel with 200 nsec integration time. The 12 samples (10 bits each) are packed in 4 words, the least significant bits being always the earlier of the 3 samples.

6.2.2 Treatment of the charged vertex from version 6.2

From version 6.2 the charged vertex reconstruction is redone at the compact level. This has been introduced to treat vertices made from tracks with the same sign. For each pair of tracks the vertex reconstruction is attempted calling the routine `FD_VTFIT3` in `fuser_vertexntrk.F` (see description below).

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C These routines calculate the vertex coordinates and some other
C parameters related to the tracks attached to this vertex.
C The number of tracks can be 2,3,4 or more ( To use it with at least
C 5 or more tracks , please contact the author )
C The following is a Kalman filter calculation which takes into
```

```
C account multiple scattering in the Kevlar window, He and chambers
C and also measurement errors in the chambers
C It is a generalisation of the code written for 2 tracks which is
C extensively used in the reconstruction of raw data events
C Main routine is FD_VTFIT3
C All calculated values appear in the VERTEXCORR_NTRK array
C Author : J.B.Cheze : November 1999
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

A vertex is saved if the closest distance of approach is less than 10 cm and the time difference between the two tracks is less than 30 ns. Positive and negative track indexes have the same meaning for the opposite sign vertexes: the track sign has to be deduced from $evt- > track[].pq$ (TRACK_PQ).

6.3 Routines used to fill compact and super-compact variables

1) The routine /lib/src/getPuNut.c (from G. Fischer) to detect consecutive triggers; this routine is called when a super-compact event is built and bits are set into $sevt- > CMPstatus$; this routine can also be called from compact (getPuNut(evt) for the C); it returns a bit coded integer:

```
bit 0: No PU data
bit 1: nut2pi0 two timeslices before the trigger
bit 2: nut2pi0 one timeslices before the trigger
bit 3: nut2pi0 at timeslices of the trigger
bit 4: nut2pi0 one timeslices after the trigger
bit 5: nut2pi0 two timeslices after the trigger
```

2)
3)
4)
5)
6)

7 HyperCOmPACT structures

This section provides descriptions for some of the variables in the HyperCOmPACT structures.

7.1 Charged hyperCmpEvent

7.1.1 Treatment of extra tracks/clusters

- Definition of extra cluster:
  $E > 1.5GeV$, separation @LKr R>15cm from each of the 3 tracks belonging to the good vertex. The first 4 extra clusters recorded into hevt->cluster.
  X : **hevt->cluster[i].x;**
  Y : **hevt->cluster[i].y;**
  time : **hevt->cluster[i].time;**
  energy : **hevt->cluster[i].energy;**
- Definition of extra track:
  A track not belonging to the good vertex. The first 4 extra tracks recorded into hevt->cluster structure.
  X @ LKr plane:**hevt->cluster[i].rmsx;**
  Y @ LKr plane: **hevt->cluster[i].rmsy;**
  time : **hevt->cluster[i].dDeadCell;**
  momentum :**hevt->cluster[i].dTrack;**

The numbers of extra tracks and clusters are coded into hevt->flag (like in the previous version of the routine).

7.1.2 hevt->flag (HEVT_FLAG)

The integer variable hevt-¿flag is used to store useful informations about the event. In order to save space, each bit has a different meaning, which is explained in table 7.1.2.

In order to extract the number the function `int get_n_of_tracks(hyperCmpEvent *hevt)` is provided (`GET_N_OF_TRACKS(QE)` in FORTRAN).

7.1.3 hevt->PUpack[3], (HEVT_PUPACK)

The first column is bit number in HEVT_PUPACK, the second is standard channel and bit numbers. HEVT_PUPACK should contain 3 words, corresponding to contents of timeslice 3,4,5. (Exception from this rule: channel N9, there 2 timeslices are packed into single bit).

NB: C indexing used everywhere! should add +1 for fortran indexing!

/* Packed Bit — PU Channel.Bit */

* 0 — 5.0 *

* 1 — 5.1 *

* 2 — 5.2 *

* 3 — 5.3 *

* 4 — 5.4 *

* 5 — 5.5 *

* 6 — 5.6 *
* 7 — 5.7 *
* 8 — 5.8 *
* 9 — 5.9 *
* 10 — 5.10 *
* 11 — 5.12 *
* 12 — 5.13 *
* 13 — 9.10 *
* 14 — 9.11 *
* 15 — 9.12 *
* 16 — 9.17 *
* 17 — 9.18 *
* 18 — 12.1 *
* 19 — 12.3 *
* 20 — 12.4 *
* 21 — 12.5 *
* 22 — 12.13 *
* 23 — 12.18 *
* 24 — 13.1 *
* 25 — 13.2 *
* 26 — 14.0 *
* 27 — 14.1 *
* 28 — 14.16 *
* 29 — 14.17 *
* 30 — 14.18 *
* 31 — 14.20 *

7.2 Neutral hyperCmpEvent

8 Useful informations when running compact and/or super-compact

This sections describes a few informations usefull when running analysis jobs.

8.1 Input/Output flags

In figure 1, it is shown how the output flags -so/-ho... do depending on the type of input the reader gets.

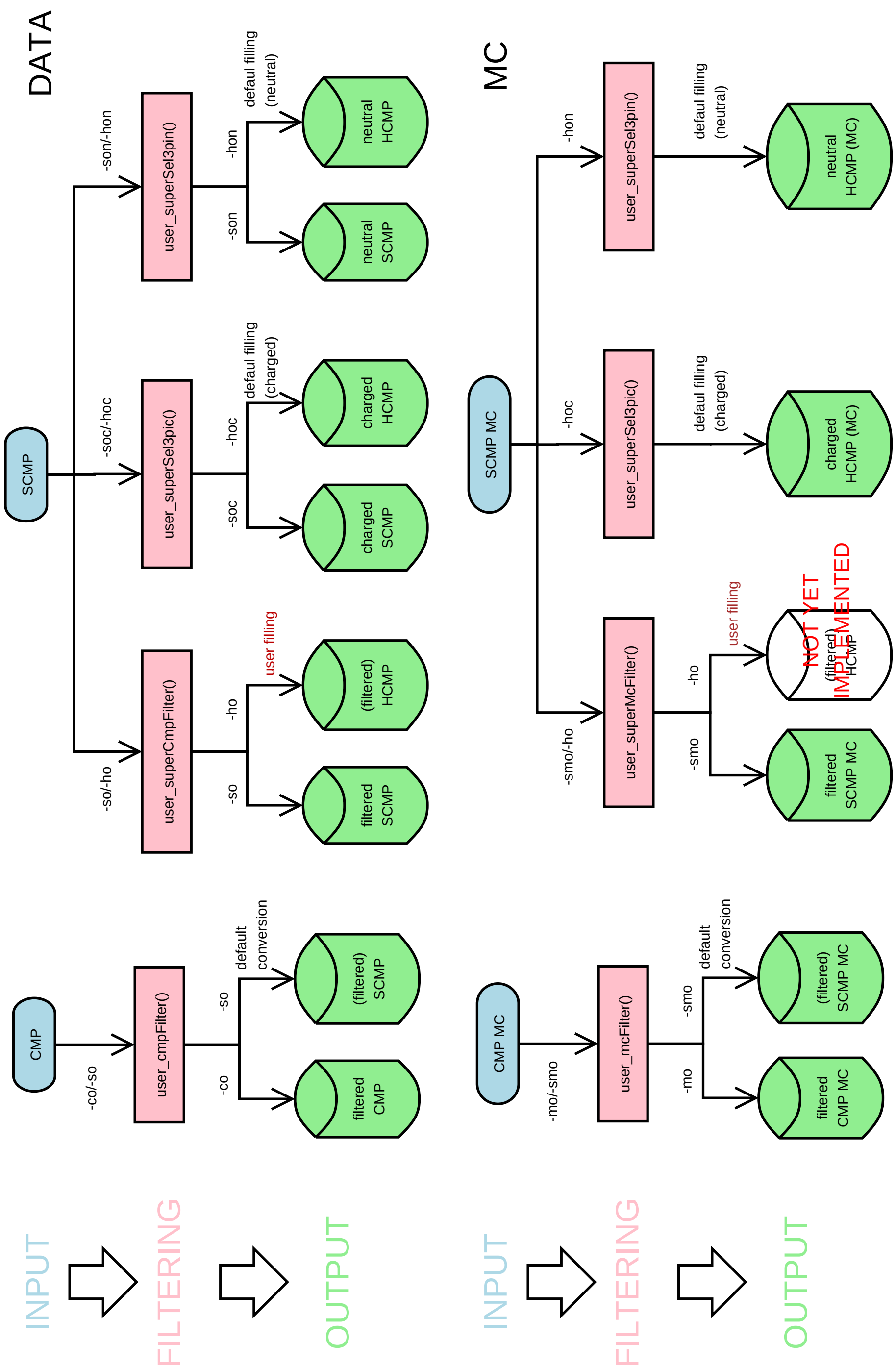

Figure 1: Scheme of the behaviours triggered by the output flags in the COmPACT reader.

8.2

The busrt database (also sometimes called compact database) is automatically accessed from the compact reader and 2 structures are filled: rdb (for run database) et bdb (for burst database); the variable names and descriptions are available at the end of this manual. In the four routines, user_burst.c, user_superBurst.c, fuser_burst.F and fuser_superBurst.F, it is shown how to access the data in the database.

The result of the routine in prepration to decide whether a burst is good or not, will fill the variable bur->dberr (BUR_DBERR) or sbur->dberr (SBUR_DBERR), depending whether the event is of compact or super-compact format.

It is possible to run compact without accessing the compact database (this is sometimes usefull when reading already filtered data or when the database is not yet ready):

```
compact -l input.list -ndb
```

(ndb stands for NoDataBase).

8.3 constants.h file

A set of usefull constants (internal to compact or physical constants like the kaon mass) is available from file lib/inc/constants.h.

8.4 Decoding errors words

The word DCHerrflagPDS: $evt->DCHerrflagPDS$ (EVT_DCHERRFLAGPDS) copy of Q(PDCH+8), the summary of DCH decoding errors (cf Table 8.4).

8.5 Which corrections were applied to this event

The word FlagCorr: $evt->FlagCorr$ (EVT_FLAGCORR) is bit coded and indicates which analysis routines were called (cf Table 10). It is copied to super-compact ($sevt->FlagCorr$ (SEVT_FLAGCORR)).

This information allows to find out *a posteriori* which compact corrections were applied to a given event.

8.6 Event lists

There are 6 *event lists* which contains information for ALL the events in the rawdata files:

8.7 Flags to select analysis routines

- Super Compact has been modified; the structure has changed (and is not backward compatible.
- The analysis routines have been moved to the rlib library for ease on maintenance; i.e. the library can be modified without any intervention from the user; this should make updates easier.
- Control on analysis routines: the variables $bur- > CallAnaRoutine$ (BUR_CALLANAROUTINE) allows the user to decide how to call the analysis routines; this has to be done from the user_burst routine. The routine (f)user_burst.example.c(F) shows how to setup the flags. *The routines badburst, CmpLkrDead (for 1997), CmpTimeOffset, epsy and std_ep, std_rcog, std_ptsq are called antway.*
  - $bur- > CallAnaRoutine = 1$ (the default in compact): let COmPACT decides.
  - $bur- > CallAnaRoutine = 0$: NO analysis routine called.
  - $bur- > CallAnaRoutine = -1$: the user selects the analysis routines by setting the following variables to a non zero value:

```
bur->CallAnaRoutine=-1;  /* the user wants to choose the routines *
bur->tocall.selcharged=1; /* to call selcharged */
bur->tocall.sel2pi0=1;    /* to call sel2pi0 */
bur->tocall.sel3pi0=1;    /* to call sel3pi0 - only if no 2pi0 cand
bur->tocall.bluefield=1;  /* to call correction for Bfield in blue
bur->tocall.lkrpedcor=1;  /* to call lkr pedestal correction (1997)
bur->tocall.hodotime=1;   /* to call hodotime */
bur->tocall.nhodtime=1;   /* to call nhodtime */
bur->tocall.lkrtime=1;    /* to call lkrtime only if there is a 2pi
bur->tocall.tagtime=0;    /* to call tagtime */
bur->tocall.muon_rec=0;   /* to call "new" muon reconstruction (199
bur->tocall.aksflag=1;    /* to call flagging of good AKS events */
bur->tocall.lkrposcor=1;  /* to call lkr position correction (1997)
bur->tocall.lkrsharing=1; /* to call lkr energy sharing between clu
bur->tocall.muon_rec=1;   /* to call "new" muon reconstruction (199
bur->tocall.muon_reject=1;/* to call muon Rejection routines (1997)
bur->tocall.geomcor=1;    /* to call geometry correction routine (1
bur->tocall.newcharged=1; /* to call the new charged analysis
routine - NOT YET */
```

  or

```
C*the user wants to choose the routines
      BUR_CALLANAROUTINE=-1
C*to call selcharged
      BUR_TOCALL_SELCHARGED=1
C*to call sel2pi0
      BUR_TOCALL_SEL2PI0=1
```

```
C*to call sel3pi0 - only if no 2pi0 cand.
      BUR_TOCALL_SEL3PI0=1
C*to call correction for Bfield in blue tube (1997)
      BUR_TOCALL_BLUEFIELD=1
C*to call lkr pedestal correction (1997)
      BUR_TOCALL_LKRPEDCOR=1
C*to call hodotime
      BUR_TOCALL_HODOTIME=1
C*to call nhodtime
      BUR_TOCALL_NHODTIME=1
C*to call lkrtime only if there is a 2pi0 cand.
      BUR_TOCALL_LKRTIME=1
C*to call tagtime
      BUR_TOCALL_TAGTIME=0
C*to call "new" muon reconstruction (1997)
      BUR_TOCALL_MUON_REC=0
C*to call flagging of good AKS events
      BUR_TOCALL_AKSFLAG=1
C* to call lkr position correction (1997)
      BUR_TOCALL_LKRPOSCOR=1
C* to call lkr energy sharing between clusters
      BUR_TOCALL_LKRSHARING=1
C*to call "new" muon reconstruction (1997)
      BUR_TOCALL_MUON_REC=1
C*to call muon rejection (1997)
      BUR_TOCALL_MUON_REJECT=1
C*to call geometry correction routine
      BUR_TOCALL_GEOMCOR=1
C*to call NEW charged event selection routine (NOT YET)
      BUR_TOCALL_NEWCHARGED=1
```

9 Super-compact scaling factors

Important: SuperCOmPACT reduces data volume and uncompression time by using integer variables for all i/o to and from disk files. This is implemented by multiplying floats by 10^n then casting as integers before copying the structure to disk. When reading the file the integer variables are converted to floats and then divided by the same scaling factor. This naturally results in truncation of a certain number of decimal places. It is intended that only non-significant figures are removed.

A list of the scaling factors can be found in the file $/afs/cern.ch/na48/offline/compact/com$ $6.0/lib/compact.x$. The list is also given below with explanations of where these factors are applied:

Here are given the lists of labels for the TS scalers (TS and L2TS).

A 1997 configuration

```
C
C *** L2TS FILE
C
      INTEGER L2TS_NL
      PARAMETER (L2TS_NL=17)
      CHARACTER*32  L2TS_LAB(L2TS_NL)
      DATA L2TS_LAB/'TS_PI+PI-','TS_L1_PI+PI-','TS_L1PP_WDOG',
     +    'TS_QOR', 'TS_PI0PI0', 'TS_NHOD', 'TS_LKRMBIAS',
     +    'TS_EPS_MBIAS', 'TS_PI+PI-PI0', 'TS_MUMUG',
     +    'TS_DALITZ', 'TS_SUSY', 'TS_RanKL/KS',
     +    'TS_L1ON_CNT', 'TS_XLOSTTR', 'TS_FLOSTTR',
     +    'TS_2PI0_NOPK'/


      INTEGER TS_NL
      PARAMETER (TS_NL=93)
      CHARACTER*32  TS_LAB(TS_NL)
      DATA TS_LAB/'TS_Q2*1mu*!AKL', 'TS_Q2*!1mu*!AKL',
     + 'TS_QX*!1mu*!AKL', 'TS_QX*!1mu', 'TS_QX/2',
     + 'TS_QX', 'TS_QX_FTIME_1', 'TS_QX_FTIME_0',
     + 'TS_QX_FT0_to_MB', 'TS_QX_FT1_to_MB',
     + 'TS_MBOX_DEBREQ', 'TS_L1_COD1_to_MB',
     + 'TS_L1_COD2_to_MB', 'TS_L1_COD3_to_MB',
     + 'TS_L1_STRB*L1ON', 'TS_L1_STRB_to_MB',
     + 'TS_<=1TRACK_DCH1', 'TS_2/3TRACKS_DC1',
     + 'TS_>=4TRACKS_DC1', 'TS_OVFLOW_DCH1', 'TS_TOTOR_AKL',
     + 'TS_QX/128', 'TS_Q2*2mu*!AKL', 'TS_TON*!1mu*!AKL',
     + 'TS_TRIG_NO_XOFF', 'TS_TRIG_NOX_WRIT', 'TS_LASER',
     + 'TS_LKR_CAL_(NZS)', 'TS_LKR_CAL_(ZS)', 'TS_HAC_PULSER',
     + 'TS_PMB_PULSER', 'TS_RANDOM_KL', 'TS_RANDOM_KS',
     + 'TS_S2HVspark', 'TS_FULL_TSlost', 'TS_XOFF_TSlost',
     + 'TS_L1_LASER', 'TS_L1_LKRCAL_NZS', 'TS_L1_LKRCAL_ZS',
     + 'TS_L1_HAC_PULS', 'TS_L1_PMB_PULS', 'TS_L1_RAND_KL',
     + 'TS_L1_RAND_KS', 'TS_L1_L1ON_TRANS', 'TS_L1_Q2*1u*!AK',
     + 'TS_L1_Q2*!1u*!AK', 'TS_L1_QX*!1u*!AK', 'TS_L1_QX*!1mu',
     + 'TS_L1_QX/2', 'TS_L1_QX', 'TS_L1_QX_FTIME_1',
     + 'TS_L1_QX_FTIME_0', 'TS_L1_OVF_DCH1', 'TS_L1_>=4TR_DCH1',
     + 'TS_L1_2/3TR_DCH1', 'TS_L1_<=1TR_DCH1', 'TS_L1_TON*!1u*!A',
     + 'TS_L1_Q2*2u*!AK', 'TS_L1_QX/128', 'TS_L1_AKL', 'TS_NT_2PI0',
     + 'TS_NT_LKR_MBIAS', 'TS_NT_2PI0_NOPK', 'TS_NT_ETOT',
     + 'TS_NT_ACCID', 'TS_NT_MUMUGAM', 'TS_NT_MUMUEE',
     + 'TS_NT_DALITZ', 'TS_NT_SUSY_2G', 'TS_NT_FTIME_0',
     + 'TS_NT_FTIME_1', 'TS_NT_TIMPEAK', 'TS_NT_SUSY_3P',
```

```
     + 'TS_CPD_PATTERN', 'TS_Q1*!1mu', 'TS_Q2', 'TS_QOR',
     + 'TS_1MU', 'TS_TON*!1MU', 'TS_Q1*1MU', 'TS_MB_OVFLOW12',
     + 'TS_MB_OVFLOW4', 'TS_MB_TOO_CPLX', 'TS_MB_INTIME',
     + 'TS_MB_ZOK_RARE', 'TS_MB_PI+PI-PI0 ', 'TS_MB_PI+PI-_OK ',
     + 'TS_MB_FATAL', 'TS_MB_FORCE_READ', 'TS_MB_ZOK', 'TS_MB_MASS_OK',
     + 'TS_MB_STRB_TO_TS','TS_NT_ETOT_BIAS'/
```

B 1998 configuration

```
C
C *** L2TS FILE
C
      INTEGER L2TS_NL
      PARAMETER (L2TS_NL=16)
      CHARACTER*32  L2TS_LAB(L2TS_NL)
      DATA L2TS_LAB/'TS_PI+PI-','TS_L1_PI+PI-','TS_L1PP_WDOG',
     +    'TS_CHAR_MBias', 'TS_PI0PI0', 'TS_NHODO','TS_LKRMBIAS',
     +    'TS_3PI0', 'TS_PI+PI-PI0', 'TS_MUMUG',
     +    'TS_DALITZ', 'TS_SUSY', 'TS_RanKL/KS',
     +    'TS_L1ONCNT', 'TS_XLOSTTR', 'TS_FLOSTTR'/


      INTEGER TS_NL
      PARAMETER (TS_NL=92)
      CHARACTER*32  TS_LAB(TS_NL)
      DATA TS_LAB/'TS_Q2*1mu*!AKL', 'TS_Q2*!1mu*!AKL',
     + 'TS_QX*!1mu*!AKL', 'TS_QX*!1mu', 'TS_TOTOR_AKL',
     + 'TS_QX/D', 'TS_QX_FTIME_1', 'TS_QX_FTIME_0',
     + 'TS_QX_FT0_to_MB', 'TS_QX_FT1_to_MB',
     + 'TS_MBOX_DEBREQ', 'TS_L1_COD1_to_MB',
     + 'TS_L1_COD2_to_MB', 'TS_L1_COD3_to_MB',
     + 'TS_L1_STRB*L1ON', 'TS_L1_STRB_to_MB',
     + 'TS_GE1TRACK_DCH1', 'TS_GE2TRACKS_DC1',
     + 'TS_GE4TRACKS_DC1', 'TS_OVFLOW_DCH1',
     + 'TS_QX', 'TS_Q2*2mu*!AKL','TS_TON',
     + 'TS_TRIG_NO_XOFF', 'TS_TRIG_NOX_WRIT','TS_LASER',
     + 'TS_LKR_CAL_(NZS)', 'TS_LKR_CAL_(ZS)','TS_HAC_PULSER',
     + 'TS_PMB_PULSER', 'TS_RANDOM_KL','TS_RANDOM_KS',
     + 'TS_S2HVspark', 'TS_FULL_TSlost','TS_XOFF_TSlost',
     + 'TS_L1_LASER', 'TS_L1_LKRCAL_NZS','TS_L1_LKRCAL_ZS',
     + 'TS_L1_HAC_PULS', 'TS_L1_PMB_PULS','TS_L1_RAND_KL',
     + 'TS_L1_RAND_KS', 'TS_L1_L1ON_TRANS','TS_L1_Q2*1u*!AK',
     + 'TS_L1_Q2*!1u*!AK', 'TS_L1_QX*!1u*!AK','TS_L1_QX*!1u',
     + 'TS_L1_QX/D', 'TS_L1_QX', 'TS_L1_QX_FTIME_1',
```

```
+ 'TS_L1_QX_FTIME_0', 'TS_L1_OVF_DCH1','TS_L1_GE4TR_DCH1',
+ 'TS_L1_GE2TR_DCH1', 'TS_L1_GE1TR_DCH1','TS_L1_TON',
+ 'TS_L1_Q2*2u*!AK', 'TS_L1_TOTOR_AKL','TS_NT_2PI0',
+ 'TS_NT_LKR_MBIAS', 'TS_NT_3PI0','TS_NT_ETOT',
+ 'TS_NT_ACCID', 'TS_NT_MUMUGAM', 'TS_NT_MUMUEE',
+ 'TS_NT_DALITZ', 'TS_NT_SUSY_2G','TS_NT_FTIME_0',
+ 'TS_NT_FTIME_1', 'TS_NT_TIMPEAK','TS_NT_SUSY_3P',
+ 'TS_CPD_PATTERN', 'TS_Q1*!1mu', 'TS_Q2','TS_QOR',
+ 'TS_1MU', 'TS_TON*!1MU', 'TS_Q1*1MU','TS_MB_OVFLOW12',
+ 'TS_MB_OVFLOW4', 'TS_MB_TOO_CPLX','TS_MB_INTIME',
+ 'TS_MB_ZOK_RARE', 'TS_MB_PI+PI-PI0 ','TS_MB_PI+PI-_OK ',
+ 'TS_MB_FATAL', 'TS_MB_FORCE_READ', 'TS_MB_ZOK','TS_MB_MASS_OK',
+ 'TS_MB_STRB_TO_TS','TS_MB_4TRACKS','TS_MB_mbias'/


---------------------------------------------------
```

C 1999 configuration

```
C
C *** L2TS FILE
C
      INTEGER L2TS_NL
      PARAMETER (L2TS_NL=17)
      CHARACTER*32  L2TS_LAB(L2TS_NL)
      DATA L2TS_LAB/'TS_PI+PI-','TS_L1_PI+PI-','TS_L1PP_WDOG',
     +    'TS_CHAR_MBias','TS_PI0PI0','TS_NHODO','TS_LKR_MBIAS',
     +    'TS_3PI0','TS_KE4','TS_MUMUG','TS_DALITZ','TS_MB_4TRACK',
     +    'TS_RanKL/KS','TS_L1CNT','TS_L1ONCNT',
     +    'TS_XLOSTTR','TS_FLOSTTR'/


      INTEGER TS_NL
      PARAMETER (TS_NL=93)
      CHARACTER*32  TS_LAB(TS_NL)
      DATA TS_LAB/'TS_Q2*1mu*!AKL','TS_Q2*!1mu*!AK/D',
     +  'TS_QX*!1mu*!AKL','TS_QX*!1mu','TS_TOTAND_AKL','TS_QX/D',
     +  'TS_Q1*2mu*!AKL','TS_TON','TS_QX_FT0_to_MB',
     +  'TS_QX_FT1_to_MB','TS_MBOX_DEBREQ','TS_L1_COD1_to_MB',
     +  'TS_L1_COD2_to_MB','TS_L1_COD3_to_MB','TS_L1_STRB*L1ON',
     +  'TS_L1_STRB_to_MB','TS_GE1TRACK_DCH1','TS_GE2TRACKS_DC1',
     +  'TS_GE4TRACKS_DC1','TS_OVFLOW_DCH1','TS_DEBU_pulser',
     +  'TS_QX','TS_QX_FTIME_0','TS_QX_FTIME_1',
     +  'TS_TRIG_NO_XOFF','TS_TRIG_NOX_WRIT','TS_LASER',
```

```
     +  'TS_LKR_CAL_(NZS)','TS_LKR_CAL_(ZS)','TS_HAC_PULSER',
     +  'TS_PMB_PULSER','TS_RANDOM_KL','TS_RANDOM_KS',
     +  'TS_S2HVspark','TS_FULL_TSlost','TS_XOFF_TSlost',
     +  'TS_L1_LASER','TS_L1_LKRCAL_NZS','TS_L1_LKRCAL_ZS',
     +  'TS_L1_HAC_PULS','TS_L1_PMB_PULS','TS_L1_RAND_KL',
     +  'TS_L1_RAND_KS','TS_L1_L1ON_TRANS','TS_L1_Q2*1u*!AK',
     +  'TS_L1_Q2*!uAKL/D','TS_L1_QX*!1u*!AK','TS_L1_QX*!1u',
     +  'TS_L1_TOTAND_AKL','TS_L1_QX/D','TS_L1_Q1*2u*!AK',
     +  'TS_L1_TON','TS_L1_OVF_DCH1','TS_L1_GE4TR_DCH1',
     +  'TS_L1_GE2TR_DCH1','TS_L1_GE1TR_DCH1','TS_L1_DEBU_puls',
     +  'TS_L1_QX','TS_L1_QX_FTIME_1','TS_L1_QX_FTIME_0',
     +  'TS_NT_2PI0','TS_NT_LKR_MBIAS','TS_NT_3PI0','TS_NT_ETOT',
     +  'TS_NT_ACCID','TS_NT_MUMUGAM','TS_NT_MUMUGG',
     +  'TS_NT_DALITZ','TS_NT_KE4NT','TS_NT_FTIME_0',
     +  'TS_NT_FTIME_1','TS_NT_TIMPEAK','TS_NT_PI0nunu',
     +  'TS_CPD_PATTERN','TS_Q1*!1mu','TS_Q2','TS_QOR','TS_1MU',
     +  'TS_TON*!1MU','TS_Q1*1MU','TS_MB_OVFLOW12',
     +  'TS_MB_OVFLOW4','TS_MB_TOO_CPLX','TS_MB_INTIME',
     +  'TS_MB_4TRACKS','TS_MB_ZOK_RARE','TS_MB_PI+PI-PI0',
     +  'TS_MB_PI+PI-_OK','TS_MB_FATAL','TS_MB_FORCE_READ',
     +  'TS_MB_ZOK','TS_MB_MASS_OK','TS_MB_STRB_TO_TS'/


--------------------------------------------------
```

D 2001 configuration

```
C
C *** L2TS FILE
C
      INTEGER L2TS_NL
      PARAMETER (L2TS_NL=18)
      CHARACTER*32  L2TS_LAB(L2TS_NL)
      DATA L2TS_LAB/'TS_PI+PI-', 'TS_L1_PI+PI-', 'TS_L1PP_WDOG',
     +     'TS_RADHYP', 'TS_SEMHYP', 'TS_NHODO', 'TS_DELTAS2',
     +     'TS_2PI0', 'TS_3PI0', 'TS_MUMU', 'TS_DALITZ', 'TS_MB_4TRACK',
     +     'TS_MULTI', 'TS_Random', 'TS_L1CNT', 'TS_L1ONCNT',
     +     'TS_XLOSTTR', 'TS_FLOSTTR'/


      INTEGER TS_NL
      PARAMETER (TS_NL=93)
      CHARACTER*32  TS_LAB(TS_NL)
      DATA TS_LAB/'TS_Q2*1mu*!AKL','TS_Q2*!1mu*!AK/D','TS_Q1*!1mu*!AKL',
```

```
     +     'TS_QX', 'TS_TOTAND_AKL', 'TS_Q1/D', 'TS_Q1*2mu', 'TS_TON',
     +     'TS_Q1_FT0_to_MB', 'TS_Q1_FT1_to_MB', 'TS_MBOX_DEBREQ',
     +     'TS_L1_COD1_to_MB', 'TS_L1_COD2_to_MB', 'TS_L1_COD3_to_MB',
     +     'TS_L1_STRB*L1ON', 'TS_L1_STRB_to_MB', 'TS_GE1TRACK_DCH1',
     +     'TS_GE2TRACKS_DC1', 'TS_GE4TRACKS_DC1', 'TS_OVFLOW_DCH1',
     +     'TS_DEBU_pulser', 'TS_Q1', 'TS_Q1_FTIME_0', 'TS_Q1_FTIME_1',
     +     'TS_TRIG_NO_XOFF', 'TS_TRIG_NOX_WRIT', 'TS_LASER',
     +     'TS_LKR_CAL_(NZS)', 'TS_LKR_CAL_(ZS)', 'TS_HAC_PULSER',
     +     'TS_PMB_PULSER', 'TS_RANDOM_KL', 'TS_RANDOM_KS',
     +     'TS_S2HVspark', 'TS_FULL_TSlost', 'TS_XOFF_TSlost',
     +     'TS_L1_LASER', 'TS_L1_LKRCAL_NZS', 'TS_L1_LKRCAL_ZS',
     +     'TS_L1_HAC_PULS', 'TS_L1_PMB_PULS', 'TS_L1_RAND_KL',
     +     'TS_L1_RAND_KS', 'TS_L1_L1ON_TRANS', 'TS_L1_Q2*1u*!AK',
     +     'TS_L1_Q2*!uAKL/D', 'TS_L1_Q1*!1u*!AK', 'TS_L1_QX',
     +     'TS_L1_TOTAND_AKL', 'TS_L1_Q1/D', 'TS_L1_Q1*2u',
     +     'TS_L1_TON', 'TS_L1_OVF_DCH1', 'TS_L1_GE4TR_DCH1',
     +     'TS_L1_GE2TR_DCH1', 'TS_L1_GE1TR_DCH1', 'TS_L1_DEBU_puls',
     +     'TS_L1_Q1', 'TS_L1_Q1_FTIME_1', 'TS_L1_Q1_FTIME_0',
     +     'TS_NT_2PI0', 'TS_NT_LKR_MBIAS', 'TS_NT_3PI0', 'TS_NT_ETOT',
     +     'TS_NT_ACCID', 'TS_NT_EHACLOW', 'TS_NT_MUMUGG',
     +     'TS_NT_DALITZ', 'TS_NT_3PI0_PEAK', 'TS_NT_FTIME_0',
     +     'TS_NT_FTIME_1', 'TS_NT_TIMPEAK', 'TS_NT_PI0nunu',
     +     'TS_CPD_PATTERN', 'TS_Q1*!1mu', 'TS_Q2', 'TS_QOR', 'TS_1MU',
     +     'TS_TON*!1MU', 'TS_Q1*1MU', 'TS_MB_MLAMBDA',
     +     'TS_MB_M>LAMBDA', 'TS_MB_DCH1_DIST', 'TS_MB_INTIME',
     +     'TS_MB_4TRACKS', 'TS_MB_ZOK_RARE', 'TS_MB_P_RATIO',
     +     'TS_MB_PI+PI-_OK', 'TS_MB_FATAL', 'TS_MB_PT', 'TS_MB_ZOK',
     +     'TS_MB_MASS_OK', 'TS_MB_STRB_TO_TS'/

---------------------------------------------------
```

E 2002 configuration

```
C
C *** L2TS FILE
C


      INTEGER L2TS_NL
      PARAMETER (L2TS_NL=18)
      CHARACTER*32  L2TS_LAB(L2TS_NL)
      DATA L2TS_LAB/'TS_PI+PI-', 'TS_L1_PI+PI-', 'TS_L1PP_WDOG',
     +     'TS_RADHYP', 'TS_SEMHYP', 'TS_NHODO', 'TS_DELTAS2',
```

```
+     'TS_2PI0', 'TS_3PI0', 'TS_MUMU', 'TS_DALITZ', 'TS_MB_4TRACK',
+     'TS_MULTI', 'TS_Random', 'TS_L1CNT', 'TS_L1ONCNT',
+     'TS_XLOSTTR', 'TS_FLOSTTR'/



      INTEGER TS_NL
      PARAMETER (TS_NL=93)
      CHARACTER*32  TS_LAB(TS_NL)
      DATA TS_LAB/'TS_Q2*1mu*!AKL','TS_Q2*!1mu*!AK/D','TS_Q1*!1mu*!AKL',
+     'TS_QX', 'TS_TOTAND_AKL', 'TS_Q1/D', 'TS_Q1*2mu', 'TS_TON',
+     'TS_Q1_FT0_to_MB', 'TS_Q1_FT1_to_MB', 'TS_MBOX_DEBREQ',
+     'TS_L1_COD1_to_MB', 'TS_L1_COD2_to_MB', 'TS_L1_COD3_to_MB',
+     'TS_L1_STRB*L1ON', 'TS_L1_STRB_to_MB', 'TS_GE1TRACK_DCH1',
+     'TS_GE2TRACKS_DC1', 'TS_GE4TRACKS_DC1', 'TS_OVFLOW_DCH1',
+     'TS_DEBU_pulser', 'TS_Q1', 'TS_Q1_FTIME_0', 'TS_Q1_FTIME_1',
+     'TS_TRIG_NO_XOFF', 'TS_TRIG_NOX_WRIT', 'TS_LASER',
+     'TS_LKR_CAL_(NZS)', 'TS_LKR_CAL_(ZS)', 'TS_HAC_PULSER',
+     'TS_PMB_PULSER', 'TS_RANDOM_KL', 'TS_RANDOM_KS',
+     'TS_S2HVspark', 'TS_FULL_TSlost', 'TS_XOFF_TSlost',
+     'TS_L1_LASER', 'TS_L1_LKRCAL_NZS', 'TS_L1_LKRCAL_ZS',
+     'TS_L1_HAC_PULS', 'TS_L1_PMB_PULS', 'TS_L1_RAND_KL',
+     'TS_L1_RAND_KS', 'TS_L1_L1ON_TRANS', 'TS_L1_Q2*1u*!AK',
+     'TS_L1_Q2*!uAKL/D', 'TS_L1_Q1*!1u*!AK', 'TS_L1_QX',
+     'TS_L1_TOTAND_AKL', 'TS_L1_Q1/D', 'TS_L1_Q1*2u',
+     'TS_L1_TON', 'TS_L1_OVF_DCH1', 'TS_L1_GE4TR_DCH1',
+     'TS_L1_GE2TR_DCH1', 'TS_L1_GE1TR_DCH1', 'TS_L1_DEBU_puls',
+     'TS_L1_Q1', 'TS_L1_Q1_FTIME_1', 'TS_L1_Q1_FTIME_0',
+     'TS_NT_2PI0', 'TS_NT_LKR_MBIAS', 'TS_NT_3PI0', 'TS_NT_ETOT',
+     'TS_NT_ACCID', 'TS_NT_EHACLOW', 'TS_NT_MUMUGG',
+     'TS_NT_DALITZ', 'TS_NT_3PI0_PEAK', 'TS_NT_FTIME_0',
+     'TS_NT_FTIME_1', 'TS_NT_TIMPEAK', 'TS_NT_PI0nunu',
+     'TS_CPD_PATTERN', 'TS_Q1*!1mu', 'TS_Q2', 'TS_QOR', 'TS_1MU',
+     'TS_TON*!1MU', 'TS_Q1*1MU', 'TS_MB_MLAMBDA',
+     'TS_MB_M>LAMBDA', 'TS_MB_DCH1_DIST', 'TS_MB_INTIME',
+     'TS_MB_4TRACKS', 'TS_MB_ZOK_RARE', 'TS_MB_P_RATIO',
+     'TS_MB_PI+PI-_OK', 'TS_MB_FATAL', 'TS_MB_PT', 'TS_MB_ZOK',
+     'TS_MB_MASS_OK', 'TS_MB_STRB_TO_TS'/
```

F 2003 configuration

L1TS Scalers

id	label	id	label	id	label
1	TS_CH_KE2-PRE/D3	32	TS_RANDOM_KL	63	TS_NT_PEAK
2	TS_Q1/D1	33	TS_RANDOM_KS	64	TS_NT_Etot
3	TS_CH_KU2-PRE/D4	34	TS_S2HVspark	65	TS_NT_Etot_short
4	TS_Q2*!AKL	35	TS_FULL_TSlost	66	TS_NT_KMU3
5	TS_Q2*!AKL_d	36	TS_XOFF_TSlost	67	TS_NT_KMU2
6	TS_Q1/D2	37	TS_L1_LASER	68	TS_NT_KE2-PRE
7	TS_CH_KE2-PRE	38	TS_L1_LKRCAL_NZS	69	TS_NT_NOPEAK
8	TS_CH_KMU2-PRE	39	TS_L1_LKRCAL_ZS	70	TS_NT_FTIME_0
9	TS_Q1+Q2_FT0_MB	40	TS_L1_HAC_PULS	71	TS_NT_FTIME_1
10	TS_Q1+Q2_FT1_MB	41	TS_L1_PMB_PULS	72	TS_NT_TIMEPEAK
11	TS_MBOX_DEBREQ	42	TS_L1_RAND_KL	73	TS_NT_Zvtx
12	TS_L1_COD1_to_MB	43	TS_L1_RAND_KS	74	TS_Q1
13	TS_L1_COD2_to_MB	44	TS_L1_L1ON_TRANS	75	TS_Q1*!Q2
14	TS_L1_COD3_to_MB	45	TS_L1_KE2-PRE/D3	76	TS_Q1*QAND
15	TS_L1_STRB*L1ON	46	TS_L1_Q1/D1	77	TS_1MU
16	TS_L1_STRB_to_MB	47	TS_L1_KMU2-PR/D4	78	TS_TON
17	TS_1trk_loose	48	TS_L1_Q2*!AKL	79	TS_Q2
18	TS_2-3trk_loose	49	TS_L1_Q2*!AKL_d	80	TS_MB_OVFLOW1-2
19	TS_HI_multipl.	50	TS_L1_Q1/D2	81	TS_MB_OVFLOW4
20	TS_OVFLOW_DCH1	51	TS_L1_KE2-PRE	82	TS_MB_TOO_CPLX
21	TS_Random_L1	52	TS_L1_KMU2-PRE	83	TS_MB_IN_TIME
22	TS_Q1+Q2	53	TS_L1_OVF_DCH1	84	TS_MB_MBX-2VTX
23	TS_Q1+Q2_FTIME_0	54	TS_L1_HI_mult	85	TS_MB_1TRK-3
24	TS_Q1+Q2_FTIME_1	55	TS_L1_2-3trk_loo	86	TS_MB_Zok_rare
25	TS_TRIG_NO_XOFF	56	TS_L1_1trk_loose	87	TS_MB_MBX-1VTX
26	TS_TRIG_NOX_WRIT	57	TS_L1_Random_L1	88	TS_MB_1TRK-2
27	TS_LASER	58	TS_L1_Q1+Q2	89	TS_MB_FATAL
28	TS_LKR_CAL_(NZS)	59	TS_L1_Q1+Q2_FT_1	90	TS_MB_FORCE_R/O
29	TS_LKR_CAL_(ZS)	60	TS_L1_Q1+Q2_FT_0	91	TS_MB_ZFAKE
30	TS_HAC_PULSER	61	TS_NT_COG	92	TS_MB_1TRK-P
31	TS_PMB_PULSER	62	TS_NT_LKR_MBIAS	93	TS_MB_STRB_TO_TS

L2TS Scalers

id	label	id	label	id	label
1	TS_MB-2VTX	7	TS_MB-1TRK-2	13	TS_2BODY-PRE
2	TS_MB-1VTX	8	TS_MB-ZFAKE	14	TS_Random
3	TS_L1PP_WDOG	9	TS_C-MBIAS	15	TS_L1CNT
4	TS_C-PRE	10	TS_N-MBIAS	16	TS_L1ONCNT
5	TS_MB-1TRK-P	11	TS_N-PRE	17	TS_XLOSTTR
6	TS_MB-1TRK-3	12	TS_KMU3-PRE	18	TS_FLOSTTR

NOTE: in C you have to use (id-1) because the arrays are 0-based.

G Common blocks description

Here is a description, as complete as possible, prepared by G. Bocquet, of the common blocks from which data were transfered from the raw0xx program to compact. This descr

```
************************************************************************
* COMMON /DCHSAC_CMPBLK/     rsa car file
************************************************************************
      INTEGER CMAXHIT            ! maximum number of hits
      INTEGER MAXTRACK           ! maximum number of tracks
      INTEGER NDCH               ! number of chambers
      PARAMETER (CMAXHIT=64)
      PARAMETER (MAXTRACK=40)
      PARAMETER (NDCH=4)
      INTEGER NHITS              ! nb of hits/view(1->4 ch1,5->8 ch2,.)
      INTEGER NHT                ! nb of hits on track
      INTEGER JPLANE             ! planes of track hits
      INTEGER JWIRE              ! wires of track hits
      REAL JTIME_C               ! drift time of track hits
      REAL JDIST_C               ! drift distance of track hits
      REAL JCOOR_C               ! coordinate of track hits
      REAL XDCH                  ! track space points in drift chambers
      INTEGER NOVERFL,OVFLBIT    ! nb of overflows and overflow bits
      INTEGER IEFF               ! efficiency

      COMMON /DCHSAC_CMPBLK/ NHITS(16),NHT(MAXTRACK),
     +     JPLANE(MAXTRACK,CMAXHIT),
     +     JWIRE(MAXTRACK,CMAXHIT),JTIME_C(MAXTRACK,CMAXHIT),
     +     JDIST_C(MAXTRACK,CMAXHIT),JCOOR_C(MAXTRACK,CMAXHIT),
     +     XDCH(MAXTRACK,NDCH,3),NOVERFL,OVFLBIT,
     +     IEFF(MAXTRACK,2)



************************************************************************
* COMMON /DCHCLU_CMPBLK/     rsa car file
************************************************************************
      INTEGER NCLUSTER,MAXCLUSTER
      PARAMETER (MAXCLUSTER=64)
      INTEGER DCH_CLUS
      COMMON/DCHCLU_CMPBLK/NCLUSTER,DCH_CLUS(3,MAXCLUSTER)
```

```
c      DCH_CLUS(1,NCLUSTER) =  view number ( 1:16)
c      DCH_CLUS(2,NCLUSTER) =  measured coodinate
c      DCH_CLUS(3,NCLUSTER) =  cluster time (0.:100.ns)
```

```
************************************************************************
* COMMON /OVRFLWSIM_CMPBLK/     res car file
************************************************************************
      integer NSOVF       ! simulated overflow condition with > 5 hits
      real    TOVF_CLOSEB     ! ovf closest to zero, but Before 0
      real    TOVF_CLOSEA     ! ovf closest to zero, but After 0
                              ! these 2 variables are obtained looping
                              ! on the chambers but number 3

      COMMON /OVRFLWSIM_CMPBLK/ NSOVF,TOVF_CLOSEB,TOVF_CLOSEA
```

```
************************************************************************
* COMMON /OVLRND_CMPBLK/        ovl car file
************************************************************************
      Integer  RNDty          ! random type: 1=Ks, 2=Kl (from PRE bank)
      Integer  RNDts          ! timestamp (from PRE bank)
      Integer  RNDrun         ! run# of the random used from (PRE bank)
      Integer  RNDbur         ! burst# of the random used(from PRE bank)
      Integer  RNDused        ! #times random used so far(from PRE bank)
      Integer  PDSused        !
      Real     RNDsps         ! sps phase  (from PSCA bank)
      Real     RND50          ! main phase (from PSCA bank)
      Integer  RND_ROVF(4)    ! ring overflow info
      Real     RND_TOVF(64)   ! TDC overflow info
      REAL     RNDKlmon_DNDT  ! Kl monitor intensity (from PSCA bank)
      REAL     RNDKsmon_DNDT  ! Ks monitor intensity (from PSCA bank)
      REAL     RNDTagmon_DNDT ! tagger monitor intensity(from PSCA)
      REAL     RNDQx_DNDT     ! Qx intensity (from PSCA bank)
      REAL     RNDAKs_DNDT    ! AKS intensity (from PSCA bank)

      COMMON /OVLRND_CMPBLK/ RNDty, RNDts, RNDrun, RNDbur, RNDused,
     +                       RNDsps, RND50, RND_TOVF, RND_ROVF,
     +                       PDSused, RNDKlmon_DNDT, RNDKsmon_DNDT,
     +                       RNDTagmon_DNDT, RNDQx_DNDT, RNDAKs_DNDT
```

```
**********************************************************************
* COMMON /AKS_REC/     rec car file
**********************************************************************
      integer nchann
      parameter (nchann=5)
      real aksrec(nchann,4)
      real aks_eff(2)
      integer iflag_aksrec
C
      COMMON /aks_REC/aksrec,aks_eff,iflag_aksrec
C
C     aksrec(n,1) = reconstructed number of mips for counter n
C     aksrec(n,2) = reconstructed time for counter n
C     aksrec(n,3) = pmb reconstruction error flag for counter n
C     aksrec(n,4) = reconstructed second time for counter n
C     aks_eff(1) = efficiency of counter n.4
C     aks_eff(2) = efficiency sigma of counter n.4
C     iflag_aksrec = not yet implemented
```

H SQLITE database variables

H.1 Variables on a Compact Structure

```
 - Bad bursts for each detector:       in BadBurst   structure
    defined on a burst or run basis
 - Time offsets per Run:               in TimeOffset structure
```

H.2 Variables on COMMON ABCOG_PARAMS

```
  ABCOG\_PARAMS\_ALPHA       Alpha parameter
  ABCOG\_PARAMS\_ALPHA_COEFF Coefficient used for alpha determination
  ABCOG\_PARAMS\_BETA        Beta parameter
  ABCOG\_PARAMS\_BETA_COEFF  Coefficient used for beta determination
  ABCOG\_PARAMS\_MKP         Mass for K+
  ABCOG\_PARAMS\_MKPERR      Error on mass shift for K+
  ABCOG\_PARAMS\_MKN         Mass shift for K-
  ABCOG\_PARAMS\_MKNERR      Error on mass shift for K-
  ABCOG\_PARAMS\_COGX1P      X of COG at DCH1 for pos. track
  ABCOG\_PARAMS\_COGY1P      Y of COG at DCH1 for pos. track
```

```
ABCOG\_PARAMS\_COGX1N      X of COG at DCH1 for neg. track
ABCOG\_PARAMS\_COGY1N      Y of COG at DCH1 for neg. track
ABCOG\_PARAMS\_COGX4P      X of COG at DCH4 for pos. track
ABCOG\_PARAMS\_COGY4P      Y of COG at DCH4 for pos. track
ABCOG\_PARAMS\_COGX4N      X of COG at DCH4 for neg. track
ABCOG\_PARAMS\_COGY4N      Y of COG at DCH4 for neg. track
ABCOG\_PARAMS\_STATUS      =0 if query of db ok, =1 if problems
ABCOG\_PARAMS\_pkp         Momentum for positive kaon beam
ABCOG\_PARAMS\_pkdxdzp     dxdz for positive kaon beam
ABCOG\_PARAMS\_pkdydzp     dydz for positive kaon beam
ABCOG\_PARAMS\_pkxoffp     x offset for positive kaom beam at z=0
ABCOG\_PARAMS\_pkyoffp     y offset for positive kaom beam at z=0
ABCOG\_PARAMS\_pkm         Momentum for negative kaon beam
ABCOG\_PARAMS\_pkdxdzm    dxdz for negative kaon beam
ABCOG\_PARAMS\_pkdydzm    dydz for negative kaon beam
ABCOG\_PARAMS\_pkxoffm    x offset for negative kaom beam at z=0
ABCOG\_PARAMS\_pkyoffm    y offset for negative kaom beam at z=0
```

The syntax of the beam parameter variables listed above is for FORTRAN user, for C the corresponding variables can be deduced in the following way:

ABCOG_PARAMS_pkdxdzm → abcog_params.pkdxdzm

bit	1997 data	1998+1999 data
0	chan 0-bit 4 HOD-L1in Qx/2	chan 0-bit 5 HOD-L1in Qx/D
1	chan 0-bit 5 HOD-L1in Qx	chan 1-bit 9 HOD-L1in Qx
2	chan 0-bit 6 HOD-L1in FT0	chan 1-bit10 HOD-L1in FT0
3	chan 0-bit 7 HOD-L1in FT1	chan 1-bit11 HOD-L1in FT1
4	chan 3-bit11 NUT-L1in Etot	chan 3-bit11 NUT-L1in Etot
5	chan 3-bit18 NUT-L1in FT0 NUT	chan 3-bit18 NUT-L1in FT0 NUT
6	chan 3-bit19 NUT-L1in FT1 NUT	chan 3-bit19 NUT-L1in FT1 NUT
7	chan 4-bit 4 SUB1-L2in Qx/2	chan 4-bit 5 SUB1-L2in Qx/D
8	chan 4-bit 5 SUB1-L2in Qx	chan 4-bit13 SUB1-L2in Qx
9	chan 4-bit 6 SUB1-L2in FT0	chan 4-bit15 SUB1-L2in FT0
10	chan 4-bit 7 SUB1-L2in FT1	chan 4-bit14 SUB1-L2in FT1
11	chan 6-bit 3 SUB3-L2in Etot	chan 6-bit 3 SUB3-L2in Etot
12	chan14-bit 0 L1 out to MBOX FT0	chan14-bit 0 L1 out to MBOX FT0
13	chan14-bit 1 L1 out to MBOX FT1	chan14-bit 1 L1 out to MBOX FT1
14	chan14-bit16 L1 out to MBOX Trg bit 0 to MBX	chan14-bit16 L1 out to MBOX Trg bit 0 to
15	chan14-bit17 L1 out to MBOX Trg bit 1 to MBX	chan14-bit17 L1 out to MBOX Trg bit 1 to
16	chan14-bit18 L1 out to MBOX Trg bit 2 to MBX	chan14-bit18 L1 out to MBOX Trg bit 2 to
17	chan14-bit19 L1 out to MBOX Strobe to MBX	chan14-bit19 L1 out to MBOX Strobe to M
18		chan14-bit20 L1 out to MBOX L1on
19		chan12-bit10 L1 out: 2tracks
20		chan12-bit 5 L1 out: QX/D
21		chan12-bit13 L1 out: QX
22		chan12-bit15 L1 out: FT0
23		chan12-bit14 L1 out: FT1
24		chan13-bit 0 L1 out: Etot
25		chan 4-bit10 SUB1-L2in: 2tracks
26		chan 5-bit 2 L2 in: TCPLX
27		chan 5-bit 3 L2 in: IN-TIME
28	chan 5-bit 9 L2 in: Pi+Pi- OK	chan 5-bit 9 L2 in: Pi+Pi- OK
29		chan 5-bit10 L2 in: fatal
30	chan 1-bit 9 DCH-L1in Qx/32	chan10-bit24 AKS PU
31	chan 4-bit14 SUB1-L2in Qx/32	

Table 6: sevt->DETstatus[0].ChTrEff[its] description

name x=l/s	meaning	Integration time <2001	 2001
evt-> beamIntKx.Integ2us (BEAMINTKX_INTEG2US)	average	2μsec	1 μsec
evt-> beamIntKx.Integ15us(BEAMINTKX_INTEG15US)	average	15μsec	200 nsec
evt-> beamIntKx.Integ30us(BEAMINTKX_INTEG30US)	average	30μsec	3 μsec
evt-> beamIntKx.Integ60us(BEAMINTKX_INTEG60US)	average	60μsec	15 μsec
evt-> beamIntKx.Qual2us (BEAMINTKX_QUAL2US)	rms	2μsec	1 μsec
evt-> beamIntKx.Qual15us (BEAMINTKX_QUAL15US)	rms	15μsec	200 nsec
evt-> beamIntKx.Qual30us (BEAMINTKX_QUAL30US)	rms	30μsec	3 μsec
evt-> beamIntKx.Qual60us (BEAMINTKX_QUAL60US)	rms	60μsec	15 μsec
evt-> beamIntKx.packsampl[1-4] (BEAMINTKX_PACKSAMP(1-4))	samples		200 nsec

Table 7: Beam intensity integrators meanings

bit	mask	mnemonic	description
0	0x1	—HF_NEUTRAL—	$\pi^{\pm}\pi^0\pi^0$ event
1	0x2	—HF_LKRDNSC—	the event was recorded without reading the LKR
2-5	0x3c	—HF_NTRACKS—	Number of tracks
6	0x40	—HD_GHOSTTR—	the first best vertex was discarded because of a ghost track
7			Empty
8-11	0xf00	—HF_DTSTAMP—	Distance to previous event (if <= 15 timestamps)
12-15	0xf000	—HF_CLUSTER—	Number of non-assoc. LKR clusters
16-19	0xf0000	—HF_CLINTIM—	Non-assoc. LKR clusters in $\pm$ 15 ns window
20-23	0xf00000	—HF_TRINTIM—	Extra tracks in $\pm$ 15 ns window

Table 8: Meaning of the bits of $hevt->flag$ (HEVT_FLAG).

<2002	Bit	meaning	verdict	2002	Bit	meaning
	0	Data corruption	no-fatal	0	Tstamp difference between headers	no-fatal
	1	Internal mixing	no-fatal	1	Wrong word count in one plane	no-fatal
	2	Global Mixing	no-fatal	2	Token time out	no-fatal
	3	Data corruption	no-fatal	3	Data ready time out	no-fatal
	4	Data corruption	no-fatal	4	Tstamp unrecovered mismatch	no-fatal
	5	Internal mixing	no-fatal	5	Wrong number of headers	no-fatal
	6	Data corruption	no-fatal	6	Header-CSC mismatch	no-fatal
	7			7		
	8			8	First word is not a header	no-fatal
	9	Data corruption	no-fatal	9	Plane number mismatch	no-fatal
	10	Global Mixing	no-fatal	10	TStamp differs from most frequent one	no-fatal
	11	Data corruption	no-fatal	11	Chamber number mismatch	no-fatal

Table 9: DCH decoding error bits description

Bit set	analysis routine called
0	fuser_selcharged
1	fuser_sel2pi0
2	fuser_sel3pi0
3	fuser_bluefield
4	fuser_lkrpedcor
5	fuser_lkrposcor
6	fuser_lkrsharing
7	fuser_hodotime
8	fuser_nhodtime
9	fuser_lkrtime
10	fuser_tagtime
11	fuser_aksflag
12	muon_rec
13	muonReject
14	user_GeomCor
15	fuser_newcharged
16	user_magnetcorr
17	muon_trackrec (uncorr. tracks)
18	muon_vertexrec
19	fuser_lkrcalcor
20	subset1 of fuser_lkrcalcor
21	subset2 of fuser_lkrcalcor
21	subset3 of fuser_lkrcalcor
23	fuser_sel2gam
24	muon_trackrec (corr. tracks)
25	fuser_lkrcalhi2k

Table 10: Correspondance bit in FlagCorr - analysis routines

$evt- > TrigWordL(i)$ EVT_TRIGWORDL(i)	Trigger words	
$evt- > TimeStampL(i)$ EVT_TIMESTAMPL(i)	Time stamps	
$evt- > DCHDecErrorL(i)$ EVT_DCHDECERRORL(i)	DCHerrflagPDS (see table above)	
$evt- > LKRHADecErrorL(i)$ EVT_LKRHACDECERRORL(i)	=Q(LPLKR+9) if=0 decoding OK	
	bit set	
	0	timestamp not consistent among the links
	1	trigger word not consistent among the links
	2	local event number not consistent among the links
	3	decoding error: recover missing trailer
	4	decoding errors
	5	No data (no LKR event header)
	6	NHitcell=0
	7	Too much data
	8	Not enough data
	9	decoding error from DCP header
	10	decoding error from channel header
	11	decoding error from trailer
	14	HAC decoding called
	15	HAC decoding problem
$evt- > NeutralInfoL(i)$ EVT_NEUTRALINFOL(i)	bit set	
	0	Routine called
	1	downscaled event
	2	KS candidate
	3	KS candidate flagged
	4	KL candidate
	5	KL candidate passing all physics cuts
	6	no KS or no KL found
	7	Pb. with REC or DCHREC or RDTK or VTX banks
$evt- > ChargedInfoL(i)$ EVT_CHARGEDINFOL(i)	= iflag1(L3B filter) + 16*iflag2(golden filter)	
	iflag1	
	1	routine called
	2	$\leq$ 4 clusters-evt kept
	3	$>$ 4clusters with one comb. in time-evt kept
	4	no comb. with geq5 clusters in time-evt kept
	5	geq5 clu. in time and fullfilling $2\pi^0$ cut.
	iflag2	
	1	routine called
	2	all banks OK
	3	geq 4 clusters
	4	geq 4 clusters in time
	5	above energy cut
	6	+ 1 comb. fullfilling $E_{tot} > cut$
	7	+ fullfilling cog cut
	8	+ fullfilling $c.\tau$ cut
	9	+ fullfilling π^0 mass cut

Table 11: Description of the content of the complete lists(i.e. one entry per recorded event)

Scaling factor	Usage
const SCF_EK = 1000;	kaon energy
const SCF_SPSPHASE = 1000;	SPS phase wrt to 40MHz clock
const SCF_EVTTIME = 1000;	event time
const SCF_MREC = 100000;	reconstructed mass
const SCF_VTXX = 100;	charged vertex: x-coord.
const SCF_VTXY = 100;	charged vertex: y-coord.
const SCF_VTXZ = 100;	charged vertex: z-coord.
const SCF_VTXCDA = 100;	charged vertex: cda
const SCF_VTXTIME = 1000;	charged vertex: time
const SCF_VTXPTSQ = 10000000;	charged vertex: p_t^2
const SCF_AKSA = 100;	z AKS
const SCF_ESC = 10000;	energy scale
const SCF_MPI0 = 100000;	π^0 mass
const SCF_RELLI = 1000;	ellipse number
const SCF_PI0MERR = 100000;	error on π^0 mass
const SCF_TAGTIME = 1000;	tagger proton time
const SCF_AKSTIME = 1000;	AKS hit time
const SCF_AKLTIME = 1000;	AKL hit time
const SCF_TRKP = 10000;	charged track momentum
const SCF_TRKQL = 1000;	track quality
const SCF_TRKX = 1000;	track x-coord. before/after magnet
const SCF_TRKY = 1000;	track y-coord. before/after magnet
const SCF_TRKZ = 100;	track z-coord. before/after magnet
const SCF_TRKDXDZ = 100000;	track slope before/after magnet
const SCF_TRKDYDZ = 100000;	track slope before/after magnet
const SCF_TRKTIME = 100;	track time
const SCF_TRKDDEAD = 1000;	d(cm) to closest dead cell
const SCF_HODTIME = 1000;	hodoscope hit time
const SCF_NHODTIME = 1000;	neutral hodoscope hit time
const SCF_LKRE = 1000;	Lkr total/cluster energy
const SCF_LKRX = 1000;	Lkr cluster x-coord.
const SCF_LKRY = 1000;	Lkr cluster y-coord.
const SCF_LKRTIME = 1000;	Lkr cluster time
const SCF_ESPY = 1000;	Neutral Trigger energy
const SCF_EOVP = 1000;	E/p
const SCF_HACE = 1000;	Hac total/cluster energy
const SCF_HACBF = 1000;	Hac E_{back}/E_{front}
const SCF_MUVTIME = 1000;	Muon veto time
const SCF_OVFLTIME = 1000;	Overflow time
const SCF_OVFLX = 100;	
const SCF_OVFLY = 100;	
const SCF_KSMTIME = 1000;	KSM time
const SCF_DTARG = 1000;	D_{target}
const SCF_PTPRIME = 1000000;	$\tilde{P}_\perp^2$

I C Interface

All the event variables are stored in an “Event” structure. A pointer called “evnt” is passed to the relevant user routines and this points to the Event structure containing the data. As usual with C variables all these names are case sensitive. From the release of version 2.0 there are now two event structures: one for normal COmPACT events “cmpEvent” and one for Ke3 events “ke3Event”.

I.1 Calling analysis routines

Some analysis routines are now in compact. Important information valid for both the fortran and the C interfaces have been given in section 4. The way to call the routines is given as an example in *user_cmpEvent.example.c*.

J COmPACT Structures

J.1 NUTkmu3

Variable	Type	Description
kmu3->nut.timestamp	int	NUT time stamp = IQ(LPNUT+1)
kmu3->nut.maxE	int	1000*Energy at IPC (GeV) = IQ(LPNUT+6)
kmu3->nut.maxCOG	int	1000*COG at IPC (cm) = IQ(LPNUT+7)
kmu3->nut.maxd	int	1000*d at IPC (cm) = IQ(LPNUT+8)
kmu3->nut.maxz	int	1000*z at IPC (cm) = IQ(LPNUT+9)
kmu3->nut.maxl	int	1000*l at IPC (lifetime units) = IQ(LPNUT+10)
kmu3->nut.maxtf	int	1000*time corr. at IPC (ns) = IQ(LPNUT+11)
kmu3->nut.spy_x[64]	int	spy channels (X view) = IQ(LPNUT+28+i)
kmu3->nut.spy_y[64]	int	spy channels (Y view) = IQ(LPNUT+92+i)
kmu3->nut.spy_xcbit[6]	int	spy ctrl bits (X view) = IQ(LPNUT+162+i)
kmu3->nut.spy_ycbit[6]	int	spy ctrl bits (Y view) = IQ(LPNUT+168+i)

J.2 RNDMsummary

Variable	Type	Description
evt->rndm.type	int	type of random used(KS=1, KL=2) = Q(LPRE+1)
evt->rndm.timestamp	int	timestamp of random used = Q(LPRE+3)
evt->rndm.run_burst	int	Not used in versions $\geq$ 3.4
evt->rndm.run	int	Run Number of random used = Q(LPRE+9)

Variable	Type	Description
evt->rndm.burst	int	Burst time stamp of random used = Q(LPRE+10)
evt->rndm.Nused	int	Nb of times RNDM evt was used so far=Q(LPRE+8)
evt->rndm.SPSphase	float	SPS phase of random evt
evt->rndm.mainphase	float	main(50Hz) phase of random evt
evt->rndm.PDSused	int	one bit/detector overlayed as in LQ(LPRE)
evt->rndm.tovrflw[64]	float	time of nearest ovflw to rndm evt
evt->rndm.rovrflw[4]	int	ring buffer overflow for each chamber
evt->rndm.KLmondNdt	float	intensity in KLmon for random event (cf doc230698)
evt->rndm.KSmondNdt	float	intensity in KSmon for random event
evt->rndm.TagmondNdt	float	intensity in TAGGERmon for random event
evt->rndm.QXdNdt	float	intensity in QX for random event
evt->rndm.AKSdNdt	float	intensity in AKS for random event

J.3 L3TRIGhist

Variable	Type	Description
evt->trhist[*ievt*].trigword	int	Trigger word for previous event ievt
evt->trhist[*ievt*].tstamp_diff	int	Timestamp difference from current event

J.4 EVTtimestamp

Variable	Type	Description
evt->tstamp[*n*].timestamp	int	Time stamp which is different
evt->tstamp[*n*].source	int	source of timestamp

J.5 PMBscaler

Variable	Type	Description
evt->scaler.n	int	counts this timeslice =Q(PSCA+x+2) (cf doc 230698)
evt->scaler.dndt	float	difference/timestamp diff. =Q(PSCA+x+3)

J.6 PMBtimeslice

Variable	Type	Description
evt->trig[*tslice*].chan[16]	int	16 channels * 24 bits=Q(PCAT+19+x)

J.7 TAGhit

Variable	Type	Description
evt->tag[*chan*].hit[*hit*].ampl	float	amplitude of hit = Q(RTAG+x+1)
evt->tag[*chan*].hit[*hit*].time	float	time of hit = Q(RTAG+x+3)
evt->tag[*chan*].hit[*hit*].chi2	float	χ^2 of hit = Q(RTAG+x+5)
evt->tag[*chan*].hit[*hit*].status	int	status code of hit = Q(RTAG+x+6)

J.8 TAGchannel

Variable	Type	Description
evt->tag[*chan*].n	int	channel number=Q(RTAG+n+1)
evt->tag[*chan*].Nhit	u_int	Number of tagger hits in this channel

Variable	Type	Description
evt->tag[*chan*].hit[15]	TAGhit	tagger hits in this channel

J.9 KSMhit

Variable	Type	Description
evt->ksm[*hit*].counter	int	counter number(1-8) = Q(PKSM+x+2)
evt->ksm[*hit*].pheight	float	pulse height(GeV) = Q(PKSM+x+3)
evt->ksm[*hit*].time	float	time of hit(ns) = Q(PKSM+x+4)
evt->ksm[*hit*].pdsflag	int	decoding flag=Q(PKSM+x+1)

J.10 BeamInt

Variable	Type	Description
evt->beamIntKx.Integ2us	int	Integration over 2μs
evt->beamIntKx.Integ15us	int	Integration over 15μs
evt->beamIntKx.Integ30us	int	Integration over 30μs
evt->beamIntKx.Integ60us	int	Integration over 60μs
evt->beamIntKx.Qual2us	int	quality of integration over 2μs
evt->beamIntKx.Qual15us	int	quality of integration over 15μs
evt->beamIntKx.Qual30us	int	quality of integration over 30μs
evt->beamIntKx.Qual60us	int	quality of integration over 60μs
evt->beamIntKx.packsamp[4]	int	packed samples for channel 2

J.11 AKScounter

Variable	Type	Description
evt->aks[*counter*].time	float	time of hit in counter (CB aks_rec_)

Variable	Type	Description
evt->aks[*counter*].time2	float	time of snd hit (available from 4.2)
evt->aks[*counter*].mips	float	number of m.i.p.s (CB aks_rec_)
evt->aks[*counter*].error	int	error flag (CB aks_rec_)

J.12 AKLhit

Variable	Type	Description
evt->akl[*hit*].pocket	int	pocket number (1-7) = Q(RAHI+2)
evt->akl[*hit*].layer	int	layer number (1,2) = Q(RAHI+3)
evt->akl[*hit*].counter	int	counter number (1-12) = Q(RAHI+4)
evt->akl[*hit*].mips	float	number of m.i.p.s = Q(RAHI+5)
evt->akl[*hit*].time	float	time of AKL hit = Q(RAHI+6)
evt->akl[*hit*].ONTflag	int	on time flag = Q(RAHI+7)

J.13 DCHcluster

Variable	Type	Description
evt->DCHclu[*clu*].view	int	View (1-16: 1-4 ch1, 5-8 ch2, 9-10 ch3, 13-16 ch4)
evt->DCHclu[*clu*].coord	int	cluster coordinate for view ($\in [0; 256] cm$)
evt->DCHclu[*clu*].time	int	cluster time (ns) (CB dchclu_cmpblk_ for entire struct.)

J.14 DCHhit

Variable	Type	Description
evt->track[*track*].hit[*hit*].plane	int	plane number (CB dchsac_cmpblk_)
evt->track[*track*].hit[*hit*].wire	int	wire number (CB dchsac_cmpblk_)
evt->track[*track*].hit[*hit*].time	float	time of hit (CB dchsac_cmpblk_)
evt->track[*track*].hit[*hit*].dist	float	distance of hit (CB dchsac_cmpblk_)
evt->track[*track*].hit[*hit*].pos	float	position of hit (CB dchsac_cmpblk_)

J.15 DCHspacepoint

Variable	Type	Description
evt->track[*track*].spnt[*dch*].x	float	x coord (CB dchsac_cmpblk_)
evt->track[*track*].spnt[*dch*].y	float	y coord (CB dchsac_cmpblk_)
evt->track[*track*].spnt[*dch*].z	float	z coord (CB dchsac_cmpblk_)

J.16 DCHtrack

Variable	Type	Description
evt->track[*track*].pq	float	track mom.*charge=Q(RDTK+4)*Q(RDTK+3)
evt->track[*track*].p	float	track momentum=Q(RDTK+4)
evt->track[*track*].q	int	not FILLED
evt->track[*track*].perr	float	error on momentum=Q(RDTK+5)
evt->track[*track*].chi2	float	χ^2 of track=Q(RDTK+6)
evt->track[*track*].bx	float	x position before magnet=Q(RDTK+7)
evt->track[*track*].by	float	y position before magnet=Q(RDTK+8)

Variable	**Type**	**Description**
evt->track[*track*].bdxdz	float	dx/dz before magnet=Q(RDTK+9)
evt->track[*track*].bdydz	float	dy/dz before magnet=Q(RDTK+10)
evt->track[*track*].x	float	x position after magnet=Q(RDTK+15)
evt->track[*track*].y	float	y position after magnet=Q(RDTK+16)
evt->track[*track*].dxdz	float	dx/dz after magnet=Q(RDTK+17)
evt->track[*track*].dydz	float	dy/dz after magnet=Q(RDTK+18)
evt->track[*track*].time	float	track time=Q(RDTK+23)
evt->track[*track*].quality	float	quality of track=Q(RDTK+24)
evt->track[*track*].spnt[4]	DCHspacepoint	space points in 4 chambers
evt->track[*track*].hodtime	float	hod. time ass. with track=Q(RDTK+25)
evt->track[*track*].hodstatus	int	hod. flags used for time calc.=Q(RDTK+26)
evt->track[*track*].nhits	int	Nb of hits - always filled (CB dchsac_cmpblk_)
evt->track[*track*].Nhit	u_int	Number of track hits - only for DCH streams
evt->track[*track*].hit[64]	DCHhit	track hits - only for DCH streams
evt->track[*track*].exhac	float	energy in HAC x strip=Q(RHCL+13)
evt->track[*track*].eyhac	float	energy in HAC y strip=Q(RHCL+14)
evt->track[*track*].ddeadcell	float	Dist. to closest dead cell(cm)=Q(RDTK+27)
evt->track[*track*].sigxx	float	Error on x (cm)=Q(RDTK+11)
evt->track[*track*].sigyy	float	Error on y (cm)=Q(RDTK+12)

Variable	Type	Description
evt->track[*track*].sigdxdx	float	Error on dx/dz =Q(RDTK+13)
evt->track[*track*].sigdydy	float	Error on dy/dz =Q(RDTK+14)
evt->track[*track*].sigxdx	float	correlation sigma(x,dx/dz)=Q(RDTK+28)
evt->track[*track*].sigxy	float	correlation sigma(x,y)=Q(RDTK+29)
evt->track[*track*].sigdxy	float	correlation sigma(dx/dz,y)=Q(RDTK+30)
evt->track[*track*].sigxdy	float	correlation sigma(x,dy/dz)=Q(RDTK+31)
evt->track[*track*].sigdxdy	float	correlation sigma(dx/dz,dy/dz)=Q(RDTK+32)
evt->track[*track*].sigydy	float	correlation sigma(y,dy/dz) =Q(RDTK+33)
evt->track[*track*].HitPattern	int	one bit per wire for efficiency studies
evt->track[*track*].efficiency[2]	int	bit coded words for eff. studies
evt->track[*track*].spareInt[2]	int	2 spare integers
evt->track[*track*].spareFloat[2]	int	2 spare floats
evt->track[*track*].LKRclu	int	LKR clu. index=0 $\rightarrow NLkr - 1$ (std_ep)
evt->track[*track*].EovP	float	E/p (comp. in std_ep)
evt->track[*track*].Espy	float	Energy in NUTspy (comp in espy)
evt->track[*track*].muvTime	float	time(ns) of closest muon in time (comp murec1198)
evt->track[*track*].anavar[20]	float	Provision for analysis variables
evt->track[*track*].anaflag[5]	int	Provision for analysis flags

J.17 DCHvertex

Variable	Type	Description
evt->vertex[*vertex*].x	float	x position(cm)=Q(RDVX+6)
evt->vertex[*vertex*].y	float	y position(cm)=Q(RDVX+7)
evt->vertex[*vertex*].z	float	z position(cm)=Q(RDVX+8)
evt->vertex[*vertex*].chi2	float	chi2 vertex fit =Q(RDVX+24)
evt->vertex[*vertex*].bdxdzPos	float	dx/dz bef.mag. Pos. track =Q(RDVX+10)
evt->vertex[*vertex*].bdydzPos	float	dy/dz bef.mag. Pos. track =Q(RDVX+11)
evt->vertex[*vertex*].bdxdzNeg	float	dx/dz bef.mag. Neg. track =Q(RDVX+12)
evt->vertex[*vertex*].bdydzNeg	float	dy/dz bef.mag. Neg. track =Q(RDVX+13)
evt->vertex[*vertex*].Nbdxdztrack	u_int	Number of dxdz for track i
evt->vertex[*vertex*].bdxdztrack[7]	float	dxdz for track i
evt->vertex[*vertex*].Nbdydztrack	u_int	Number of dxdz for track i
evt->vertex[*vertex*].bdydztrack[7]	float	dxdz for track i
evt->vertex[*vertex*].cda	float	closest approach(cm)=Q(RDVX+9)
evt->vertex[*vertex*].ErrorFlag	int	1:Error ana. done, 0: not done=Q(RDVX+23)
evt->vertex[*vertex*].BlueFlag	int	1:BlueField used, 0: not used=Q(RDVX+14)
evt->vertex[*vertex*].ppipi	float	$\pi^+\pi^-$ vertex momentum(GeV)=Q(RDVX+3)
evt->vertex[*vertex*].mpipi	float	$\pi^+\pi^-$ invariant mass(GeV)=Q(RDVX+4)
evt->vertex[*vertex*].mlambda	float	Λ^0 mass(GeV) - **All the following variables**

Variable	Type	Description
evt->vertex[*vertex*].mAlambda	float	Anti-Λ^0 mass - **computed from**
evt->vertex[*vertex*].iptrk	int	index of the positive track
evt->vertex[*vertex*].intrk	int	index of the negative track
evt->vertex[*vertex*].Nitrack	u_int	Number of Index of tracks i in this vertex
evt->vertex[*vertex*].itrack[7]	int	Index of tracks i in this vertex
evt->vertex[*vertex*].rcog	float	cog(cm) **rlib/srcVertexVariables.c**
evt->vertex[*vertex*].ptsqks	float	_$P_\perp^2$ (GeV^2) for K_s
evt->vertex[*vertex*].ptsqkl	float	_$P_\perp^2$ (GeV^2) for K_l
evt->vertex[*vertex*].ptprimeks	float	_$\tilde{P_\perp}^2$ (GeV^2) for K_s
evt->vertex[*vertex*].ptprimekl	float	_$\tilde{P_\perp^2}$ (GeV^2) for K_l
evt->vertex[*vertex*].dtintks	float	_$D_{target}^{in}(cm)$ for K_s
evt->vertex[*vertex*].dtintkl	float	_$D_{target}^{in}(cm)$ for K_l
evt->vertex[*vertex*].dtouttks	float	_$D_{target}^{out}(cm)$ for K_s
evt->vertex[*vertex*].dtouttkl	float	_$D_{target}^{out}(cm)$ for K_l
evt->vertex[*vertex*].dtinvks	float	_$D_{vertex}^{in}(cm)$ for K_s
evt->vertex[*vertex*].dtinvkl	float	_$D_{vertex}^{in}(cm)$ for K_l
evt->vertex[*vertex*].dtoutvks	float	_$D_{vertex}^{out}(cm)$ for K_s
evt->vertex[*vertex*].dtoutvkl	float	_$D_{vertex}^{out}(cm)$ for K_l
evt->vertex[*vertex*].eangle	float	Kaon Ener.(GeV) (open angle)
evt->vertex[*vertex*].ctau	float	(lifetime units)
evt->vertex[*vertex*].PhiDecay	float	Decay plane phi (rd)
evt->vertex[*vertex*].Asp	float	_$(P_- - P_+)/(P_- + P_+)$: signed asymetry
evt->vertex[*vertex*].hodotimeAna	float	(comp. by USER_HODOTIME routine) (ns)
evt->vertex[*vertex*].type	int	vertex type - (fuser_newcharged)

Variable	Type	Description
evt->vertex[*vertex*].cuts	int	bit coded: which cuts passed (fuser_newcharged)
evt->vertex[*vertex*].iflag	int	see comments in fuser_newcharged.F
evt->vertex[*vertex*].anavar[20]	float	Provision for analysis variables
evt->vertex[*vertex*].anaflag[5]	int	Provision for analysis flags

J.18 DCHNROtdof

Variable	Type	Description
evt->DCHnro.dof[*idof*].tdof[*it*].ttime	float	Q(LPDOFNR+10) tdc dof time relative to the Timestamp(ns)
evt->DCHnro.dof[*idof*].tdof[*it*].tstatus	int	Q(LPDOFNR+11) status word for tdc dof

J.19 DCHNROdof

Variable	Type	Description
evt->DCHnro.dof[*idof*].dcherror	int	Q(LPDOFNR+1) DECDCH error code, 0=raw data format OK
evt->DCHnro.dof[*idof*].nro	int	Q(LPDOFNR+2) Nb of planes having ring ovf (=1)
evt->DCHnro.dof[*idof*].ipl	int	Q(LPDOFNR+7) plane number= only 1
evt->DCHnro.dof[*idof*].rstatus	int	Q(LPDOFNR+8) status word for ring dof
evt->DCHnro.dof[*idof*].Ntdof	u_int	Number of tdc overflows info
evt->DCHnro.dof[*idof*].tdof[45]	DCHNROtdof	tdc overflows info

J.20 DCHNROhit

Variable	Type	Description
evt->DCHnro.hit[*ihit*].id	float	Q(LPDCHNR+x+1)
evt->DCHnro.hit[*ihit*].iplane	float	Q(LPDCHNR+x+2) plane number (increasing order)
evt->DCHnro.hit[*ihit*].iwire	float	Q(LPDCHNR+x+3) wire number (increasing order)
evt->DCHnro.hit[*ihit*].time	float	Q(LPDCHNR+x+4) drift time (nsec)
evt->DCHnro.hit[*ihit*].itrack	float	Q(LPDCHNR+x+5)
evt->DCHnro.hit[*ihit*].itslot	float	Q(LPDCHNR+x+6) time slot info

J.21 DCHNROgen

Variable	Type	Description
evt->DCHnro.evtime	float	Q(LPDCHNR+1)
evt->DCHnro.timest	int	Q(LPDCHNR+5) .and. Q(LPDCHNR+6)
evt->DCHnro.trigw	int	Q(LPDCHNR+7)
evt->DCHnro.flag	int	Q(LPDCHNR+8)
evt->DCHnro.Nhit	u_int	Number of hits info
evt->DCHnro.hit[1792]	DCHNROhit	hits info
evt->DCHnro.Ndof	u_int	Number of overflows info for the nro plane
evt->DCHnro.dof[1]	DCHNROdof	overflows info for the nro plane

J.22 DCHFEstatus

Variable	Type	Description
evt->DCHFE.inttrig	int	Q(LPDOFE+8)
evt->DCHFE.F1par	int	
evt->DCHFE.pipefull	int	
evt->DCHFE.pipefullWIRE[8]	int	

J.23 DCHmult

Variable	Type	Description
evt->DCHEFFmult.MBXPlaneEff[6]	int	number of hit for plane packed
evt->DCHEFFmult.L1Trk24Eff[2]	int	multiplicity in DCH1 for L1 trigger packed
evt->DCHEFFmult.DCHSnowErr[2]	int	16 bit for DCH snow effect(2-1)(4-3)
evt->DCHEFFmult.MBXmult	int	multiplicity in DCH
evt->DCHEFFmult.Trk24ON	int	L1Trk24=1 −¿ trigger OK

J.24 SGNwire

Variable	Type	Description
evt->chamber[*dch*].spnt[*spnt*].wire[*n*].number	int	wire numb. (CB dchsgn_cmpblk_)
evt->chamber[*dch*].spnt[*spnt*].wire[*n*].drifttime	float	drift time (CB dchsgn_cmpblk_)

J.25 SGNspacepoint

Variable	Type	Description
evt->chamber[*dch*].spnt[*spnt*].x	float	x coordinate (CB dchsgn_cmpblk_)
evt->chamber[*dch*].spnt[*spnt*].y	float	y coordinate (CB dchsgn_cmpblk_)
evt->chamber[*dch*].spnt[*spnt*].Nwire	u_int	Number of Wire Nb + drift time
evt->chamber[*dch*].spnt[*spnt*].wire[8]	SGNwire	Wire Nb + drift time
evt->chamber[*dch*].spnt[*spnt*].quality[4]	int	qual. flag/view (CB dchsgn_cmpblk_)

J.26 SGNaccidental

Variable	Type	Description
evt->chamber[*dch*].acc[*acc*].x	float	x coordinate (CB dchsgn_cmpblk_)

Variable	Type	Description
evt->chamber[*dch*].acc[*acc*].y	float	y coordinate (CB dchsgn_cmpblk_)
evt->chamber[*dch*].acc[*acc*].time	float	Accidental time (CB dchsgn_cmpblk_)

J.27 SGNchamber

Variable	Type	Description
evt->chamber[*dch*].Nspnt	u_int	Number of space point array
evt->chamber[*dch*].spnt[50]	SGNspacepoint	space point array
evt->chamber[*dch*].Nacc	u_int	Number of accidental array
evt->chamber[*dch*].acc[50]	SGNaccidental	accidental array
evt->chamber[*dch*].tovrflw[16]	float	nearest ovflw times to evt(CB dchsgn_cmpblk_)
evt->chamber[*dch*].rovrflw	int	ring buffer ovrflw (CB dchsgn_cmpblk_)

J.28 SGNtrack

Variable	Type	Description
evt->SGNtrk[*track*].p	float	momentum*charge of track (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].chi2	float	χ^2 of track - NOT filled
evt->SGNtrk[*track*].bx	float	x position before magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].by	float	y position before magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].bdxdz	float	dx/dz before magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].bdydz	float	dy/dz before magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].x	float	x position after magnet (CB dchsgn_cmpblk_)

Variable	Type	Description
evt->SGNtrk[*track*].y	float	y position after magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].dxdz	float	dx/dz after magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].dydz	float	dy/dz after magnet (CB dchsgn_cmpblk_)
evt->SGNtrk[*track*].spnt[4]	int	space points on track (CB dchsgn_cmpblk_)

J.29 SGNvertex

Variable	Type	Description
evt->SGNvtx[*vertex*].x	float	x position (CB dchsgn_cmpblk_)
evt->SGNvtx[*vertex*].y	float	y position (CB dchsgn_cmpblk_)
evt->SGNvtx[*vertex*].z	float	z position (CB dchsgn_cmpblk_)
evt->SGNvtx[*vertex*].cda	float	closest approach (CB dchsgn_cmpblk_)
evt->SGNvtx[*vertex*].flag	int	vertex flag - NOT filled

J.30 HODhit

Variable	Type	Description
evt->hod[*hit*].plane	int	plane number - NOT used
evt->hod[*hit*].counter	int	counter hit = Q(RHHI+5)
evt->hod[*hit*].x	float	x coord of DCH track = Q(RHHI+6)
evt->hod[*hit*].y	float	y coord of DCH track = Q(RHHI+7)
evt->hod[*hit*].pheight	float	pulse height (from PDS) = Q(RHHI+10)

Variable	Type	Description
evt->hod[*hit*].time	float	time of hit (all corrections added) = Q(RHHI+13)
evt->hod[*hit*].uctime	float	uncorrected time of hit = Q(RHHI+11)
evt->hod[*hit*].pdsflag	int	flag from PDS data structure = Q(RHHI+3)

J.31 HODneuthit

Variable	Type	Description
evt->hodneut[*hit*].plane	int	Plane number (1=V, 2=H) = Q(LPHOD+x+2)
evt->hodneut[*hit*].counter	int	scintillator number (1-64) = Q(LPHOD+x+3)
evt->hodneut[*hit*].pulse	float	pulse height (mips) = Q(LPHOD+x+4)
evt->hodneut[*hit*].time	float	time (ns) from ramp = Q(LPHOD+x+5)

J.32 LKRcluster

Variable	Type	Description
evt->lkr[*cluster*].energy	float	Cluster energy(GeV)=Q(RLCL+4)
evt->lkr[*cluster*].eenergy	float	NOT filled from 4.5
evt->lkr[*cluster*].e2sampall	float	$< E >$ (samp. 1+2) all cells in clu.(GeV)=Q(RLCL+16)
evt->lkr[*cluster*].e77	float	clu. energy in 7x7 cells(GeV)=Q(RLCL+17)
evt->lkr[*cluster*].x	float	x coord of cluster(cm)=Q(RLCL+7)
evt->lkr[*cluster*].y	float	y coord of cluster(cm)=Q(RLCL+8)

Variable	Type	Description
evt->lkr[*cluster*].rmsx	float	X cluster width (cell u.) =Q(RLCL+9)
evt->lkr[*cluster*].rmsy	float	Y cluster width (cell u.) =Q(RLCL+10)
evt->lkr[*cluster*].imax	int	index of maximum energy cell=Q(RLCL+3)
evt->lkr[*cluster*].time	float	Time(ns)=Q(RLCL+11)
evt->lkr[*cluster*].etime	float	NOT filled from 4.5
evt->lkr[*cluster*].tlatcell	float	time of most energetic lateral cell(ns)=Q(RLCL+13)
evt->lkr[*cluster*].ddeadcell	float	distance of closest dead cell(cm)=Q(RLCL+14)
evt->lkr[*cluster*].ucenergy	float	uncorrected cluster energy(GeV)=Q(RLCL+15)
evt->lkr[*cluster*].cellsread	int	Number of cells read out=Q(RLCL+2)
evt->lkr[*cluster*].status	int	status bits=Q(RLCL+6)
evt->lkr[*cluster*].spachacorr	float	corr. applied for spacecharge effect=Q(RLCL+18)
evt->lkr[*cluster*].ecorrke3	float	cl. energy corrected with ke3 factor(GeV)=Q(RLCL+19)
evt->lkr[*cluster*].spareInt1	int	1 spare integers in case
evt->lkr[*cluster*].spareFloat1	float	Chi2 shower shape = Q(RLCL+12)
evt->lkr[*cluster*].gainmax	int	gain of cell with max. energy
evt->lkr[*cluster*].Ecellmax	float	energy of the highest energy cell
evt->lkr[*cluster*].timeraw	float	cluster time before t0 corr.
evt->lkr[*cluster*].mctailcorr	float	energy tail corr from MC
evt->lkr[*cluster*].anavar[5]	float	Provision for analysis variables

Variable	Type	Description
evt->lkr[*cluster*].anaflag[5]	int	Provision for analysis flags

J.33 KABhit

Variable	Type	Description
evt->kab[*idet*].strip[*istrip*].hit[*hit*].tlead	float	Leading time of hit = Q(PKAB+xxx+4)
evt->kab[*idet*].strip[*istrip*].hit[*hit*].ttrail	float	Trailing time of hit = Q(PKAB+xxx+5)
evt->kab[*idet*].strip[*istrip*].hit[*hit*].trackID	int	MC Track index ($1 \rightarrow gen\_track$) = Q(PKAB+xxx+1)

J.34 KABstrip

Variable	Type	Description
evt->kab[*idet*].strip[*istrip*].Nhit	u_int	Number of hits in this strip
evt->kab[*idet*].strip[*istrip*].hit[50]	KABhit	hits in this strip

J.35 KABdet

Variable	Type	Description
evt->kab[*idet*].strip[48]	KABstrip	number of strips in this detector

J.36 KABtrack

Variable	Type	Description
evt->KABtrk[*track*].pq	float	track mom.*charge
evt->KABtrk[*track*].p	float	track momentum
evt->KABtrk[*track*].q	int	not FILLED
evt->KABtrk[*track*].UPorDOWN	int	first station UP=1 or DOWN=2
evt->KABtrk[*track*].perr	float	error on momentum
evt->KABtrk[*track*].chi2	float	χ^2 (quality) of track

Variable	Type	Description
evt->KABtrk[*track*].x	float	x position in second station
evt->KABtrk[*track*].y	float	y position in second station
evt->KABtrk[*track*].xUorD	float	x position in UP or Down station
evt->KABtrk[*track*].yUorD	float	y position in UP or Down station
evt->KABtrk[*track*].sigxxUorD	float	Error on xUorD
evt->KABtrk[*track*].sigyyUorD	float	Error on yUorD
evt->KABtrk[*track*].time	float	track time
evt->KABtrk[*track*].timeUorD	float	track time at UP or Down station
evt->KABtrk[*track*].timeST2	float	track time at second station
evt->KABtrk[*track*].dxdz	float	dx/dz
evt->KABtrk[*track*].dydz	float	dy/dz
evt->KABtrk[*track*].RecFlag	int	reconstruction flag
evt->KABtrk[*track*].sigxx	float	Error on x (cm)
evt->KABtrk[*track*].sigyy	float	Error on y (cm)
evt->KABtrk[*track*].sigtt	float	Error on time (ns)
evt->KABtrk[*track*].sigdxdx	float	Error on dx/dz
evt->KABtrk[*track*].sigdydy	float	Error on dy/dz
evt->KABtrk[*track*].sigpy	float	correlation sigma(p,y)
evt->KABtrk[*track*].sigxt	float	correlation sigma(x,time)
evt->KABtrk[*track*].sigxdx	float	correlation sigma(x,dx/dz)
evt->KABtrk[*track*].sigtdx	float	correlation sigma(time,dx/dz)
evt->KABtrk[*track*].sigpx	float	correlation sigma(p,x) not FILLED
evt->KABtrk[*track*].sigxy	float	correlation sigma(x,y) not FILLED
evt->KABtrk[*track*].sigpt	float	correlation sigma(p,time) not FILLED

Variable	Type	Description
evt->KABtrk[*track*].sigyt	float	correlation sigma(y,time) not FILLED
evt->KABtrk[*track*].sigpdx	float	correlation sigma(p,dx/dz) not FILLED
evt->KABtrk[*track*].sigydx	float	correlation sigma(y,dx/dz) not FILLED
evt->KABtrk[*track*].spareInt[2]	int	2 spare integers
evt->KABtrk[*track*].spareFloat[2]	float	2 spare floats
evt->KABtrk[*track*].anavar[20]	float	Provision for analysis variables
evt->KABtrk[*track*].anaflag[5]	int	Provision for analysis flags

J.37 KABFTWindow

Variable	Type	Description
evt->KABFADC[0].Channel[*i*].TWindow[*i*].MaxPulseHeight	int	max pulseheight in this window
evt->KABFADC[0].Channel[*i*].TWindow[*i*].TFirst	float	time of 1st sample (w/r to timestamp)
evt->KABFADC[0].Channel[*i*].TWindow[*i*].TMax	float	time of max pulseheight in this window
evt->KABFADC[0].Channel[*i*].TWindow[*i*].NPulseHeight	u_int	Number of pulseheight of sample
evt->KABFADC[0].Channel[*i*].TWindow[*i*].PulseHeight[300]	int	pulseheight of sample

J.38 KABFChannel

Variable	Type	Description
evt->KABFADC[0].Channel[*i*].Id	int	channel number
evt->KABFADC[0].Channel[*i*].Mode	int	sampling mode (time step size in ns (1/2)
evt->KABFADC[0].Channel[*i*].DECflag	int	error code from dectag routine
evt->KABFADC[0].Channel[*i*].NTWindow	u_int	Number of time windows in this channel

Variable	Type	Description
evt->KABFADC[0].Channel[*i*].TWindow[60]	KABFTWindow	time windows in this channel

J.39 KABFADC

Variable	Type	Description
evt->KABFADC[0].TimeStamp	int	timestamp
evt->KABFADC[0].Offset	int	r/o offset for timestamp
evt->KABFADC[0].NChannel	u_int	Number of FADC channels
evt->KABFADC[0].Channel[20]	KABFChannel	FADC channels

J.40 NHOhit

Variable	Type	Description
evt->nho[*hit*].counter	int	counter number(1-32) = Q(RNHI+4)
evt->nho[*hit*].x	float	x coord of hit(cm) = Q(RNHI+5)
evt->nho[*hit*].y	float	y coord of hit(cm) = Q(RNHI+6)
evt->nho[*hit*].pheight	float	pulse height (from PDS) = Q(RNHI+9)
evt->nho[*hit*].time	float	time of hit(ns) (all corr. added) = Q(RNHI+11)
evt->nho[*hit*].uctime	float	uncorr. time of hit(ns) = Q(RNHI+10)
evt->nho[*hit*].pdsflag	int	flag from PDS data structure = Q(RNHI+3)

J.41 HACcluster

Variable	Type	Description
evt->hac[*cluster*].energy	float	Energy = Q(RHCL+4)
evt->hac[*cluster*].bfratio	float	back to front ratio of energy = Q(RHCL+5)

Variable	Type	Description
evt->hac[*cluster*].x	float	x coord of cluster = Q(RHCL+7)
evt->hac[*cluster*].y	float	y coord of cluster = Q(RHCL+8)
evt->hac[*cluster*].rmsx	float	cluster x-RMS(strips) = Q(RHCL+11)
evt->hac[*cluster*].rmsy	float	cluster y-RMS(strips) = Q(RHCL+12)
evt->hac[*cluster*].emaxx	float	Max Ener. in strip at extrapolated track x = Q(RHCL+13)
evt->hac[*cluster*].emaxy	float	Max Ener. in strip at extrapolated track y= Q(RHCL+14)
evt->hac[*cluster*].time	float	Cluster time = Q(RHCL+6)
evt->hac[*cluster*].recflag	int	Status bits = Q(RHCL+15) (cf NA48 note 98-4)

J.42 HACcell

Variable	Type	Description
evt->phac[*icell*].idcell	int	cell id = Q(PHAC+x+1)
evt->phac[*icell*].energy	float	cell energy = Q(PHAC+x+3)

J.43 MUVhit

Variable	Type	Description
evt->muv[*hit*].x	float	x position of hit = Q(RMUH+1)
evt->muv[*hit*].y	float	y position of hit = Q(RMUH+2)
evt->muv[*hit*].time	float	time of hit = Q(RMUH+7)

Variable	Type	Description
evt->muv[*hit*].dtime	float	accuracy of the time = Q(RMUH+8)
evt->muv[*hit*].status	int	Number of hits in time with track
evt->muv[*hit*].plane[2]	int	Bit coded word for the planes used in the hit
evt->muv[*hit*].chi2	float	χ^2. Meaningful only if trackID != -1
evt->muv[*hit*].trackID	int	Track index ($0 \rightarrow evt- > N\_track - 1$)
evt->muv[*hit*].vertexID	int	Vertex index ($0 \rightarrow evt- > N\_track - 1$)

J.44 PMUVhit

Variable	Type	Description
evt->pmuv[*chan*].hit[*hit*].time	float	time of hit = Q(PMUV+xxx+1)
evt->pmuv[*chan*].hit[*hit*].width	float	time over threshold = Q(PMUV+xxx+2)
evt->pmuv[*chan*].hit[*hit*].info	float	pulse status = Q(PMUV+xxx+3)

J.45 PMUVchannel

Variable	Type	Description
evt->pmuv[*chan*].n	int	channel number = Q(PMUV+x+1)
evt->pmuv[*chan*].Nhit	u_int	Number of hits in this channel
evt->pmuv[*chan*].hit[20]	PMUVhit	hits in this channel

J.46 L3filter

Variable	Type	Description
evt->filter[*n*].index	int	filter index
evt->filter[*n*].info	int	filter information

J.47 L3particle

Variable	Type	Description
evt->part[n].type	int	particle type

J.48 etaCluster

Variable	Type	Description
evt->etaSumm.clu[iclu].flag	int	0: not meson cand.; 1: meson cand.-Q(RETA+x+1)
evt->etaSumm.clu[iclu].ncell	int	Ncell readout Q(RETA+x+2)
evt->etaSumm.clu[iclu].imax	int	Index of cell with max. energy Q(RETA+x+3)
evt->etaSumm.clu[iclu].nrec	int	Nb of cells with recov. pb in clu. rec. Q(RETA+x+4)
evt->etaSumm.clu[iclu].Ecell[25]	float	E(GeV) each cell around clu. centre-Q(RETA+x+5+i)

J.49 etaSummary

Variable	Type	Description
evt->etaSumm[0].flag	int	1=$\pi^0 \to \gamma\gamma$, 2=$\eta \to \gamma\gamma$, 3=$\eta \to 3\pi^0$,4: ch. =Q(RETA+1)
evt->etaSumm[0].xvtxz	float	Neutral vertex (from Ks target)=Q(RETA+2)
evt->etaSumm[0].xcog	float	Meson center of gravity=Q(RETA+3)
evt->etaSumm[0].eMeson	float	Meson energy (GeV)=Q(RETA+4)
evt->etaSumm[0].Nclu	u_int	Number of List of eta clusters
evt->etaSumm[0].clu[50]	etaCluster	List of eta clusters

J.50 ananeut

Variable	Type	Description
evt->aneut.esc	float	Energy scale factor (input to user_sel2pi0)
evt->aneut.iflag	int	1 for $2\pi^0$ cand., 2 for $3\pi^0$,0 if not, -1 no recbank
evt->aneut.cem[3]	float	$\pi^0\pi^0$ masses - sel2pi0/3pi0
evt->aneut.inc[6]	int	clusters ids - sel2pi0/3pi0
evt->aneut.relli	float	R ellipse - sel2pi0/3pi0
evt->aneut.ctau	float	c*tau (lifetimes) - sel2pi0/3pi0
evt->aneut.ekaon	float	Kaon energy (GeV) - sel2pi0/3pi0
evt->aneut.LKRtime	float	time from LKR (ns) - LKRtime
evt->aneut.LKRNHODtime	float	time from LKR+NHOD (ns) - LKRtime
evt->aneut.ntUsed	int	Nb of cells used to compute time - LKRtime
evt->aneut.NHODtime	float	time from NHOD - nhodotime
evt->aneut.cuts	int	To store bits for selection

J.51 anacharg

Variable	Type	Description
evt->acharg.ktype	int	vertex type: 1 for Ks, 2 for Kl
evt->acharg.iflag	int	bit0: 1 for $\pi^+\pi^-$ cand., 0 if not
evt->acharg.ivertex	int	best vertex index
evt->acharg.cuts	int	bits describing cuts (newcharged)

J.52 anachodcount

Variable	Type	Description
evt->achod.acount[*icount*].Plane	float	Planes indices used
evt->achod.acount[*icount*].Counter	float	Counters indices used
evt->achod.acount[*icount*].Time	float	Time for counters used
evt->achod.acount[*icount*].sig	float	sig

J.53 anacharghod

Variable	Type	Description
evt->achod.hodotime	float	Charged hod. event time
evt->achod.itrace[2]	int	tracks index to be used by hodotime
evt->achod.Nacount	u_int	Number of Counters used for hodotime
evt->achod.acount[20]	anachodcount	Counters used for hodotime
evt->achod.hsigg	float	hsigg
evt->achod.vsigg	float	vsigg

J.54 anaprothit

Variable	Type	Description
evt->atag.aprot[*proton*].prothit[*prothit*].indexhit[2]	int	1st and Snd index of hit

J.55 anaproton

Variable	Type	Description
evt->atag.aprot[*proton*].protTime	float	proton time
evt->atag.aprot[*proton*].protStat	int	proton code
evt->atag.aprot[*proton*].Nprothit	u_int	Number of hits belonging to proton
evt->atag.aprot[*proton*].prothit[20]	anaprothit	hits belonging to proton

J.56 anatagger

Variable	Type	Description
evt->atag.Naprot	u_int	Number of identified protons
evt->atag.aprot[100]	anaproton	identified protons
evt->atag.nprotLadder	int	nprotLadder
evt->atag.nprotMonitor	int	nprotMonitor

J.57 anaaks

Variable	Type	Description
evt->aaks.flag	int	see doc. 04-06-98 fuser_aksflag description

J.58 anacalled

Variable	Type	Description
evt->acall.selcharged	int	0: not; >0: called
evt->acall.sel2pi0	int	0: not; >0: called
evt->acall.sel3pi0	int	0: not; >0: called
evt->acall.bluefield	int	0: not; 1: called
evt->acall.lkrpedcor	int	0: not; 1: called
evt->acall.lkrposcor	int	0: not; 1: called
evt->acall.lkrsharing	int	0: not; 1: called
evt->acall.hodotime	int	0: not; 1: called
evt->acall.nhodtime	int	0: not; 1: called
evt->acall.lkrtime	int	0: not; 1: called
evt->acall.tagtime	int	0: not; 1: called
evt->acall.aksflag	int	0: not; 1: called
evt->acall.muon_rec	int	0: not; 1: called
evt->acall.muon_reject	int	0: not; 1: called
evt->acall.geomcor	int	0: not; 1: called
evt->acall.magnetcor	int	0: not; 1: called
evt->acall.newcharged	int	0: not; 1: called
evt->acall.muon_trackrec	int	0: not; 1: called
evt->acall.muon_vtxrec	int	0: not; 1: called
evt->acall.lkrcalcor	int	0: not; 1: called

Variable	Type	Description
evt->acall.lkrcalcor1	int	0: not; 1: called
evt->acall.lkrcalcor2	int	0: not; 1: called
evt->acall.lkrcalcor3	int	0: not; 1: called
evt->acall.lkrcalhi2k	int	0: not; 1: called
evt->acall.sel2gam	int	0: not; >0: called
evt->acall.sel3pic	int	0: not; 1: called
evt->acall.sel3pin	int	0: not; 1: called

J.59 cmpEvent

Variable	Type	Description
evt->n	int	trigger event number
evt->trigword	int	trigger word (bits 16 and 17 for RaNDoMs)
evt->timestamp	int	event timestamp
evt->recflag	int	reconstruction flags
evt->FlagCorr	int	1bit/analysis routine called
evt->evtfspare[10]	float	dummy spare floats
evt->evtispare[10]	int	dummy spare ints
evt->dbErr	int	non-zero if error from SQLITE db
evt->rndm	RNDMsummary	RaNDoM events summary
evt->Ntstamp	u_int	Number of time stamps!=evt->timestamp
evt->tstamp[11]	EVTtimestamp	time stamps!=evt->timestamp
evt->KLmon	PMBscaler	KL monitor scaler
evt->KLmon3us	PMBscaler	KL monitor scaler 3μs ago
evt->QXmon	PMBscaler	QX monitor scaler
evt->QXmon3us	PMBscaler	QX monitor scaler 3μs ago
evt->KSmon	PMBscaler	KS monitor scaler
evt->AKSmon	PMBscaler	AKs monitor scaler

Variable	Type	Description
evt->mainsphase	float	50Hz mains phase to 40MHz clock - from PSCA
evt->SPSphase	float	SPS frequency phase to 40MHz clock - from PSCA
evt->mainsphaseRaw	int	raw mains phase
evt->SPSphaseRaw	int	raw SPS phase
evt->FEmainsphase	float	mains phase (1st event method) - from PSCA
evt->FESPSphase	float	SPS phase (1st event method) - from PSCA
evt->Ntrig	u_int	Number of PMB pattern unit data
evt->trig[16]	PMBtimeslice	PMB pattern unit data
evt->PATerrflag	int	Pattern unit error flag = IQ(PCAT+18)
evt->PATstatus	int	Pattern unit status bits - NOT filled
evt->TAGantic	int	anti-counter hit bits - NOT filled
evt->Ntag	u_int	Number of Tagger channel data
evt->tag[60]	TAGchannel	Tagger channel data
evt->TAGerrflag	int	Tagger error flag = Q(RTAG+3)
evt->TAGstatus	int	Tagger status bits = Q(RTAG+1)
evt->Nksm	u_int	Number of KS monitor hit data
evt->ksm[20]	KSMhit	KS monitor hit data
evt->KSMerrflag	int	KSM error flag = Q(PKSM+1)
evt->KSMstatus	int	KSM status bits = Q(RANT+4)
evt->beamIntKs	BeamInt	integrated Ks intensity
evt->beamIntKl	BeamInt	integrated Kl intensity

Variable	Type	Description
evt->AKLtime	float	AKL event time = Q(RANT+2)
evt->Nakl	u_int	Number of Anti-counter hit data
evt->akl[100]	AKLhit	Anti-counter hit data
evt->AKLerrflag	int	AKL error flag = Q(RANT+1)
evt->AKLstatus	int	Anti-counter status bits = Q(RANT+4)
evt->AKSmips[5]	float	NOT USED
evt->AKStime[5]	float	NOT USED
evt->aks[7]	AKScounter	AKS counter data
evt->AKSerrflag	int	AKS error flag - NOT filled
evt->AKSstatus	int	AKS status flag (CB aks_rec_)
evt->DCHbz	float	z before magnet in DCHs=Q(RDCH+2)
evt->DCHz	float	z after magnet in DCHs=Q(RDCH+3)
evt->DCHbzCorr	float	Corr. z bef. magnet (comp. in TrackVertexCorr)
evt->DCHzCorr	float	Corr. z aft. magnet (comp. in TrackVertexCorr)
evt->DCHnhits[16]	int	Nhits/view×4 chambers (CB dchsac_cmplk_)
evt->DCHovrflow	int	overflow bits (CB dchsac_cmplk_)
evt->BestVertex	int	NOT used from 4.3.3
evt->NDCHclu	u_int	Number of Clusters in each view
evt->DCHclu[100]	DCHcluster	Clusters in each view
evt->Ntrack	u_int	Number of track structures
evt->track[40]	DCHtrack	track structures
evt->NtrackCorr	u_int	Number of track structures after corr.

Variable	Type	Description
evt->trackCorr[40]	DCHtrack	track structures after corr.
evt->VTXfrRDVX	int	0: vtx built from compact; !=0: rec. from RDVX
evt->Nvertex	u_int	Number of vertex structures
evt->vertex[100]	DCHvertex	vertex structures
evt->NvertexCorr	u_int	Number of vertex structures after corr.
evt->vertexCorr[100]	DCHvertex	vertex structures after corr.
evt->DCHerrflag	int	DCH error flag= Q(RDCH+5)
evt->DCHerrflagPDS	int	DCH error flag from PDS = Q(PDCH+8)
evt->DCHstatus	int	MBOX simu. bits - from RCAT or PU(ovl)
evt->MBOXdeadtime	int	MBOX deadtime ON (comp. in getMBOXdeadTime.c)
evt->DCHnro	DCHNROgen	DCH New readout test structure
evt->DCHFE	DCHFEstatus	DCH Fe readout status structure
evt->DCHEFFmult	DCHmult	DCH multiplicity info structure
evt->SGNtime	int	DCH time reference
evt->chamber[4]	SGNchamber	chamber structures
evt->NSGNtrk	u_int	Number of track structures - NOT avail. from 4.3
evt->SGNtrk[40]	SGNtrack	track structures - NOT avail. from 4.3
evt->NSGNvtx	u_int	Number of vertex structures - NOT avail. from 4.3
evt->SGNvtx[100]	SGNvertex	vertex structures - NOT avail. from 4.3

Variable	Type	Description
evt->HODtime	float	Event time (ns)= Q(RHOD+2)
evt->Nhod	u_int	Number of hodoscope hit data
evt->hod[50]	HODhit	hodoscope hit data
evt->HODneutflag	int	set to 1 if Nhits was above limit
evt->Nhodneut	u_int	Number of hodoscope hit data
evt->hodneut[16]	HODneuthit	hodoscope hit data
evt->HODerrflag	int	Hodoscope error flag= Q(RHOD+1)
evt->HODstatus	int	Hodoscope status bits= Q(RHOD+3)
evt->LKREtotCell	float	Energy sum of cells (7^{th} samp.) (GeV) = Q(PLKR+7)
evt->LKRenergy	float	Clusters energy sum (GeV)=Q(RLKR+2)
evt->Nlkr	u_int	Number of LKR cluster data
evt->lkr[50]	LKRcluster	LKR cluster data
evt->LKRerrflag	int	LKR error flag= Q(RLKR+3)
evt->LKRstatus	int	LKR status bits (set to zero)
evt->LKRdownscaled	int	LKR downscale flag (=1 downsc., =0 not downsc.)
evt->Nkabhit	int	Total number of raw KABES hits
evt->kab[6]	KABdet	KAB detector data
evt->NKABtrk	u_int	Number of KABES track data
evt->KABtrk[50]	KABtrack	KABES track data
evt->KABerrflag	int	KAB error flag= Q(RKAB+5)

Variable	Type	Description
evt->KABerrflagPDS	int	KAB error flag from PDS = Q(PKAB+6)
evt->NKABFADC	u_int	Number of KABES FADC data (filled only in the dedicated stream)
evt->KABFADC[1]	KABFADC	KABES FADC data (filled only in the dedicated stream)
evt->NUThitmap[4]	int	hit map (2 projections x 64 bits) - from PNUT
evt->NUTxpeak[4]	int	x peaks times=(-3→0)=IQ(PNUT+x+171)
evt->NUTypeak[4]	int	y peaks times=(-3→0)=IQ(PNUT+x+172)
evt->NUTnxpeak[4]	int	x peaks (bef. LUT) (-3→ 0)=IQ(PNUT+x+20)
evt->NUTnypeak[4]	int	y peaks (bef. LUT) (-3→ 0)=IQ(PNUT+x+21)
evt->NUTm0[2]	int	0th moment (2 proj.)= IQ(PNUT+x+22,23)
evt->NUTm1[2]	int	1st moment (2 proj.)= IQ(PNUT+x+24,25)
evt->NUTm2[2]	int	2nd moment (2 proj.)= IQ(PNUT+x+26,27)
evt->NUTm0smp[6]	int	0th moment (sample -2,-1,0,+1,+2) - from PNUT
evt->NUTm1smp	int	1st moment (sample 0) - from PNUT
evt->NUTm2smp	int	2nd moment (sample 0) - from PNUT
evt->NUThacsmp	int	HAC (sample 0)= IQ(PNUT+x+170)-upto compact-4.3

Variable	Type	Description
evt->NUThacEsum[6]	int	HAC (sample -2,-1,0,+1,+2)= IQ(PNUT+x+170)-fr compact-4.4
evt->NUTts	int	Trigger control bits (sample 0) IQ(PNUT+x+19)
evt->spy_x[64]	int	spy chan.(X view)=IQ(PNUT+x+28+i)
evt->spy_y[64]	int	spy chan.(Y view)=IQ(PNUT+x+92+i)
evt->NUTerrflag	int	Neutral trigger error flag=IQ(PNUT+5)
evt->NUTstatus	int	status bits - NOT filled
evt->NHOtime	float	Event time(ns)= Q(RNHO+2)
evt->Nnho	u_int	Number of Neutral hod. hit data
evt->nho[50]	NHOhit	Neutral hod. hit data
evt->NHOerrflag	int	Neutral hod. error flag= Q(RNHO+1)
evt->NHOstatus	int	Neutral hod. status bits= Q(RNHO+3)
evt->HACenergy	float	Total energy in the HAC= Q(RHAC+2)
evt->NGoodHAC	int	Nb of HAC clu. ass. to tracks=Q(RHAC+3)
evt->Nhac	u_int	Number of HAC cluster data
evt->hac[50]	HACcluster	HAC cluster data
evt->HACerrflag	int	HAC error flag (set to 0)
evt->HACstatus	int	HAC status bits (set to 0)
evt->Nphac	u_int	Number of HAC cell data
evt->phac[176]	HACcell	HAC cell data
evt->Nmuv	u_int	Number of MUV hit data

Variable	Type	Description
evt->muv[50]	MUVhit	MUV hit data
evt->Npmuv	u_int	Number of Muon veto channel PDS data
evt->pmuv[56]	PMUVchannel	Muon veto channel PDS data
evt->Npmuvnew	u_int	Number of New Muon veto channel PDS data
evt->pmuvnew[56]	PMUVchannel	New Muon veto channel PDS data
evt->MUVerrflag	int	MUV dec. status= Q(RMUV+3)
evt->MUVstatus	int	MUV rec. status= Q(RMUV+1)
evt->L3evtype	int	Event type - NOT USED
evt->L3status	int	L3 status bits - NOT USED
evt->L3ErrDec	int	L3 flag for decoding errors
evt->L3ErrRec	int	L3 flag for reconstruction errors
evt->L3rtrstatus	int	real-time-rec. status bits - NOT USED
evt->L3trigword[2]	int	Level 3 trigger words
evt->L3ONLINEtrigword[2]	int	Level 3 trigger words at run time
evt->L3Btrigword[2]	int	Level 3 trigger words at repro
evt->L3ActionDownScale	int	Downscaling flag for cuts per trigger in L3
evt->L3ActionDownScale2	int	Downscaling flag for cuts per trigger in L3
evt->L3FilterDownScale	int	Downscaling flag for filter in L3
evt->L3ActionOvl	int	Overlay flag per trigger in L3
evt->L3ActionOvl2	int	Overlay flag per trigger in L3
evt->L3FilterOvl	int	Overlay flag per filter in L3

Variable	Type	Description
evt->L3CutFlag	int	Flagged cuts in L3
evt->L3CutFlag2[10]	int	Flagged cuts in L3
evt->L3raredecay	int	rare decay bits
evt->L3evsteer	int	event steering information
evt->Nfilter	u_int	Number of L3 filter information
evt->filter[16]	L3filter	L3 filter information
evt->Npart	u_int	Number of L3 particles
evt->part[10]	L3particle	L3 particles
evt->Ntrhist	u_int	Number of NOT USED
evt->trhist[10]	L3TRIGhist	NOT USED
evt->L3ncells	int	number of cells
evt->L3ncellabt	int	number of cells above threshold
evt->L3rawenergy	float	raw LKR energy
evt->L3dchhits[4]	int	number of DCH hits
evt->L3ep	float	best vertex highest E/p
evt->L3MinDist	float	Minimum distance for E/p calculation
evt->L3zvert	float	best vertex z-coord
evt->L3cda	float	best vertex closest distance of approach
evt->L3pipimass	float	2-track invariant mass for best vertex
evt->L3ppair	float	2-track momentum for best vertex
evt->L3cogc	float	momentum weighted cog for b vtx
evt->L3tau	float	proper lifetime of best vertex
evt->L3nclus	int	number of LKR clusters
evt->L3pi0pi0mass[2]	float	two π^0 mass combinations
evt->L3cogn	float	centre of gravity for neutrals
evt->L2Bstatus	int	L2B Bits 0-15:clust.,16:error,17:acc.,18:treat.

Variable	Type	Description
evt->L2Bcomclust	int	L2B cluster combination
evt->L2Bshadr[6]	int	L2B channel address of shower max.
evt->L2Bshx[6]	int	L2B Shower x coord. ($\times$100)
evt->L2Bshy[6]	int	L2B Shower y coord. ($\times$100)
evt->L2Bshenergy[6]	int	L2B Shower Energy ($\times$1000)
evt->L2Bshtime[6]	int	L2B Shower Time ($\times$10)
evt->L2Bshxrms[6]	int	L2B Shower x rms ($\times$100)
evt->L2Bshyrms[6]	int	L2B Shower y rms ($\times$100)
evt->L3spare[2]	float	spare L3 words
evt->L3ShowerWidth	float	Shower Width computed in L3 - NOT USED
evt->status	int	Overall event status bits
evt->CmpFilter	int	Compact Filter bits selection - NOT FILLED
evt->NovrflwSim	int	Number of simulated overflows
evt->ovrflwSimBef	float	overflow closest to 0, Before 0
evt->ovrflwSimAft	float	overflow closest to 0, After 0
evt->NTrigWordL	u_int	Number of TrigWord (ALL evts in burst)
evt->TrigWordL[70000]	int	TrigWord (ALL evts in burst)
evt->NTimeStampL	u_int	Number of TimeStamp (ALL evts in burst)
evt->TimeStampL[70000]	int	TimeStamp (ALL evts in burst)
evt->NDCHDecErrorL	u_int	Number of DCH Dec. error (ALL evts in burst)

Variable	Type	Description
evt->DCHDecErrorL[70000]	int	DCH Dec. error (ALL evts in burst)
evt->NLKRHACDecErrorL	u_int	Number of LKR+HAC Dec. error (ALL evts in burst)
evt->LKRHACDecErrorL[70000]	int	LKR+HAC Dec. error (ALL evts in burst)
evt->NChargedInfoL	u_int	Number of Charged Filter bits (ALL evts in burst)
evt->ChargedInfoL[70000]	int	Charged Filter bits (ALL evts in burst)
evt->NNeutralInfoL	u_int	Number of Neutral Filter bits (ALL evts in burst)
evt->NeutralInfoL[70000]	int	Neutral Filter bits (ALL evts in burst)
evt->NetaSumm	u_int	Number of struct. used for η runs
evt->etaSumm[1]	etaSummary	struct. used for η runs
evt->NTHMPackedData	u_int	Number of THM packed data
evt->THMPackedData[90]	int	THM packed data
evt->spareInt[10]	int	10 spare integers in case
evt->spare01Int[10]	int	10 spare integers in case
evt->spareFloat[10]	float	10 spare floats in case
evt->spare01Float[10]	float	10 spare floats in case
evt->nTrigBef	int	Ntriggers in prev. 100ms-NOT YET FILLED
evt->timeToPrev	int	time to prev. trigger(ns)-NOT YET FILLED
evt->aneut	ananeut	structure to store neutral variables
evt->acharg	anacharg	structure to store charged variables

Variable	Type	Description
evt->achod	anacharghod	structure to hold ch. hodo. analysis
evt->atag	anatagger	structure to hold tagger analysis
evt->aaks	anaaks	structure to hold aks analysis variables
evt->acall	anacalled	one word per ana. routine

J.60 XoffDet

Variable	Type	Description
bur->XoffDet.data[3]	float	NOT used from 4.2
bur->XoffDet.Ntransition	float	Nb of XOFF transitions OFF → ON
bur->XoffDet.FracTimeON	float	Fraction of time XOFF was ON (rel. to XoffBurstLen)
bur->XoffDet.FirstTimeON	float	First time in burst XOFF was ON (s)

J.61 NSstat

Variable	Type	Description
bur->beamNS.(P42/K12).bend	int	0: OK; 1 out of toler. (magnets:1A,coll.:1mm
bur->beamNS.(P42/K12).quad	int	quad
bur->beamNS.(P42/K12).trim	int	trim
bur->beamNS.(P42/K12).scraper	int	scraper
bur->beamNS.(P42/K12).coll	int	coll
bur->beamNS.(P42/K12).colr	int	colr

J.62 BeamNonStandard

Variable	Type	Description
bur->beamNS.P42	NSstat	Non Standard statistique for P42

Variable	Type	Description
bur->beamNS.K12	NSstat	Non Standard statistique for K12
bur->beamNS.TargT4	float	Intensity in T4 ($\times 10^{10} prot.$)
bur->beamNS.TargT10	float	Intensity in T10 ($\times 10^{10} prot.$)
bur->beamNS.TargKsY	float	KS target X position (mm)
bur->beamNS.TargKsX	float	KS target Y position (mm)
bur->beamNS.TaxMot7	float	Collimator postion (mm)
bur->beamNS.TaxMot8	float	Collimator postion (mm)
bur->beamNS.TaxMot17	float	Collimator postion (mm)
bur->beamNS.TaxMot18	float	Collimator postion (mm)
bur->beamNS.T10CollStart	float	T10 Collimator positon
bur->beamNS.T10CollEnd	float	T10 Collimator positon
bur->beamNS.CollProtTag	float	Collimator-proton tagging: 1(in)/0(out)
bur->beamNS.CollProtTagX	float	Coll.-proton tagging: x pos. (mm)
bur->beamNS.CollProtTagY	float	Coll.-proton tagging: y pos. (mm)
bur->beamNS.CollCleanColl	float	Collimator: 1(in)/0(out)
bur->beamNS.CollCleanCollX	float	Coll. : x pos. (mm)
bur->beamNS.CollCleanCollY	float	Coll. : y pos. (mm)
bur->beamNS.CollDefColl	float	Defining collimator: 1(in)/0(out)
bur->beamNS.CollDefCollX	float	Defining coll.: x pos. (mm)
bur->beamNS.CollDefCollY	float	Defining coll.: y pos. (mm)
bur->beamNS.CryDisHor	float	Crystal x pos. (mm)

Variable	Type	Description
bur->beamNS.CryDisVer	float	Crystal y pos. (mm)
bur->beamNS.CryRotHor	float	Crystal rot. in V. plane
bur->beamNS.CryRotVer	float	Crystal rot. in H. plane
bur->beamNS.CryTemp	float	Crystal temperature
bur->beamNS.XAksDistHor	float	AKS x pos. (mm)
bur->beamNS.XAksRotHor	float	AKS rot. in H. plane
bur->beamNS.XAksRotVer	float	AKS rot. in V. plane
bur->beamNS.P42Conv	float	P42 converter: 1(in)/0(out)
bur->beamNS.K12Conv	float	K12 converter: 1(in)/0(out)
bur->beamNS.VaccPump55	float	Pressure
bur->beamNS.VaccPump56	float	Pressure
bur->beamNS.VaccPump98	float	Pressure
bur->beamNS.VaccPump58	float	Pressure
bur->beamNS.VaccValve56	float	Pressure
bur->beamNS.VaccValve58	float	Pressure
bur->beamNS.SpsBend1K12	float	Current for bend 1 in K12 (K12-B1) (A). This is the Achromat 1
bur->beamNS.SpsBend2K12	float	Current for bend 2 in K12 (K12-B2) (A)
bur->beamNS.SpsBend3K12	float	Current for bend 3 in K12 (K12-B3) (A)
bur->beamNS.SpsBend4K12	float	Current for bend 4 in K12 (K12-B4) (A)
bur->beamNS.SpsBend5K12	float	Current for bend 5 in K12 (K12-B5) (A)
bur->beamNS.P42FileNumber	float	File name
bur->beamNS.K12FileNumber	float	File name

J.63 BeamStandard

Variable	Type	Description
bur->beamS.SpsDate	float	Date
bur->beamS.SpsHour	float	Time
bur->beamS.SpsT4Int	float	Intensity on T4 ($10^{10} protons$)

Variable	Type	Description
bur->beamS.SpsT4Sym	float	Symmetry in T4 (<0 if ups. count. missing)
bur->beamS.SpsT10Int	float	Intensity in T10 ($10^{10} protons$)
bur->beamS.SpsT4	float	Nb of magnets with errors at T4
bur->beamS.SpsK12	float	Nb of magnets with errors at K12
bur->beamS.SpsP0	float	Nb of magnets with errors at P0
bur->beamS.SpsMnp33Coil1	float	Current for coil1 of MNP33(K12-B6) (A)
bur->beamS.SpsMnp33Coil2	float	Current for coil2 of MNP33(K12-B7) (A)
bur->beamS.SpsBend1K12	float	Current for bend 1 in K12 (K12-B1) (A)
bur->beamS.SpsStatus	float	Machine Status Word
bur->beamS.spare[4]	float	spare

J.64 TSscalars

Variable	Type	Description
bur->ts.TSscal.Nscaler	u_int	Number of TS scalers
bur->ts.TSscal.scaler[100]	int	TS scalers

J.65 L2TSscalars

Variable	Type	Description
bur->ts.L2TSscal.Nscaler	u_int	Number of L2 TS scalers
bur->ts.L2TSscal.scaler[20]	int	L2 TS scalers

J.66 AKLscalars

Variable	Type	Description
bur->ts.AKLscal.Akl1OR	int	AKL: AKL1:OR scal.
bur->ts.AKLscal.Akl2OR	int	AKL: AKL2:OR scal.
bur->ts.AKLscal.Akl3OR	int	AKL: AKL3:OR scal.
bur->ts.AKLscal.Akl4OR	int	AKL: AKL4:OR scal.

Variable	Type	Description
bur->ts.AKLscal.Akl5OR	int	AKL: AKL5:OR scal.
bur->ts.AKLscal.Akl6OR	int	AKL: AKL6:OR scal.
bur->ts.AKLscal.Akl7OR	int	AKL: AKL7:OR scal.
bur->ts.AKLscal.AklOR	int	AKL: AKL :OR scal.
bur->ts.AKLscal.Akl1OrAnd	int	AKL: AKL1: OrAnd scal.
bur->ts.AKLscal.Akl2OrAnd	int	AKL: AKL2: OrAnd scal.
bur->ts.AKLscal.Akl3OrAnd	int	AKL: AKL3: OrAnd scal.
bur->ts.AKLscal.Akl4OrAnd	int	AKL: AKL4: OrAnd scal.
bur->ts.AKLscal.Akl5OrAnd	int	AKL: AKL5: OrAnd scal.
bur->ts.AKLscal.Akl6OrAnd	int	AKL: AKL6: OrAnd scal.
bur->ts.AKLscal.Akl7OrAnd	int	AKL: AKL7: OrAnd scal.
bur->ts.AKLscal.AklOrAnd	int	AKL: AKL : OrAnd scal.

J.67 TriggerSupervisor

Variable	Type	Description
bur->ts.TSscal97	TSscalars97	1997 TS scalars
bur->ts.TSscal	TSscalars	TS scalars
bur->ts.L2TSscal97	L2TSscalars97	1997 L2 TS scalars
bur->ts.L2TSscal	L2TSscalars	L2 TS scalars
bur->ts.AKLscal	AKLscalars	AKL scalars

J.68 BeamMonitor

Variable	Type	Description
bur->BM.BadMac	int	0: OK; > 0: bctr MAC had problem
bur->BM.FastSpillKs	float	Eff. spill length seen by Delco-KS
bur->BM.averageAks	float	Average AKS intensity (Hz)

Variable	Type	Description
bur->BM.MaxAks	float	Maximum int. of AKS readings (Hz)
bur->BM.RmsAks	float	RMS int. of AKS readings (Hz)
bur->BM.SlowDCKs	float	Slow Duty Cycle-KS
bur->BM.ExtrLenKs	float	Extraction Length-KS ($\in [0-1]$)
bur->BM.FastSpillKl	float	Eff. spill length seen by Delco-KS
bur->BM.averageBctr	float	Average BCtr intensity (Hz)
bur->BM.MaxBctr	float	Maximum int. of BCtr readings (Hz)
bur->BM.RmsBctr	float	RMS int. of BCtr readings (Hz)
bur->BM.SlowDCKl	float	Slow Duty Cycle-KL
bur->BM.ExtrLenKl	float	Extraction Length-KL ($\in [0-1]$)
bur->BM.spare[3]	float	spares
bur->BM.Nsample	int	Nb of samples for vectors
bur->BM.Fclock	int	Final clock
bur->BM.Fhac1	int	Single part. trigger in HAC monitor
bur->BM.Fhac2	int	$\geq$ 2 part. in HAC (top$\times$bot + left$\times$right)
bur->BM.Fmuons	int	Muon veto trigger
bur->BM.Faks	int	Final aks count
bur->BM.Ft1t2	int	Final t1 t2 count
bur->BM.Fqx	int	Final Qx count
bur->BM.Fbctr	int	Final BCTR count
bur->BM.FTwoMu	int	2 muons rate from MUV
bur->BM.Fhv1	int	BCtr downscaled
bur->BM.Fhvd1	int	BCtr downscaled further
bur->BM.Fhvd2	int	BCtr downscaled further
bur->BM.Fhvd3	int	BCtr downscaled further
bur->BM.Fhvd4	int	BCtr downscaled further
bur->BM.Fdelco	int	Final Delco count

Variable	Type	Description
bur->BM.Fakl	int	AKL total OR
bur->BM.FksmDelco	int	Final ksmDelco count
bur->BM.Fksm	int	Final KSM count
bur->BM.clock[400]	int	Clock counts (100kHz) bef. readings
bur->BM.aks[400]	int	AKS coincidence
bur->BM.t1t2[400]	int	Tagger monitor counters (counts per interval)
bur->BM.qx[400]	int	QX from charged hodoscope
bur->BM.bctr[400]	int	Direct beamcounter
bur->BM.delco[400]	int	BCtr delayed self-coincidence (w=20ns, del=57ns)
bur->BM.ksmDelco[400]	int	KS target mon. delayed self-co (w=20ns del=??ns)
bur->BM.ksm[400]	int	KS targer monitor

J.69 BeamMonitor01

Variable	Type	Description
bur->BM01.BadMac	int	0: OK; > 0: bctr MAC had problem
bur->BM01.FastSpillKs	float	Eff. spill length seen by Delco-KS
bur->BM01.averageAks	float	Average AKS intensity (Hz)
bur->BM01.MaxAks	float	Maximum int. of AKS readings (Hz)
bur->BM01.RmsAks	float	RMS int. of AKS readings (Hz)
bur->BM01.SlowDCKs	float	Slow Duty Cycle-KS
bur->BM01.ExtrLenKs	float	Extraction Length-KS ($\in [0-1]$)
bur->BM01.FastSpillKl	float	Eff. spill length seen by Delco-KS
bur->BM01.averageBctr	float	Average BCtr intensity (Hz)

Variable	Type	Description
bur->BM01.MaxBctr	float	Maximum int. of BCtr readings (Hz)
bur->BM01.RmsBctr	float	RMS int. of BCtr readings (Hz)
bur->BM01.SlowDCKl	float	Slow Duty Cycle-KL
bur->BM01.ExtrLenKl	float	Extraction Length-KL ($\in [0-1]$)
bur->BM01.spare[3]	float	spares
bur->BM01.Nsample	int	Nb of samples for vectors
bur->BM01.Fclock	int	Final clock
bur->BM01.Fhac1	int	Single part. trigger in HAC monitor
bur->BM01.Fhac2	int	≥ 2 part. in HAC (top$\times$bot + left$\times$right)
bur->BM01.Fmuons	int	Muon veto trigger
bur->BM01.Faks	int	Final aks count
bur->BM01.Ft1t2	int	Final t1 t2 count
bur->BM01.Fqx	int	Final Qx count
bur->BM01.Fbctr	int	Final BCTR count
bur->BM01.FTwoMu	int	2 muons rate from MUV
bur->BM01.Fhv1	int	BCtr downscaled
bur->BM01.Fhvd1	int	BCtr downscaled further
bur->BM01.Fhvd2	int	BCtr downscaled further
bur->BM01.Fhvd3	int	BCtr downscaled further
bur->BM01.Fhvd4	int	BCtr downscaled further
bur->BM01.Fdelco	int	Final Delco count
bur->BM01.Fakl	int	AKL total OR
bur->BM01.FksmDelco	int	Final ksmDelco count
bur->BM01.Fksm	int	Final KSM count
bur->BM01.clock[800]	int	Clock counts (100kHz) bef. readings
bur->BM01.aks[800]	int	AKS coincidence
bur->BM01.t1t2[800]	int	Tagger monitor counters (counts per interval)
bur->BM01.qx[800]	int	QX from charged hodoscope
bur->BM01.bctr[800]	int	Direct beamcounter

Variable	Type	Description
bur->BM01.delco[800]	int	BCtr delayed self-coincidence (w=20ns, del=57ns)
bur->BM01.ksmDelco[800]	int	KS target mon. delayed self-co (w=20ns del=??ns)
bur->BM01.ksm[800]	int	KS targer monitor

J.70 LkrCalib

Variable	Type	Description
bur->LkrCalib.trigger[30]	int	trigger for calibration
bur->LkrCalib.DAC	float	DAC value

J.71 DataMerger

Variable	Type	Description
bur->DM.error	int	???
bur->DM.DetError	int	???

J.72 TagClock

Variable	Type	Description
bur->TagClk.phase	float	Clock0 Phase
bur->TagClk.RefVolt1	float	Reference Voltage
bur->TagClk.RefVolt2	float	Reference Voltage

J.73 MUVScalars

Variable	Type	Description
bur->MUVscal.Muons	int	μ's scal.
bur->MUVscal.TwoMuons	int	2 μ scal.

J.74 BadBurst

Variable	Type	Description
bur->BadB.Call	int	if >0: badburst routine called
bur->BadB.Skip	int	if >0 entire burst is skipped
bur->BadB.Lkr	int	LKR flag (comp. in fuser_badburst.F)
bur->BadB.Dch	int	Dch flag (comp. in fuser_badburst.F)
bur->BadB.Nut	int	Nut flag (comp. in fuser_badburst.F)
bur->BadB.Mbx	int	Mbx flag (comp. in fuser_badburst.F)
bur->BadB.Hac	int	Hac flag (comp. in fuser_badburst.F)
bur->BadB.Tag	int	Tag flag (comp. in fuser_badburst.F)
bur->BadB.Muv	int	Muv flag (comp. in fuser_badburst.F)
bur->BadB.HodC	int	Ch. Hod. flag (comp. in fuser_badburst.F)
bur->BadB.HodN	int	Neutr. Hod. flag (comp. in fuser_badburst.F)
bur->BadB.PMB	int	PMB flag (comp. in fuser_badburst.F)
bur->BadB.Aks	int	AKS flag (comp. in fuser_badburst.F)
bur->BadB.Akl	int	AKL flag (comp. in fuser_badburst.F)
bur->BadB.Clk	int	Clock flag (comp. in fuser_badburst.F)
bur->BadB.Ksm	int	KSM flag (comp. in fuser_badburst.F)
bur->BadB.Kab	int	Kabes flag (comp. in fuser_badburst.F)
bur->BadB.NoEps	int	Not a $\frac{\epsilon'}{\epsilon}$ burst

Variable	Type	Description
bur->BadB.Phys	int	Burst not good for physics (Kcharged)
bur->BadB.spare1	int	spare
bur->BadB.spare2	int	spare
bur->BadB.spare3	int	spare
bur->BadB.spare4	int	spare

J.75 TimeOffset

Variable	Type	Description
bur->tOffst.Version	float	Version of timin-offset file read-in
bur->tOffst.Tag	float	Tagger offset
bur->tOffst.Aks	float	Aks offset
bur->tOffst.Akl	float	Akl offset
bur->tOffst.Hod	float	Ch. hod. offset
bur->tOffst.Nho	float	Neutr. hod. offset
bur->tOffst.Dch	float	Dch offset
bur->tOffst.Lkr	float	Lkr offset
bur->tOffst.Hac	float	Hac offset
bur->tOffst.Muv	float	Muv offset
bur->tOffst.LkrTag	float	Lkr-Tag offset
bur->tOffst.LkrNhod	float	Lkr-nhod offset
bur->tOffst.LkrAkl	float	Lkr-Akl offset
bur->tOffst.LkrHac	float	Lkr-Hac offset
bur->tOffst.KabPlus	float	Fine tuning of Kabes offset for K plus
bur->tOffst.KabMinus	float	Fine tuning of Kabes offset for K Minus

J.76 superTimeOffset

This structure is used for compact supercompact and hypercompact bursts. Here the version for compact bursts is exemplified. In order to access time offsets from supercompact and hypercompact, you should substitute “bur->” with “sbur->” or “hbur->” depending on which kind of data you are analyzing.

Variable	Type	Description
bur->tOffst.Version	float	Version of timin-offset file read-in
bur->tOffst.Tag	float	Tagger offset
bur->tOffst.Aks	float	Aks offset
bur->tOffst.Kab	float	Kabes offset
bur->tOffst.Nmv	float	New Muon Veto offset
bur->tOffst.Akl	float	Akl offset
bur->tOffst.Hod	float	Ch. hod. offset
bur->tOffst.Nho	float	Neutr. hod. offset
bur->tOffst.Dch	float	Dch offset
bur->tOffst.Lkr	float	Lkr offset
bur->tOffst.Hac	float	Hac offset
bur->tOffst.Muv	float	Muv offset
bur->tOffst.LkrTag	float	Lkr-Tag offset
bur->tOffst.LkrNhod	float	Lkr-nhod offset
bur->tOffst.LkrAkl	float	Lkr-Akl offset
bur->tOffst.LkrHac	float	Lkr-Hac offset
bur->tOffst.KabPlus	float	Fine tuning of Kabes offset for K plus
bur->tOffst.KabMinus	float	Fine tuning of Kabes offset for K Minus

J.77 anatocall

Variable	Type	Description
bur->tocall.selcharged	int	0: not; >0: to call
bur->tocall.sel2pi0	int	0: not; >0: to call
bur->tocall.sel3pi0	int	0: not; >0: to call
bur->tocall.bluefield	int	0: not; 1: to call
bur->tocall.lkrpedcor	int	0: not; 1: to call
bur->tocall.lkrposcor	int	0: not; 1: to call
bur->tocall.lkrsharing	int	0: not; 1: to call
bur->tocall.hodotime	int	0: not; 1: to call
bur->tocall.nhodtime	int	0: not; 1: to call
bur->tocall.lkrtime	int	0: not; 1: to call
bur->tocall.tagtime	int	0: not; 1: to call

Variable	Type	Description
bur->tocall.aksflag	int	0: not; 1: to call
bur->tocall.muon_rec	int	0: not; 1: to call
bur->tocall.muon_reject	int	0: not; 1: to call
bur->tocall.geomcor	int	0: not; 1: to call
bur->tocall.magnetcor	int	0: not; 1: to call
bur->tocall.newcharged	int	0: not; 1: to call
bur->tocall.muon_trackrec	int	0: not; 1: to call
bur->tocall.muon_vtxrec	int	0: not; 1: to call
bur->tocall.lkrcalcor	int	0: not; 1: to call
bur->tocall.lkrcalcor1	int	0: not; 1: to call
bur->tocall.lkrcalcor2	int	0: not; 1: to call
bur->tocall.lkrcalcor3	int	0: not; 1: to call
bur->tocall.lkrcalhi2k	int	0: not; 1: to call
bur->tocall.sel2gam	int	0: not; >0: to call

J.78 Burst

Variable	Type	Description
bur->majorVer	int	COmPACT major version number
bur->minorVer	int	COmPACT minor version number
bur->patch	int	COmPACT patch number
bur->time	int	burst time
bur->hepdb[20]	int	Timestamps from HEPdb - not filled yet
bur->brtype	int	Type of burst (data/MC/..)
bur->bmtype	int	Type of beam (K_S,K_L,K_S+K_L,....)
bur->intensity	float	number protons on T10
bur->nrun	int	run number
bur->nevent	int	— Not filled
bur->dataset	int	Data set value
bur->KSMsca[32]	int	KS monitor scalers

Variable	Type	Description
bur->NHOsca[32]	int	Neutral hodoscope scalers
bur->HODVsca[64]	int	Hodoscope vertical scalers
bur->HODHsca[64]	int	Hodoscope horizontal scalers
bur->logic[128]	int	Logic scalers
bur->syps[32]	int	Symmetric pre-trigger scaler
bur->TSsca[124]	int	NOT filled - back. compat.
bur->ts	TriggerSupervisor	TS data
bur->XoffBurstLen	float	Burst Length (s)
bur->XoffDch	XoffDet	Xoff data for DCH
bur->XoffLkr	XoffDet	Xoff data for LKR
bur->XoffNut	XoffDet	Xoff data for NUT
bur->XoffHac	XoffDet	Xoff data for HAC
bur->XoffAkl	XoffDet	Xoff data for AKL
bur->XoffTag	XoffDet	Xoff data for TAG
bur->XoffHod	XoffDet	Xoff data for HOD
bur->XoffMuv	XoffDet	Xoff data for MUV
bur->TagClk	TagClock	Tagger Clock Data
bur->beamNS	BeamNonStandard	Beam Non Standard data
bur->beamS	BeamStandard	Beam Standard data
bur->BM	BeamMonitor	Beam Monitor data
bur->BM01	BeamMonitor01	Beam Monitor data for longer spill
bur->DM	DataMerger	Data Merger data
bur->LkrCalib	LkrCalib	LKR scalars data
bur->MUVscal	MUVScalars	Muon Veto scalars
bur->NTrigWord	u_int	Number of TrigWord (ALL evts in burst)
bur->TrigWord[70000]	int	TrigWord (ALL evts in burst)
bur->NTimeStamp	u_int	Number of TimeStamp (ALL evts in burst)

Variable	Type	Description
bur->TimeStamp[70000]	int	TimeStamp (ALL evts in burst)
bur->NDCHDecError	u_int	Number of DCH Dec. error (ALL evts in burst)
bur->DCHDecError[70000]	int	DCH Dec. error (ALL evts in burst)
bur->NLKRHACDecError	u_int	Number of LKR+HAC Dec. error (ALL evts in burst)
bur->LKRHACDecError[70000]	int	LKR+HAC Dec. error (ALL evts in burst)
bur->NChargedInfo	u_int	Number of Charged Filter bits (ALL evts in burst)
bur->ChargedInfo[70000]	int	Charged Filter bits (ALL evts in burst)
bur->NNeutralInfo	u_int	Number of Neutral Filter bits (ALL evts in burst)
bur->NeutralInfo[70000]	int	Neutral Filter bits (ALL evts in burst)
bur->BadB	BadBurst	Structure holding Bad Burst data
bur->ChamberDz[4]	float	Chamber Delta(z) w.r.t. to Geom file
bur->tOffst	superTimeOffset	Structure holding time offsets
bur->CallAnaRoutine	int	see text in user-guide
bur->tocall	anatocall	one word per ana. routine to call
bur->getEoBbefore	int	NOT USED yet
bur->dberr	int	Database error code (0=OK;-1=NOK)

J.79 L3errorcounter

Variable	Type	Description
eob->L3errctr.error	int	generic error

Variable	Type	Description
eob->L3errctr.decevt	int	event decoding errors
eob->L3errctr.decTAG	int	tagger decoding errors
eob->L3errctr.decAKS	int	AKS decoding errors
eob->L3errctr.decAKL	int	anticounter decoding errors
eob->L3errctr.decDCH	int	driftchamber decoding errors
eob->L3errctr.decHOD	int	hodoscope decoding errors
eob->L3errctr.decNHO	int	neutral hodoscope decoding errors
eob->L3errctr.decLKR	int	liquid krypton decoding errors
eob->L3errctr.decHAC	int	hadron calorimeter decoding errors
eob->L3errctr.decMUV	int	muon veto decoding errors
eob->L3errctr.decPAT	int	pattern unit decoding errors
eob->L3errctr.decNUT	int	neutral trigger decoding errors
eob->L3errctr.decL3	int	level 3 decoding errors
eob->L3errctr.recTAG	int	tagger reconstruction errors
eob->L3errctr.recAKS	int	AKS reconstruction errors
eob->L3errctr.recAKL	int	AKL reconstruction errors
eob->L3errctr.recDCH	int	driftchamber reconstruction errors
eob->L3errctr.recHOD	int	hodoscope reconstruction errors
eob->L3errctr.recNHO	int	neutral hodoscope reconstruction errors
eob->L3errctr.recLKR	int	liquid krypton reconstruction errors
eob->L3errctr.recHAC	int	hadron calorimeter reconstruction errors
eob->L3errctr.recMUV	int	muon veto reconstruction errors

Variable	Type	Description
eob->L3errctr.recBTH	int	both DCH reconstructions' errors
eob->L3errctr.comdch_nodch	int	computation err. (no DCH info)
eob->L3errctr.comep_nodch	int	comp. error on E/p (no DCH info)
eob->L3errctr.comep_nolkr	int	comp. error on E/p (no LKR info)
eob->L3errctr.comep_doubleclus	int	comp. error on E/p (two clu.)
eob->L3errctr.comep_noclus	int	comp. error on E/p (no clu.)
eob->L3errctr.comep_sameclus	int	comp. error on E/p (2 tracks on clu.)
eob->L3errctr.commbox_nodch	int	COMMBOX error counters
eob->L3errctr.comnut_nolkr	int	COMNUT error counters
eob->L3errctr.coml3b_nolkr	int	COML3B error counters
eob->L3errctr.spare	int	spare variable for compatibility

J.80 decResult

Variable	Type	Description
eob->aerrdec.decRes[*idec*].code	int	Record code 0=master, 1=echan, 2=log
eob->aerrdec.decRes[*idec*].echan	int	Error ID in call to errcountf()
eob->aerrdec.decRes[*idec*].log	int	Error log value in call to errcountf()
eob->aerrdec.decRes[*idec*].Nerr	int	Number of errors
eob->aerrdec.decRes[*idec*].SatFlag	int	Flag if log values truncated
eob->aerrdec.decRes[*idec*].evtno	int	if satflag=1,Event no. where sat. occurred
eob->aerrdec.decRes[*idec*].ts	int	if satflag=1,Timestamp where sat. occurred

Variable	Type	Description
eob->aerrdec.decRes[*idec*].x1	int	Internal variable (ask Giles)
eob->aerrdec.decRes[*idec*].x2	int	Internal variable (ask Giles)

J.81 anaerrdec

Variable	Type	Description
eob->aerrdec.NdecRes	u_int	Number of holds errors from decod. - (comp. in eobdec)
eob->aerrdec.decRes[2000]	decResult	holds errors from decod. - (comp. in eobdec)

J.82 pisaCounter

Variable	Type	Description
eob->pisaMon.monXXX[iSample].counts[64]	int	Number of counts per counter

J.83 pisaCounterSample

Variable	Type	Description
eob->pisaMon.sample[*i*].clock	int	
eob->pisaMon.sample[*i*].phase	int	
eob->pisaMon.sample[*i*].effspill[6]	int	
eob->pisaMon.sample[*i*].mom0_jura	float	
eob->pisaMon.sample[*i*].mom0_saleve	float	
eob->pisaMon.sample[*i*].mom1_jura[2]	float	
eob->pisaMon.sample[*i*].mom1_saleve[2]	float	
eob->pisaMon.sample[*i*].mom2_jura[2]	float	
eob->pisaMon.sample[*i*].mom2_saleve[2]	float	

J.84 pisaMonitors

Variable	Type	Description
eob->pisaMon.time	int	Burst time
eob->pisaMon.Nsample	u_int	Number of samples (7.2,2004)

Variable	Type	Description
eob->pisaMon.sample[100]	pisaCounterSample	samples (7.2,2004)
eob->pisaMon.mainz[16]	int	(new 7.2)
eob->pisaMon.NmonJura	u_int	Number of Jura counter samples (backw. compat.)
eob->pisaMon.monJura[10]	pisaCounter	Jura counter samples (backw. compat.)
eob->pisaMon.NmonSaleve	u_int	Number of Saleve counter samples (backw. compat.)
eob->pisaMon.monSaleve[10]	pisaCounter	Saleve counter samples (backw. compat.)
eob->pisaMon.NmonAux	u_int	Number of Aux counter samples (backw. compat.)
eob->pisaMon.monAux[10]	pisaCounter	Aux counter samples (backw. compat.)

J.85 EoBMagnet

Variable	Type	Description
eob->MagnetProbes.time	int	time of measurement
eob->MagnetProbes.hallVsob[6]	float	field values at start of burst
eob->MagnetProbes.hallTsob[6]	float	probe temp at start of burst
eob->MagnetProbes.hallVeob[6]	float	field values at end of burst
eob->MagnetProbes.hallTeob[6]	float	probe temp at end of burst
eob->MagnetProbes.voltmetersob	float	different probe type at sob
eob->MagnetProbes.voltmetereob	float	different probe type at eob

J.86 EndofBurst

Variable	Type	Description
eob->nevent	int	number of events in burst
eob->nwrite	int	number of events written to output
eob->nkill	int	number of events killed by cuts
eob->status	int	status bits of rec program
eob->SGNnreset[3]	int	number of reset vectors
eob->SGNineff[384]	int	inefficiency for each group of 16 wires
eob->SGNeff[384]	int	efficiency for each group of 16 wires
eob->SGNberr[20]	int	number of r/o errors in burst
eob->SGNbadbit	int	plane with most readout errors
eob->L3errctr	L3errorcounter	Level 3 error counter structure
eob->NEventList	int	Nb of events in the list
eob->NTrigWord	u_int	Number of TrigWords (ALL evts in burst)
eob->TrigWord[70000]	int	TrigWords (ALL evts in burst)
eob->NTimeStamp	u_int	Number of TimeStamp (ALL evts in burst)
eob->TimeStamp[70000]	int	TimeStamp (ALL evts in burst)
eob->NDCHDecError	u_int	Number of DCH Dec. error (ALL evts in burst)
eob->DCHDecError[70000]	int	DCH Dec. error (ALL evts in burst)
eob->NLKRHACDecError	u_int	Number of LKR Dec. error (ALL evts in burst)
eob->LKRHACDecError[70000]	int	LKR Dec. error (ALL evts in burst)

Variable	Type	Description
eob->NChargedInfo	u_int	Number of Charged Filter bits (ALL evts in burst)
eob->ChargedInfo[70000]	int	Charged Filter bits (ALL evts in burst)
eob->NNeutralInfo	u_int	Number of Neutral Filter bits (ALL evts in burst)
eob->NeutralInfo[70000]	int	Neutral Filter bits (ALL evts in burst)
eob->NProcError	u_int	Number of Error List of original processing
eob->ProcError[10000]	int	Error List of original processing
eob->NProcErrorRepro	u_int	Number of Error List of reprocessing
eob->ProcErrorRepro[10000]	int	Error List of reprocessing
eob->MagnetProbes	EoBMagnet	Magnet Probes data from End of Burst
eob->pisaMon	pisaMonitors	Pisa Monitors sstructure
eob->aerrdec	anaerrdec	volatile structure for decoded errors
eob->L3spare_int[80]	int	spare space to insert extra variables!
eob->L3spare_float[20]	float	spare space to insert extra variables!

J.87 ke3Event

Variable	Type	Description
ke3->n	int	trigger event number
ke3->timestamp	int	event timestamp
ke3->ecell[49]	float	cell energies (7×7)
ke3->fcell[49]	int	cell flags (7×7)
ke3->adc[45]	int	ADC counts ((3×3)×5)
ke3->ixelec	int	x index of cell hit by e
ke3->iyelec	int	y index of cell hit by e

Variable	Type	Description
ke3->xclus	float	X coord of cluster centre
ke3->yclus	float	Y coord of cluster centre
ke3->eelec	float	energy of electron
ke3->pelec	float	momentum of electron
ke3->xelec	float	extrapolated x position of e on LKR
ke3->yelec	float	extrapolated y position of e on LKR
ke3->apxelec	float	angle of e wrt. LKr projectivity (x proj)
ke3->apyelec	float	angle of e wrt. LKr projectivity (y proj)
ke3->eraw	float	uncorrected cluster energy
ke3->cellsread	float	number of cells read out
ke3->spachacorr	float	corr. applied for spacecharge effect=Q(RLCL+18)
ke3->ecorrke3	float	cl. energy corrected with ke3 factor=Q(RLCL+19)
ke3->e2samp7	float	energy in first smpl of 7*7 cells Q(RLCL+21)
ke3->eHAC	float	HAC Clu. energy behind el.(GeV)=Q(LRKE3+13)
ke3->distHAC	float	min. dist.(HAC clu.-elec track) (cm)=Q(LRKE3+14)
ke3->npmuv	int	number of hits PMUV Q(LRKE3+15)
ke3->eCluNear	float	Closest cluster: E(GeV)=Q(LRLCL+4)
ke3->idCluNear	int	Closest cluster: cell max id=Q(LRLCL+3)
ke3->xCluNear	float	Closest cluster: x(cm)=Q(LRCL+7)
ke3->yCluNear	float	Closest cluster: y(cm)=Q(LRCL+8)

Variable	Type	Description
ke3->pPion	float	Pion: momentum(GeV)=Q(LRKE3+10)
ke3->xPion	float	Pion: x(cm) (extrap. at LKR)=Q(LRKE3+11)
ke3->yPion	float	Pion: y(cm) (extrap. at LKR)=Q(LRKE3+12)
ke3->zvertex	float	z position of vertex
ke3->spnt[4]	DCHspacepoint	space points in 4 chambers

J.88 kmu3Event

Variable	Type	Description
kmu3->Nnut	u_int	Number of detailed NUT information for kum3 data
kmu3->nut[1]	NUTkmu3	detailed NUT information for kum3 data
kmu3->cmpevt	cmpEvent	COmPACT event structure

J.89 mcPlaneTrak

Variable	Type	Description
evt->mcsim[isim].part[ipart].plane[ipl].icode	int	plane code - NOT USED from 4.0
evt->mcsim[isim].part[ipart].plane[ipl].xyz[3]	float	NOT USED from 4.0
evt->mcsim[isim].part[ipart].plane[ipl].zPlane	float	z of Plane = Q(LTRAK+x+4)
evt->mcsim[isim].part[ipart].plane[ipl].dxdz	float	dx/dz in plane= Q(LTRAK+x+5)
evt->mcsim[isim].part[ipart].plane[ipl].dydz	float	dy/dz in plane= Q(LTRAK+x+6)
evt->mcsim[isim].part[ipart].plane[ipl].Edep	float	depos.Ener.= Q(LTRAK+x+7)
evt->mcsim[isim].part[ipart].plane[ipl].accept	int	accept. flag= Q(LTRAK+x+9)

J.90 mcParticle

Variable	Type	Description
evt->mcsim[isim].part[ipart].type	int	particle type=Q(LPART+1)
evt->mcsim[isim].part[ipart].p[4]	float	four-mom.=Q(LPART+2,3,4,5)
evt->mcsim[isim].part[ipart].pvertex[3]	float	prod. vertex=Q(LPART+9,10,11)
evt->mcsim[isim].part[ipart].dvertex[3]	float	decay vertex=Q(LPART+12,13,14)
evt->mcsim[isim].part[ipart].spin[3]	float	spin of generated particle -x,y,z-
evt->mcsim[isim].part[ipart].xbmag[3]	float	pos. bef. magnet - NOT USED
evt->mcsim[isim].part[ipart].xamag[3]	float	pos. aft. magnet - NOT USED
evt->mcsim[isim].part[ipart].pbmag[3]	float	bef. magnet - NOT USED
evt->mcsim[isim].part[ipart].pamag[3]	float	mom. aft. magnet - NOT USED
evt->mcsim[isim].part[ipart].icodes	int	NOT USED from 4.0
evt->mcsim[isim].part[ipart].Nplane	u_int	Number of Traking planes
evt->mcsim[isim].part[ipart].plane[20]	mcPlaneTrak	Traking planes

J.91 mcSIM

Variable	Type	Description
evt->mcsim[isim].mcType	int	MC type (NMC/NASIM)=Q(LSIM+2)
evt->mcsim[isim].mcVersion	int	version of MC used=Q(LSIM+3)
evt->mcsim[isim].geoVersion	int	version of geometry used=Q(LSIM+4)
evt->mcsim[isim].shwVersion	int	version of shower library used=Q(LSIM+5)
evt->mcsim[isim].simYear	int	data year simulated=Q(LSIM+1)

Variable	Type	Description
evt->mcsim[isim].accRate	float	accidental rate=Q(LSIM+18)
evt->mcsim[isim].ksklRatio	float	K_S/K_L ratio=Q(LSIM+16)
evt->mcsim[isim].cnRatio	float	charged/neutral decay ratio=Q(LSIM+17)
evt->mcsim[isim].randSeed[3]	int	random seeds=Q(LSIM+7,8,9)
evt->mcsim[isim].mcword[4]	int	Q(LSIM+10,11,12,13) (NASIM $\neq$ NMC)
evt->mcsim[isim].nasim[4]	int	NOT USED from 4.0
evt->mcsim[isim].time	int	event time=Q(LEVT+6)
evt->mcsim[isim].magnetSim	int	magnet simulation used=Q(LSIM+6)
evt->mcsim[isim].options	int	MC options=Q(LSIM+15)
evt->mcsim[isim].Npart	u_int	Number of 'true' particle data
evt->mcsim[isim].part[20]	mcParticle	'true' particle data

J.92 mcEvent

Variable	Type	Description
evt->Nmcsim	u_int	Number of SIM structure
evt->mcsim[5]	mcSIM	SIM structure
evt->cmpevt	cmpEvent	COmPACT event structure

J.93 DETstatus

Variable	Type	Description
sevt->DETstatus.AKL	int	AKL bit coded status
sevt->DETstatus.DCH	int	DCH bit coded status
sevt->DETstatus.HOD	int	CHOD bit coded status
sevt->DETstatus.HAC	int	HAC bit coded status
sevt->DETstatus.LKR	int	LKR bit coded status

Variable	Type	Description
sevt->DETstatus.KAB	int	KAB bit coded status
sevt->DETstatus.NHO	int	NHOD bit coded status
sevt->DETstatus.MUV	int	MUV bit coded status
sevt->DETstatus.MBX	int	MBX bit coded status
sevt->DETstatus.NTR	int	NTR bit coded status
sevt->DETstatus.LV3	int	L3 filter bits (0-15: neutral, 16-31: charged)
sevt->DETstatus.LV3Trig	int	L3trigword[0](bits 0-15) + L3FilterDownScale(bits 16-31)
sevt->DETstatus.LV3TrigRare	int	L3trigword[1](bits 0-31)
sevt->DETstatus.LV3ABTrig	int	L3ONLINEtrigword[0](bits 0-15)+L3Btrigword[0](bits 16-31)
sevt->DETstatus.LV3ATrigRare	int	L3ONLINEtrigword[1](bits 0-31)
sevt->DETstatus.LV3BTrigRare	int	L3Btrigword[1](bits 0-31)
sevt->DETstatus.ChTrEff[10]	int	Charged Trigger Efficiency

J.94 KABstrack

Variable	Type	Description
sevt->KABtrak[*track*].p	float	track momentum
sevt->KABtrak[*track*].q	int	track charge
sevt->KABtrak[*track*].UPorDOWN	int	first station UP=1 or DOWN=2
sevt->KABtrak[*track*].perr	float	error on momentum
sevt->KABtrak[*track*].chi2	float	χ^2 (quality) of track
sevt->KABtrak[*track*].x	float	x position in second station
sevt->KABtrak[*track*].y	float	y position in second station
sevt->KABtrak[*track*].xUorD	float	x position in UP or Down station

Variable	Type	Description
sevt->KABtrak[*track*].yUorD	float	y position in UP or Down station
sevt->KABtrak[*track*].time	float	track time
sevt->KABtrak[*track*].timeUorD	float	track time at UP or Down station
sevt->KABtrak[*track*].timeST2	float	track time at second station
sevt->KABtrak[*track*].dxdz	float	dx/dz
sevt->KABtrak[*track*].dydz	float	dy/dz
sevt->KABtrak[*track*].sigxx	float	Squared error on x (cm^2)
sevt->KABtrak[*track*].sigyy	float	Squared error on y (cm^2)
sevt->KABtrak[*track*].sigtt	float	Squared error on time (ns^2)
sevt->KABtrak[*track*].sigdxdx	float	Squared error on dx/dz
sevt->KABtrak[*track*].sigdydy	float	Squared error on dx/dz
sevt->KABtrak[*track*].RecFlag	int	reconstruction flag
sevt->KABtrak[*track*].anavar[20]	float	Provision for analysis variables
sevt->KABtrak[*track*].anaflag[5]	int	Provision for analysis flags

J.95 trak

Variable	Type	Description
sevt->track[*track*].p	float	momentum
sevt->track[*track*].q	int	charge
sevt->track[*track*].quality	float	track quality
sevt->track[*track*].chi2	float	track chi2
sevt->track[*track*].bx	float	x-coordinate before magnet
sevt->track[*track*].by	float	y-coordinate before magnet
sevt->track[*track*].bdxdz	float	dx/dz before magnet
sevt->track[*track*].bdydz	float	dy/dz before magnet
sevt->track[*track*].x	float	x-coordinate after magnet

Variable	**Type**	**Description**
sevt->track[*track*].y	float	y-coordinate after magnet
sevt->track[*track*].dxdz	float	dx/dz after magnet
sevt->track[*track*].dydz	float	dy/dz after magnet
sevt->track[*track*].perr	float	Error on track momentum
sevt->track[*track*].sigxx	float	Error on x (cm)=Q(RDTK+11)
sevt->track[*track*].sigyy	float	Error on y (cm)=Q(RDTK+12)
sevt->track[*track*].sigdxdx	float	Error on dx/dz =Q(RDTK+13)
sevt->track[*track*].sigdydy	float	Error on dy/dz =Q(RDTK+14)
sevt->track[*track*].sigxdx	float	NOT FILLED - correlation sigma(x,dx/dz)=Q(RDTK+28)
sevt->track[*track*].sigxy	float	correlation sigma(x,y)=Q(RDTK+29)
sevt->track[*track*].sigdxy	float	correlation sigma(dx/dz,y)=Q(RDTK+30)
sevt->track[*track*].sigxdy	float	correlation sigma(x,dy/dz)=Q(RDTK+31)
sevt->track[*track*].sigdxdy	float	correlation sigma(dx/dz,dy/dz)=Q(RDTK+32)
sevt->track[*track*].sigydy	float	correlation sigma(y,dy/dz) =Q(RDTK+33)
sevt->track[*track*].time	float	track time
sevt->track[*track*].hodTime	float	track time in hodoscope
sevt->track[*track*].hodstatus	int	hod. flags used for time calc.=Q(RDTK+26)
sevt->track[*track*].aklTime	float	nearest akl hit around cluster time (see doc)
sevt->track[*track*].aklTime67	float	time of AKL hit closest to event time (see doc)
sevt->track[*track*].aklflag	int	records th nuber of hit in various timeslices

Variable	Type	Description
sevt->track[*track*].iMuon	int	index of assciated muon (-1. if none)
sevt->track[*track*].iHac	int	associated HAC cluster
sevt->track[*track*].iClus	int	assoc. LKR clu. $0 \leq index < Ncluster$ (-1 if none)
sevt->track[*track*].dDeadCell	float	distance to nearest LKR dead cell
sevt->track[*track*].hitPattern	int	one bit per wire for efficiency studies
sevt->track[*track*].efficiency[2]	int	bit coded words for eff. studies
sevt->track[*track*].spare[5]	float	spare variables

J.96 cluster

Variable	Type	Description
sevt->cluster[*iclus*].energy	float	cluster energy
sevt->cluster[*iclus*].x	float	x-coordinate of cluster
sevt->cluster[*iclus*].y	float	y-coordinate of cluster
sevt->cluster[*iclus*].time	float	cluster time
sevt->cluster[*iclus*].hacTime	float	nearest hac hit around cluster time
sevt->cluster[*iclus*].hodTime	float	nearest ch. hodoscope hit around cluster time
sevt->cluster[*iclus*].aklTime	float	nearest akl hit around cluster time (see doc)
sevt->cluster[*iclus*].aklTime67	float	time of AKL hit closest to event time (see doc)
sevt->cluster[*iclus*].aklflag	int	number of hits in various time slices
sevt->cluster[*iclus*].dDeadCell	float	distance to closest dead cell
sevt->cluster[*iclus*].status	int	status bits
sevt->cluster[*iclus*].iTrack	int	index of associated track (-1 if none)
sevt->cluster[*iclus*].rmsx	float	x cluster width (lkr-¿rmsx) - NOT filled for ε'

Variable	Type	Description
sevt->cluster[*iclus*].rmsy	float	y cluster width (lkr-¿rmsy) - NOT filled for ε'
sevt->cluster[*iclus*].mctailcorr	float	energy tail corr from MC
sevt->cluster[*iclus*].spare[5]	float	spare variables

J.97 vtxtracks

Variable	Type	Description
sevt->vtx[*ivertex*].vtxtracks[*itrack].iTrack*	int	index of track
sevt->vtx[*ivertex*].vtxtracks[*itrack].bdxdz*	float	corrected dxdz before the magnet at the vertex
sevt->vtx[*ivertex*].vtxtracks[*itrack].bdydz*	float	corrected dydz before the magnet at the vertex

J.98 SCvertex

Variable	Type	Description
sevt->vtx[*ivertex*].Nvtxtrack	u_int	Number of track data in this vertex
sevt->vtx[*ivertex*].vtxtrack[7]	vtxtracks	track data in this vertex
sevt->vtx[*ivertex*].charge	int	total charge of this vertex
sevt->vtx[*ivertex*].cda	float	closest approach
sevt->vtx[*ivertex*].cog	float	cog (cm)
sevt->vtx[*ivertex*].chi2	float	χ^2 of vertex
sevt->vtx[*ivertex*].x	float	x position
sevt->vtx[*ivertex*].y	float	y position
sevt->vtx[*ivertex*].z	float	z position
sevt->vtx[*ivertex*].tVtx	float	vertex time from spectrometer times
sevt->vtx[*ivertex*].tVtxHodo	float	Vertex time defined by the hodoscope
sevt->vtx[*ivertex*].hacTime	float	closest hit in the hac around tVtx
sevt->vtx[*ivertex*].aklTime	float	closest hit of AkL around tVtx

Variable	Type	Description
sevt->vtx[*ivertex*].aklTime67	float	closest hit of AkL around tVtx (see doc)
sevt->vtx[*ivertex*].aklflag	int	aklflag as defined in fuser_akl.F. It is used for both aklTime and aklTime67
sevt->vtx[*ivertex*].anavar[5]	float	Provision for analysis variables
sevt->vtx[*ivertex*].anaflag[5]	int	Provision for analysis flags

J.99 muon

Variable	Type	Description
sevt->muon[*imuon*].x	float	x position
sevt->muon[*imuon*].y	float	y position
sevt->muon[*imuon*].time	float	time from muon detector
sevt->muon[*imuon*].chi2	float	chi2 from the muon reconstruction
sevt->muon[*imuon*].status	int	coded for hit planes
sevt->muon[*imuon*].plane[2]	int	bit coded for hit map
sevt->muon[*imuon*].iTrk	int	track index

J.100 pmuon

Variable	Type	Description
sevt->pmuon[*ipmuon*].time	float	time of the photomultiplier channel.
sevt->pmuon[*ipmuon*].channel	int	photo multiplier channel

J.101 hacclus

Variable	Type	Description
sevt->hacclus[*ihacclus*] .energy	float	energy of cluster
sevt->hacclus[*ihacclus*] .x	float	x position of cluster
sevt->hacclus[*ihacclus*] .y	float	y position of cluster
sevt->hacclus[*ihacclus*] .time	float	time of cluster
sevt->hacclus[*ihacclus*] .bfratio	float	back to front ratio

J.102 SCDCHmult

Variable	Type	Description
sevt->SCDCHEFFmult.MBXPlaneEff[6]	int	number of hit for plane packed
sevt->SCDCHEFFmult.L1Trk24Eff[2]	int	multiplicity in DCH1 for L1 trigger packed
sevt->SCDCHEFFmult.DCHSnowErr[2]	int	16 bit for DCH snow effect(2-1)(4-3)
sevt->SCDCHEFFmult.MBXmult	int	multiplicity in DCH
sevt->SCDCHEFFmult.Trk24ON	int	L1Trk24=1 –¿ trigger OK

J.103 PUtslice

Variable	Type	Description
sevt->pu[*tslice*].chan[16]	int	16 channels * 24 bits x 10 tslice

J.104 MaChit

Variable	Type	Description
sevt->mch[*hit*].counter	int	(ONLY MaC evs) counter number(1-8) Q(PKSM+x+2)
sevt->mch[*hit*].pheight	float	(ONLY MaC evs) pulse height(GeV) = Q(PKSM+x+3)
sevt->mch[*hit*].time	float	(ONLY MaC evs) time of hit(ns) = Q(PKSM+x+4)
sevt->mch[*hit*].pdsflag	int	(ONLY MaC evs) decoding flag=Q(PKSM+x+1)

J.105 SRNDMsummary

Variable	Type	Description
sevt->rndm.type	int	type of random used(KS=1, KL=2) = Q(LPRE+1)
sevt->rndm.timestamp	int	timestamp of random used = Q(LPRE+3)
sevt->rndm.run	int	Run Number of random used = Q(LPRE+9)
sevt->rndm.burst	int	Burst time stamp of random used = Q(LPRE+10)
sevt->rndm.Nused	int	Number of times this RNDM evt was used so far
sevt->rndm.SPSphase	float	SPS phase of random evt
sevt->rndm.mainphase	float	main(50Hz) phase of random evt
sevt->rndm.tovrflw[64]	float	time of nearest ovflw to rndm evt
sevt->rndm.KLmondNdt	float	intensity in KLmon for random event
sevt->rndm.KSmondNdt	float	intensity in KSmon for random event
sevt->rndm.QXdNdt	float	intensity in QX for random event

J.106 superCmpEvent

Variable	Type	Description
sevt->dbErr	int	non-zero if error from SQLITE db
sevt->CMPstatus	int	COmPACT analysis bit coded status
sevt->FlagCorr	int	which corrections were applied to this event
sevt->CmpFilter	int	which filter was applied to compact event
sevt->DETstatus[1]	DETstatus	subdetector bit coded status

Variable	Type	Description
sevt->rndm	SRNDMsummary	RaNDoM events summary
sevt->nEvt	int	trigger event number
sevt->trigWord	int	trigger word
sevt->timeStamp	int	time stamp
sevt->nTrigBef	int	number of triggers in interval preceding this event
sevt->timeToPrev	int	time to previous event
sevt->SPSPhase	float	SPS phase to 40MHz clock
sevt->MainsPhase	float	Mains 50Hz phase to 40MHz clock
sevt->SPSPhaseRaw	int	raw SPS phase
sevt->MainsPhaseRaw	int	raw mains phase
sevt->KLmondNdt	float	intensity in KLmon
sevt->QXdNdt	float	intensity in QX
sevt->pu[10]	PUtslice	PMB pattern unit data
sevt->Nmch	u_int	Number of Mannelli counter hit data
sevt->mch[20]	MaChit	Mannelli counter hit data
sevt->LKRdownscaled	int	LKR downscale flag (=1 downsc., =0 not downsc.)
sevt->LKRenergy	float	total energy in LKR
sevt->HACenergy	float	total energy in HAC
sevt->NKABstrak	u_int	Number of charged object - Kabes tracks
sevt->KABstrak[50]	KABstrack	charged object - Kabes tracks
sevt->Ntrack	u_int	Number of charged object - tracks
sevt->track[40]	trak	charged object - tracks
sevt->Nvtx	u_int	Number of charged object -vertices
sevt->vtx[100]	SCvertex	charged object -vertices

Variable	Type	Description
sevt->Nmuon	u_int	Number of charged object - muon
sevt->muon[12]	muon	charged object - muon
sevt->Npmuon	u_int	Number of photomultiplier muon hits
sevt->pmuon[56]	pmuon	photomultiplier muon hits
sevt->Nhacclus	u_int	Number of HAC clusters
sevt->hacclus[25]	hacclus	HAC clusters
sevt->Ncluster	u_int	Number of LKR objects - clusters
sevt->cluster[25]	cluster	LKR objects - clusters
sevt->SCDCHEFFmult	SCDCHmult	structure for MBX eff study
sevt->tsPrev	int	previous trigger timestamp
sevt->tsNext	int	next trigger timestamp
sevt->twPrev	int	previous trigger trigger word
sevt->twNext	int	next trigger trigger word
sevt->spareInt[2]	int	2 spare integers
sevt->spareFloat[2]	float	2 spare floats

J.107 STSscal

Variable	Type	Description
sbur->TSscal.Nscaler	u_int	Number of TS scalers
sbur->TSscal.scaler[100]	int	TS scalers

J.108 SL2TSscal

Variable	Type	Description
sbur->L2TSscal.Nscaler	u_int	Number of L2 TS scalers
sbur->L2TSscal.scaler[20]	int	L2 TS scalers

J.109 superBurst

Variable	Type	Description
sbur->majorVer	int	COmPACT major version number
sbur->minorVer	int	COmPACT minor version number
sbur->patch	int	COmPACT patch number
sbur->time	int	burst time
sbur->brtype	int	Type of burst (data/MC/..)
sbur->nrun	int	run number
sbur->NTrigWord	int	Number of trig word (all evts in burst)
sbur->intensity	int	number of protons on T10
sbur->intensityT4	int	number of protons on T4
sbur->ctauOrigin	int	z(mm) used to compute $c.\tau$
sbur->tOffstVer	int	tim. offset version used to prod. SC
sbur->MNP33current	int	current of the spectrometer magnet
sbur->Bend1current	int	current of achromat 1
sbur->flags	int	Flags for turning on and of correction routines
sbur->TSscal	STSscal	TS scalars
sbur->L2TSscal	SL2TSscal	L2TS scalars
sbur->dataset	int	Data set value
sbur->tOffst	superTimeOffset	Structure holding time offsets
sbur->BadB	BadBurst	Structure holding Bad Burst data
sbur->dberr	int	Database error code (0=OK;-1=NOK)

J.110 superPisaCounter

Variable	Type	Description
seob->sPisaMon.smonXXX.average[64]	float	average over 10 samples for each counter

Variable	Type	Description
seob->sPisaMon.smonXXX.stddev[64]	float	standard deviation over 10 samples for each counter

J.111 superPisaMonitors

Variable	Type	Description
seob->sPisaMon.juraCounts[64]	int	sum of Jura counter information
seob->sPisaMon.saleveCounts[64]	int	sum of Saleve counter information
seob->sPisaMon.auxCounts[64]	int	sum of Aux counter information

J.112 superPisaMonitors04

Variable	Type	Description
seob->pisaMon.time	int	Burst time
seob->pisaMon.Nsample	u_int	Number of samples, same as eob (7.2,2004)
seob->pisaMon.sample[100]	pisaCounterSample	samples, same as eob (7.2,2004)
seob->pisaMon.mainz[16]	int	(new 7.2)

J.113 superEndofBurst

Variable	Type	Description
seob->magic	int	Magic number
seob->DCHplanehit[32]	int	inefficiency for each group of 16 wires
seob->DCHcardhit[512]	int	efficiency for each group of 16 wires
seob->sMagnetProbes	EoBMagnet	Magnet Probes data from End of Burst
seob->sPisaMon	superPisaMonitors	Pisa Monitors structure (2003)
seob->pisaMon	superPisaMonitors04	Pisa Monitors structure (2004)

J.114 superMcParticle

Variable	Type	Description
evt->part[ipart].type	int	particle type=Q(LPART+1)
evt->part[ipart].p[4]	float	four-mom.=Q(LPART+2,3,4,5) (E,px,py,pz)
evt->part[ipart].pvertex[3]	float	prod. vertex=Q(LPART+9,10,11) (x,y,z)
evt->part[ipart].dvertex[3]	float	decay vertex=Q(LPART+12,13,14) (x,y,z)
evt->part[ipart].spin[3]	float	spin of generated particle =Q(LPART+) (x,y,z)

J.115 superMcDecay

Variable	Type	Description
evt->decay.Ktype	int	Kaon type
evt->decay.Dtype	int	Decay type
evt->decay.dvertex[3]	float	Decay vertex (x,y,z)
evt->decay.p[4]	float	Kaon four-mom. (E,px,py,pz)

J.116 superMcEvent

Variable	Type	Description
evt->decay	superMcDecay	MC event information
evt->Npart	u_int	Number of For each particle
evt->part[20]	superMcParticle	For each particle
evt->scmpevt	superCmpEvent	superCOmPACT event structure

J.117 hyperBurst

Variable	Type	Description
hbur->majorVer	int	COmPACT major version number
hbur->minorVer	int	COmPACT minor version number
hbur->BadB	BadBurst	Structure holding Bad Burst data
hbur->tOffst	superTimeOffset	Structure holding time offsets
hbur->time	int	burst time
hbur->brtype	int	Type of burst (data/MC/..)
hbur->nrun	int	run number
hbur->sample	int	sample number
hbur->flag	int	burst status flag
hbur->spare[2]	float	2 spare floats

J.118 hyperKabTrk

Variable	Type	Description
hevt->kabtrk[*iktrk*]p	float	track momentum
hevt->kabtrk[*iktrk*]q	int	track charge
hevt->kabtrk[*iktrk*]UPorDOWN	int	first station UP=1 or DOWN=2
hevt->kabtrk[*iktrk*]perr	float	error on momentum
hevt->kabtrk[*iktrk*]chi2	float	χ^2 (quality) of track
hevt->kabtrk[*iktrk*]x	float	x position in second station
hevt->kabtrk[*iktrk*]y	float	y position in second station
hevt->kabtrk[*iktrk*]yUorD	float	y position in UP or Down station
hevt->kabtrk[*iktrk*]time	float	track time
hevt->kabtrk[*iktrk*]dxdz	float	dx/dz
hevt->kabtrk[*iktrk*]dydz	float	dy/dz

J.119 hyperCluster

Variable	Type	Description
hevt->cluster[*iclus*].energy	float	energy
hevt->cluster[*iclus*].x	float	X position
hevt->cluster[*iclus*].y	float	Y position
hevt->cluster[*iclus*].time	float	time
hevt->cluster[*iclus*].rmsx	float	RMS X
hevt->cluster[*iclus*].rmsy	float	RMS Y
hevt->cluster[*iclus*].dDeadCell	float	distance to nearest dead cell
hevt->cluster[*iclus*].dTrack	float	dist from closest track
hevt->cluster[*iclus*].flag	int	flag
hevt->cluster[*iclus*].spare[2]	float	2 spare floats

J.120 hyperTrack

Variable	Type	Description
hevt->track[*itrack*].q	int	charge
hevt->track[*itrack*].p	float	momentum
hevt->track[*itrack*].perr	float	momentum error
hevt->track[*itrack*].time	float	track time form dch
hevt->track[*itrack*].hodotime	float	track time from hodoscope
hevt->track[*itrack*].bx	float	x position before the magnet
hevt->track[*itrack*].by	float	y position before the magnet
hevt->track[*itrack*].x	float	x position after the magnet
hevt->track[*itrack*].y	float	y position after the magnet
hevt->track[*itrack*].bdxdz	float	dx/dz before magnet
hevt->track[*itrack*].bdydz	float	dy/dz before magnet
hevt->track[*itrack*].nbfbdxdz	float	dx/dz before magnet w/o bluefield
hevt->track[*itrack*].nbfbdydz	float	dy/dz after magnet w/o bluefield
hevt->track[*itrack*].dxdz	float	dx/dz after magnet
hevt->track[*itrack*].dydz	float	dy/dz after magnet

Variable	Type	Description
hevt->track[*itrack*].eop	float	E over p for track
hevt->track[*itrack*].flag	int	flags
hevt->track[*itrack*].spare[2]	float	2 spare floats

J.121 hyperVertex

Variable	Type	Description
hevt->vtx[*vtx*].q	int	sum of charge
hevt->vtx[*vtx*].mass	float	invariant kaon mass
hevt->vtx[*vtx*].p	float	momentum of kaon
hevt->vtx[*vtx*].cda	float	Closest distance of apporach
hevt->vtx[*vtx*].pos[3]	float	x,y,z of vertex
hevt->vtx[*vtx*].spare[2]	float	2 spare floats

J.122 hyperMcParticle

Variable	Type	Description
hevt->part[ipart].type	int	particle type (NASIM)
hevt->part[ipart].p[4]	float	four-mom. (E,px,py,pz)
hevt->part[ipart].pvertex[3]	float	prod. vertex (x,y,z)
hevt->part[ipart].dvertex[3]	float	decay vertex (x,y,z)

J.123 hyperCmpEvent

Variable	Type	Description
hevt->dbErr	int	non-zero if error from SQLITE db
hevt->flag	int	flag bit 0 on neutral, charged otherwise
hevt->timestamp	int	time stamp
hevt->spsphase	float	sps phase
hevt->mainsphase	float	mains phase
hevt->trigword	int	trigger word

Variable	Type	Description
hevt->PUpack[3]	int	packed Pattern Units bits (access with ...)
hevt->pu04[3]	int	pattern unit 4, 3 time slots (not filled from in 7.1
hevt->pu14[3]	int	pattern unit 14, 3 time slots (not filled from in 7.1
hevt->pu06[3]	int	pattern unit 6, 3time slots (not filled in 7.1)
hevt->Ncluster	u_int	Number of cluster info. Not filled for charged evts.
hevt->cluster[4]	hyperCluster	cluster info. Not filled for charged evts.
hevt->Ntrack	u_int	Number of hyper tracks
hevt->track[3]	hyperTrack	hyper tracks
hevt->vtx[1]	hyperVertex	hyper vertex
hevt->kabtrk[1]	hyperKabTrk	hyper kabes track
hevt->NmcPart	u_int	Number of MC generated particles
hevt->mcPart[20]	hyperMcParticle	MC generated particles
hevt->spare[2]	float	2 spare floats

J.124 GeomPhaSpa

Variable	Type	Description
Geom->BeamK(l/s).x	float	phase space dispersion *10**3: x
Geom->BeamK(l/s).y	float	phase space dispersion *10**3: y
Geom->BeamK(l/s).dx	float	phase space dispersion *10**3: dx
Geom->BeamK(l/s).dy	float	phase space dispersion *10**3: dy
Geom->BeamK(l/s).angle[2]	float	beam angle: [0]: absolute; [1]: relative

J.125 GeomPos1

Variable	Type	Description
Geom->X.x	float	x position in cm
Geom->X.y	float	y position in cm
Geom->X.z	float	z position in cm
Geom->X.r	float	Radius in cm
Geom->X.l	float	toto

J.126 GeomPos2

Variable	Type	Description
Geom->Detx.x	float	x position in cm
Geom->Detx.y	float	y position in cm
Geom->Detx.z	float	z position in cm
Geom->Detx.rin	float	Rin in cm
Geom->Detx.OuterOctagon[2]	float	Outer Octagon (square) 2 values
Geom->Detx.thickness	float	Thickness in

J.127 GeomPos3

Variable	Type	Description
Geom->X.x	float	x position in cm
Geom->X.y	float	y position in cm

Variable	Type	Description
Geom->X.z	float	z position in cm
Geom->X.angle	float	Rotation

J.128 GeomPosAkl

Variable	Type	Description
Geom->AklPocket[ipock].x	float	x position in cm
Geom->AklPocket[ipock].y	float	y position in cm
Geom->AklPocket[ipock].z	float	z position in cm
Geom->AklPocket[ipock].InnerOctagon[2]	float	Inner Octagon
Geom->AklPocket[ipock].OuterOctagon[2]	float	Outer Octagon

J.129 GeomPosStrip

Variable	Type	Description
Geom->Kab[ikab].Strip[istrip].x	float	x position in cm
Geom->Kab[ikab].Strip[istrip].y	float	y position in cm
Geom->Kab[ikab].Strip[istrip].z	float	z position in cm

J.130 GeomKabes

Variable	Type	Description
Geom->Kab[ikab].PosKabes	GeomPos3	Kabes station position
Geom->Kab[ikab].Strip[2]	GeomPosStrip	Position of first strip in each detector

J.131 GeomPosPlane

Variable	Type	Description
Geom->Dch[ich].Plane[iplane].x	float	x position in cm
Geom->Dch[ich].Plane[iplane].y	float	y position in cm
Geom->Dch[ich].Plane[iplane].z	float	z position in cm
Geom->Dch[ich].Plane[iplane].x1	float	x1 position in cm
Geom->Dch[ich].Plane[iplane].angle	float	angle of plane (in degrees)

J.132 GeomChamber

Variable	Type	Description
Geom->Dch[ich].PosChamber	GeomPos2	Chamber position
Geom->Dch[ich].PosPlane[8]	GeomPosPlane	Planes positions

J.133 GeomDecay

Variable	Type	Description
Geom->Decay.zNeutral	float	Begining of neutral decay region
Geom->Decay.zCharged	float	Begining of charged decay region

J.134 GeomVirtualZ

Variable	Type	Description
Geom->Det.z	float	Virtual point after the magnet
Geom->Det.bz	float	Virtual point before the magnet

J.135 GeomCompact

Variable	Type	Description
Geom->BeamKl	GeomPhaSpa	KL beam phase space
Geom->BeamKs	GeomPhaSpa	KS beam phase space
Geom->KlTarget	GeomPos1	KL target position
Geom->KlDefColl	GeomPos1	KL defining collimator position
Geom->KlDefCollExit	GeomPos1	KL defining coll. exit position
Geom->KlFinColl	GeomPos1	KL final collimator position
Geom->KlFinCollExit	GeomPos1	KL final coll. exit position
Geom->KsTarget	GeomPos1	KS target position

Variable	Type	Description
Geom->KsFinColl	GeomPos1	KS final collimator position
Geom->KsFinCollExit	GeomPos1	KS final coll. exit position
Geom->AksConverter	GeomPos1	AKS converter position
Geom->AksCounter	GeomPos1	AKS counter position
Geom->AksConvThick	float	AKS converter thickness
Geom->AklPocket[7]	GeomPosAkl	AKL pocket position
Geom->Dch[4]	GeomChamber	Drift Chamber geometrical description
Geom->Magnet	GeomPos2	Magnet position
Geom->Hodoscope	GeomPos2	Hodoscope geometrical description
Geom->Lkr	GeomPos2	LKR calo geometrical description
Geom->Hac	GeomPos2	Hac calo geometrical description
Geom->Muv1	GeomPos2	Muon veto 1 geometrical description
Geom->Muv2	GeomPos2	Muon veto 2 geometrical description
Geom->Muv3	GeomPos2	Muon veto 3 geometrical description
Geom->Kab[3]	GeomKabes	Kabes geometry description
Geom->Decay	GeomDecay	Begining of decay region
Geom->DCH	GeomVirtualZ	virtual points for vertex of the driftchambers

In compact the run structure is called rdb; data are accessed by `rdb->nBur` for instance. In compact the burst structure is called bdb; data are accessed by `bdb->time` for instance.

```
/*------------------------------------------------------*/
/* Structure definitions used to provide "user-friendly"  */
/* names for the data returned via void * from CDB       */
/*                                                      */
/*                                   Bruce Hay 3.12.98  */
/*------------------------------------------------------*/


/*------------------------------------------------------*/
/* Run structures:                                      */
/*------------------------------------------------------*/

typedef struct {
  int   dummy;

  /* General info: ---------------------------------------*/

  int   nBur;          /* no. bursts */              # NBSTS_ALL
  int   nBurBad;       /* no. of bad bursts */       # NBSTS_BAD
  int   typBad;        /* type of bad bursts */      # BADR

  /* Good events: ----------------------------------------*/

  int   nKsPiPi;       /* no. of Ks Pi+Pi- */        # RUN_KSPIMPIP
  int   nKlPiPi;       /* no. of Kl Pi+Pi- */        # RUN_KLPIMPIP
  int   nKsPi0Pi0;     /* no. of Ks Pi0Pi0 */        # RUN_KSPI0PI0
  int   nKlPi0Pi0;     /* no. of Kl Pi0Pi0 */        # RUN_KLPI0PI0

  /* Charged performance: --------------------------------*/

  float ovfRand;       /* % overflow in random events */  # NRNDOF    $$
  float ovfPi0Pi0;     /* % overflow in Pi0Pi0 events */  # NPI0OF    $$
  float massKs;        /* Ks->Pi+Pi- mass */              # FITMKS(2) $$
  float merrKs;        /* Ks->Pi+Pi- mass resolution */   # FITMKS(3) $$
  float resPt2;        /* Pt2 resolution */               # RESPT2    $$
  float resBkC;        /* residual background estimator */# RSBC      $$
  float mbxDead;       /* Massbox deadtime */             # DEAD      $$
  float tEffC;         /* trigger efficiency */           # TEFFCH    $$

  /* Neutral performance: --------------------------------*/
```

```
  float massPi0;      /* Pi0->gg mass */                 # FITPI0(2) $$
  float merrPi0;      /* Pi0->gg mass resolution */      # FITPI0(3) $$
  float resEll;       /* Rellipse resolution */          # RESELL    $$
  float resBkN;       /* residual background estimator */# RSBN      $$
  float tEffN;        /* trigger efficiency */           # TEFFNT    $$

  /* Tagger: -----------------------------------------------*/

  float tagEff;       /* tagger efficiency */            # TAGEFF    $$
  float tagDil;       /* tagger dilution */              # TAGDIL    $$

  float multTs;       /* % multiple time-stamp events */ # MULT

  /* Beam: -------------------------------------------------*/

  float totT10;       /* intensity: total # p on T10 */  # RUN_INTENS

} runMap;


/*---------------------------------------------------------*/
/* Burst structures:                                       */
/*---------------------------------------------------------*/

typedef struct {
  float Lkr;          /* % Lkr Xoff time */
  float Dch;          /* % Dch Xoff time */
  float Nut;          /* % Nut Xoff time */
  float Hac;          /* % Hac Xoff time */
  float Akl;          /* % Akl Xoff time */
  float Tag;          /* % Tag Xoff time */
  float Muv;          /* % Muv Xoff time */
  float Hod;          /* % Hod Xoff time */
} Xoff;
                   # BUR_XOFFLKR_FRACTIMEON, BUR_XOFFDCH_FRACTIMEON,
                     BUR_XOFFNUT_FRACTIMEON, BUR_XOFFHAC_FRACTIMEON,
                     BUR_XOFFAKL_FRACTIMEON, BUR_XOFFTAG_FRACTIMEON,
                     BUR_XOFFMUV_FRACTIMEON, BUR_XOFFHOD_FRACTIMEON

typedef struct {

  /* General info: -----------------------------------------*/

  int   time;         /* burst time: 32-bit Unix time */ # BUR_TIME
  int   badBur;       /* bad burst: 1 bit/system */      # BADB
```

```
int   nEvt;         /* no. events in L3 list */        # BUR_NTRIGWORD
float intT10;       /* intensity: ppp on T10 */        # INTENSITY


/* Charged chain: --------------------------------------*/

int   nWDog;        /* no. watchdogs in L3 list */     # BST_L3IN(3)
int   qx2Scal;      /* Qx/2 scaler */                  # TSQX2          $$
int   etotScal;     /* Etot scaler */                  # TSETOT         $$
int   nL1PiPi;      /* no. L1PiPi in L3 list */        # BST_L3IN(IDS)  $$
int   tsFL1PiPi;    /* first charged time-stamp: L1PiPi */ # TFIRST(IDS)$$
int   tsLL1PiPi;    /* last  charged time-stamp: L1PiPi */ # TLAST(IDS) $$
int   mbxScal;      /* Massbox trigger scaler */       # CHARSCAL       $$
int   nMbx;         /* no. Massbox triggers in L3 list */ # BST_L3IN(1) $$
int   nFiltCh;      /* no. of L3 filtered events */    # BST_L3CHRG     $$
int   nGoodCh;      /* no. of "good" charged events */                  $$
                                          # BST_KSPIMPIP+BST_KLPIMPIP
int   tsFMbx;       /* first charged time-stamp: MBX */ # TFIRST(1)     $$
int   tsLMbx;       /* last  charged time-stamp: MBX */ # TLAST(1)      $$


/* Neutral chain: --------------------------------------*/

int   nNhod;        /* no. NHOD in L3 list */          # BST_L3IN(6)   $$
int   tsFNhod;      /* first neutral time-stamp: NHOD */ # TFIRST(6)   $$
int   tsLNhod;      /* last  neutral time-stamp: NHOD */ # TLAST(6)    $$
int   pi0pi0Scal;   /* Pi0Pi0 trigger scaler */        # NEUTSCAL      $$
int   nPi0Pi0;      /* no. Pi0Pi0 triggers in L3 list */ # BST_L3IN(5) $$
int   nFiltNe;      /* no. of L3 filtered events */    # BST_L3NEUT    $$
int   nGoodNe;      /* no. of "good" neutral events */                 $$
                                          # BST_KSPI0PI0+BST_KLPI0PI0
int   tsFPi0Pi0;    /* first neutral time-stamp: Pi0Pi0 */ # TFIRST(5) $$
int   tsLPi0Pi0;    /* last  neutral time-stamp: Pi0Pi0 */ # TLAST(5)  $$


/* Dedicated variables: --------------------------------*/

int   dchErr[2];    /* DCH readout errors */           # originally
                     # ERR2, ERR9, since May 2000 only dummy values
                     # -1, -2; may be recycled
float mbxDead;      /* % Massbox deadtime */           # DEAD    $$
float mbxIll;       /* % Massbox illegal deadtime */   # ILDEAD  $$
float multTs;       /* % multiple time-stamp events */ # MULT
float nhitsTrack;   /* average no. hits/track */       # AVHITS
float mbxNoTrack;   /* % Massbox events with no track */ # NTRAK $$
```

```
Xoff  xoff;         /* XOFF structure */              # (see above)


/* Marco's variables: ----------------------------------*/

float bCorr[4];     /* beam correlation variables */            $$
                       # Corbmksm(1),Corqxaks(1),Mcorbmksm,Mcorqxaks
float mult[9];      /* detector multiplicities */               $$
             # Xmtag,Xmaks,Xmakl,Xmlkr,Xmnho,Xmdch,Xmhod,Xmhac,Xmmuv
float stat[4];      /* Wilcoxon and Kolmogorov-Smirnov */       $$
                    /* burst statistics test variables */
                       # Wz,TWz,KUprob,TKUprob

/* PMB variable: ---------------------------------------*/

int   pmbErr;       /* PMB error bit-packed word: bits 00-05 */  $$
                    /* are aks, akl, hod, ksm, nho and pu    */ # pmberr

/* Lkr file summary variables: -------------------------*/

int   summStat;     /* summary status */       # (not in Fortran code) $$
int   lkrDt1;       /* (logfile burst time - FIC time) */             $$
                                               # (not in Fortran code)
int   lkrDt2;       /* (logfile burst time - matched burst time) */    $$
                                               # (not in Fortran code)
int   nevLkr;       /* number of events seen by Lkr */                $$
                                               # (not in Fortran code)


/* errcompact variables: ------------------------------*/            $$

int   nmixEvt;      /* number of mixed events */              # ERR1
int   nIBwcBad;     /* number of IB/wordcount mismatches */  # ERR2
int   nevtNumBad;
       /* number of local/global event number mismatches */  # ERR3
int   nEmptyPdch;   /* number of empty list of hits from PDCH */
                                                              # ERR4
int   nDecPLKRbad;  /* number of decoding problems in PLKR bank */
                                                              # ERR5
int   nAccPLKRbad;
          /* number of problems to access PLKR in CALREAD */ # ERR6
int   nTooManyCells;/* number of times too many cells in CALREAD */
                                                              # ERR7
int   nDCmissThr;
       /* number of times DC missed being above threshold */ # ERR8
```

```
  int   nDCfalseThr;
         /* number of times DC falsely above threshold */      # ERR9
  int   nctssNo;      /* ctss: no : different M - que? */      # ERR10
  int   nctssYes;     /* ctss: yes: different M - que? */      # ERR11
  int   nerrCnt;
         /* no. of events counted by error counting system */  # IEVT

  /* DCH inefficiency variable: -------------------------*/

  int   dchIneff;
     /* bit-packed (00-31) DCH plane > ineff threshold */ # dchineff $$

  /* new DCH eff. variable from Eddy 02-6-99 ------------*/

  int dchEff;    /* drift chamber efficiency variable */ # DCHEFF    $$

  /* 2 spare ints ---------------------------------------*/
  /* int spare[2];   changed on 1999 Aug 6: */
  /* int spare; */

  int dchEff2;                                           # DCHEFF2   $$
  float bst_tageff; /* tagger efficiency per burst */    # BST_TAGEFF

  /* 2 new Raphael variables 19-02-99 -------------------*/

  float ovfNeut;      /* % of overflows in neutral events */ # NPIOOF
  float fracKs;       /* % of Ks in good vertex events */    # KSRAT

} burMap;
```

K Compact up-dates for 2007 analysis

Please note that the changes described in the next section on the compact structure are applied only on the production(s) coming from reprocessing and not in the production (goldcmp34) coming from L3 during 2007 data taking.

K.1 Compact structure

Only the contents of two variables are changed with respect to the previous (7.2) version. In the HODneuthit structure (charged hodoscope structure for all the hits, including the hits not associated to a track) now:

HODneuthit

Variable	Type	Description
evt− hodneut[hit].plane	int	Flag from PDS data
evt− hodneut[hit].counter	int	Scintillator number (1-128)

K.2 SuperCompact structure

For the 2007 run the KABstrack, HACclus and MaChit structures, given that the corresponding sub-detectors where not present during 2007 data taking, were used to store other quantities. Then please take in mind that for the 2007 run the data included on such supercompact variables are meaningless for the user.
In the following please take care of the variables type, some quantities usually stored on integer variables are now stored on floating variables.

A new event block has defined to include the following variables:

New superCmpEvent

Variable	**Type**	**Description**
SEVT_NHOD	int	Number of Charged Hod. hit data
SEVT_HODERRFLAG	int	Charged Hod. error flag
SEVT_HODSTATUS	int	Charged Hod. status bits
SEVT_NNHO	int	Number of Neutral Hod. hit data
SEVT_NHOERRFLAG	int	Neutral Hod. error flag
SEVT_NHOSTATUS	int	Neutral Hod. status bits
SEVT_NHODNEUT	int	Number of Charged Hod. hit data (all)
SEVT_HODNEUTFLAG	float	Set to 1 if Nhits was above limit

The following variables are added for Lkr's clusters:

cluster

Variable	Type	Description
SCLUSTER_TLATCELL(iclus)	float	Time of most energetic lateral cell (ns)
SCLUSTER_UCENERGY(iclus)	float	Uncorrected cluster energy (GeV)
SCLUSTER_SPACHACORR(iclus)	float	Correction applied for spacecharge effect
SCLUSTER_E77(iclus)	float	Cluster energy in 7×7 cells (GeV)
SCLUSTER_ECELLMAX(iclus)	float	Energy of the cell with highest energy (GeV)
SCLUSTER_E2SAMPALL(iclus)	float	Energy from the first two time slots (pedestal) of all cluster cells (GeV)
SCLUSTER_GAINMAX(iclus)	int	Gain of the cell with highest energy

Then already from 2003 in a spare variable of the superCompact cluster structure are coded two integer variables:

sevt->cluster[iclus].spare[1]=imax(iclus)+1000000*cellsread(iclus)

where imax is the index of the cell with highest energy in the cluster and cellsread is the number of cells read for each cluster.
A way to extract the two variables with C code, once cellsread[iclus] and imax[iclus] are declared integer, is the following:

```
cellsread[iclus]=sevt->cluster[iclus].spare[1]/1000000;
imax[iclus]=sevt->cluster[iclus].spare[1]-1000000*cellsread[iclus];
```

The following new superCompact structures are then added:

HODhit

Variable	Type	Description
SHOD_PHEIGHT(hit)	float	Pulse height (ADC counts)
SHOD_COUNTER(hit)	int	Scintillator number (1-128)
SHOD_PDSFLAG(hit)	int	Flag from PDS data
SHOD_TIME(hit)	float	Time of hit (ns)
SHOD_X(hit)	float	x coord. of DCH track (cm)
SHOD_Y(hit)	float	y coord. of DCH track (cm)

NHOhit

Variable	Type	Description
SNHO_PHEIGHT(hit)	float	Pulse height (ADC counts)
SNHO_COUNTER(hit)	float	Counter number (1-32)
SNHO_PDSFLAG(hit)	int	Flag from PDS data
SNHO_TIME(hit)	float	Time of hit (ns)

HODneuthit

Variable	Type	Description
SHODNEUT_PHEIGHT(hit)	float	Pulse height (ADC counts)
SHODNEUT_COUNTER(hit)	float	Scintillator number (1-128)
SHODNEUT_PDSFLAG(hit)	float	Flag from PDS data
SHODNEUT_TIME(hit)	float	Time of hit (ns)