The Calorimeter LEVEL2 package v. 3.07 README

Emiliano Mocchiutti

21st July 2005

This is the ROOT function written to calibrate and process the calorimeter data creating a final level output that can be used in the data analysis. The output format (ROOT or PAW) can be chosen at running time tuning an input variable. The program connects to the Trieste MySQL server which stores the calibration tables for the calorimeter. Before using this program user must process data with the tracker ground software version 1.00. The program is able to handle both tracker ntuples and rootples, checking first for rootples and then for ntuples.

WARNING1: The alignment between calorimeter and tracker is still temporary even if a good approximation is used.

WARNING2: Calorimeter self-trigger events are not processed with this software version.

==========

INDEX

==========

1) INSTALLATION

==========

The software has been rearranged to satisfy the PAMELA repository requirements.

This CaloLEVEL2.c program requires:

- the correct set up of the PAMELA environment;

- calorimeter COMMON package;

- root2paw package;

- tracker ground software v. 1.00;

- a working ROOT version;

- a working YODA version;

- the MySQL client package installed;

To compile the libraries you will also need:

- YODA source;

- GCC with Fortran support;

- CERN libraries.

- MySQL development package installed (check you have also installed the mysql_config program).

This software has been successfully compiled and tested with the following programs:

- ROOT 4.03/02 9 February 2005

- YODA 4_400

- CERN 2002

- CERN 2003

- CERN 2004

- gcc (GCC) 3.2.3 20030502 (Red Hat Linux 3.2.3-49)

- gcc (GCC) 3.4.1 (Mandrakelinux 10.1 3.4.1-4mdk)

- mysql Ver 11.18 Distrib 3.23.58, for redhat-linux-gnu (i686)

Installation steps:

a) download the CaloLEVEL2 software (plus the calorimeter common package software if not already installed). You can find the software on the afs repository:

/afs/ba.infn.it/user/pamela

or you can download it from the WEB, in this case go to:

http://pcba28.ba.infn.it/cgi-bin/cvsweb.cgi

from the menu “CVS Root” choose “PAMELA repository”, click on “calo/LEVEL2” and click on “Download this directory in tarball or zip archive” depending on your preferences.

b) once you have downloaded your package unpack it somewhere.

c) enter the “LEVEL2” directory, you will find these file and directories:

doc/ macros/ inc/ src/ bin/ lib/ Makefile calib/ data/

d) check you have installed the calorimeter common package and the root2paw package (the latest one is needed to retrieve the tracker level2 informations).

e) make sure you have set up your PAMELA environmental variables and directories (if you have installed the calocommon package you should already have done this). To do so, choose a path where you want to install the PAMELA software (let's say /mydirectory/pamela/) and create the following directories: bin , lib , src , inc , macros , docs , calib .
Then edit your login configuration file (people using bash shell will edit $HOME/.bashrc , people using tcsh will edit $HOME/.tcshrc and so on), and add the following environmental variables:

```
export PAM_BIN=/mydirectory/pamela/bin
export PAM_LIB=/mydirectory/pamela/lib
export PAM_SRC=/mydirectory/pamela/src
export PAM_INC=/mydirectory/pamela/inc
export PAM_MACROS=/mydirectory/pamela/macros
export PAM_DOC=/mydirectory/pamela/docs
```

```
export PAM_CALIB=/mydirectory/pamela/calib
export PAM_YODASRC=/mydirectory/pamela/yoda/yodaX_YYY/
export PAM_YODALIB=/mydirectory/pamela/yoda/lib/
```

The last two variables determine where is the source of YODA and the path to the installed YODA libraries.
I also suggest to add the PAMELA bin directory to your default bin path but this is not essential:

```
export PATH=$PATH:$PAM_BIN
```

People using tcsh will use “setenv” instead of “export” and will put a space at the place of the sign “equal”. This set up has to be done only once for all the packages.

f) cd to the LEVEL2 directory and give:

```
 make all install
```

the program will check the system and create the needed libraries for the CaloLEVEL2 program; then it will install the CaloLEVEL2 libraries and macros in the paths you have given.
If you are upgrading an older version, use “make all forceinstall” to force the installation of latest programs or “make all upgrade” to force the installation and to delete the old libraries.

==========

2) USER GUIDE

==========

The macro CaloLEVEL2.c has been compiled to improve the processing velocity. To load the compiled macro follow this step: start the ROOT session...

```
bash> root
```

...load the macro...

```
root[0] .x CaloLEVEL2.C
```

Notice that this step is different respect to the past. You must use “.x” instead of “.L” and the program name now is “CaloLEVEL2.C” whit “.C” (upper C) instead of “.c”. You can still run the program loading and interpreting the macro the same way it was done with the old software but this way the processing will be about 30 times slower.

To run the program once the compiled macro is loaded you can just give:

```
root [1] CaloLEVEL2("/mypath/to/filesfromyoda/dw_050518_00100/");
```

Calling CaloLEVEL2 this way we are assuming that:
1) we have already processed the tracker data with tracker software version 1.00 and we have put the tracker level2 output in the YODA structure or we have processed the tracker level2 ntuple with GroundDataConver (root2paw package) and we have produced a rootple stored in the YODA structure;
2) we want to store the output in the YODA structure;
3) the calorimeter level2 output does not exist;
4) we want as output a rootple.
For further informations read below.

The output will be like this:

```
 Filename will be:
/mypath/to/filesfromyoda/dw_050518_00100/dw_050518_00100.Physics.Level2.Calori
```

```
meter.Event.root

 Not in FORCE mode, check the existence of LEVEL2 data:

Error in <TFile::TFile>: file
/mypath/to/filesfromyoda/dw_050518_00100/dw_050518_00100.Physics.Level2.Calori
meter.Event.root  does not exists.

 OK, I will create it!

 Using calorimeter calibration file:
 /mydirectory/pamela/calib/CaloADC2MIP.root

 Try the connection to the MySQL database in Trieste...
 ...OK, the connection is fine!

 Using database "romemuons", table "calocalib_dw_050518_001"

 TRACKER: loading the magnetic field maps...


 Opening first map   : -/mydirectory/pamela/calib/bfield_n3.rz-


 Opening second map  : -/mydirectory/pamela/calib/bfield_n4.rz-


 ...done!

 Check the existence of tracker data...
 ...found tracker level2 ROOTPLE:
/mypath/to/filesfromyoda/dw_050518_00100//Physics/Level2/dw_050518_00900.Physi
cs.Level2.Tracker.Event.root

 Processed events:

 ** SECTION 0 **
 - event at time 154802. From time 114823 to time 356400
   use calibration at time 114823, file
            /mypath/to/filesfromyoda/dw_050518_00100/

 ** SECTION 1 **
 - event at time 154802. From time 114876 to time 356400
   use calibration at time 114876, file
           /mypath/to/filesfromyoda/dw_050518_00100/

 ** SECTION 2 **
 - event at time 154802. From time 114929 to time 356400
   use calibration at time 114929, file
           /mypath/to/filesfromyoda/dw_050518_00100/

 ** SECTION 3 **
 - event at time 154802. From time 114989 to time 356400
   use calibration at time 114989, file
           /mypath/to/filesfromyoda/dw_050518_00100/


 Finished, exiting...
```

This is the standard way to call CaloLEVEL2. Read carefully the output to check for any WARNING. With the standard call any ERROR will stop the program. By default the program will create a rootple called (as in the example) dw_050518_00100.Physics.Level2.Calorimeter.Event.root

in the directory
"/mypath/to/filesfromyoda/dw_050518_00100/Physics/Level2/".
By default the program will try to connect to the MySQL database server in Trieste (username and password can be found in the COMMON/src/readmy.c file, package COMMON). If any problem is found in connecting the program will stop running.

WARNING: the program will query the database only to know the calibration it has to use in processing files and will not retrieve any calibration data from the database. This means that you must provide the unpacked files with calibration data. The program will look by default for unpacked data in the same directory of the input file and with the same YODA level of the input file. For example if I want to process on gundam in Rome the file "/mypath/to/filesfromyoda/DW_050323_00103" and according to the Trieste database the calorimeter calibrations for that file can be found in the file "DW_050322_009" then I must have both these files unpacked in Rome with the same level. Hence the YODA directory for the calibration file must be "/mypath/to/filesfromyoda/DW_050322_00903/" (ending in "03"!). The Trieste database has been built with the files listed in the "List of PAMELA acquisition taken in night shifts in Tor Vergata Clean room" database, so if you have unpacked all that files you should have no problems. The program will give an error if it will not be able to find the correct calibration file.

To check the connection to MySQL database and the calibration files needed you can use the script "CaloMySQLFINDCALIBS" that can be found in the macro CaloFINDCALIBS.c (version greater or equal than 2.01).

The number of queries to the database depends on the number of calibrations needed to process the file and not on the number of events of the file. There will be four queries for any calibration needed, independently on the number of events.

The possible input options are the following:

short int CaloLEVEL2(TString filename, Tstring TrackerDir="", TString outDir ="", TString Framework = "root", Int_t FORCE = 0)

- filename : is the path to the YODA unpacked data directory;

- TrackerDir: the path to the directory which contains tracker level2 data (ntuples or rootples). NOTICE: the program will search for ntuples which name is of the form "DW_YYMMDD_NNN_level2.rz" and for rootples which name is of the form "DW_YYMMDD_NNNnn.Physics.Level2.Tracker.Event.root" If TrackerDir is left empty (default) the program will search for ntuples or rootples in the YODA structure, that is in the directory: /mypath/to/filesfromyoda/DW_ YYMMDD_NNNnn/Physics/Level2/

- outDir : is the output directory, the directory where you want to store the calorimeter level2 data. If not given the program will create DW_000000_00000/Physics/Level2 as output directory. The filename will always be of the form

  DW_000000_00000.Physics.Level2.Calorimeter.Event.{root/rz}

- Framework : this flag allows the user to choose as output a rootple ("root", default value) or a ntuple ("paw");

- FORCE : when set to 0 the program will check if the output file exists. If so it will give a warning and it will exit without overriding the existing file. Set this flag to 1 to force the processing of data; in this case the old file will be lost. In FORCE mode the program will try to recover any error and to go on processing the file. Check carefully the output for any error!

3) OUTPUT FORMAT

The generated rootple has the following format:

```
*******************************************************************************
*Tree    :CaloLevel2: PAMELA Level2 calorimeter data                          *
*Entries :      140 : Total =          156014 bytes  File  Size =      26722 *
*        :          : Tree compression factor =   1.00                       *
*******************************************************************************
*Br    0 :OBT       : OBT/I                                                  *
*Entries :      140 : Total  Size=       1186 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    1 :pkt_num   : pkt_num/I                                              *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    2 :pro_num   : pro_num/I                                              *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    3 :trigty    : trigty/F                                               *
*Entries :      140 : Total  Size=       1204 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    4 :good      : good/I                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    5 :perr      : perr[4]/I                                              *
*Entries :      140 : Total  Size=       2878 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    6 :swerr     : swerr[4]/I                                             *
*Entries :      140 : Total  Size=       2884 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    7 :crc       : crc[4]/I                                               *
*Entries :      140 : Total  Size=       2872 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    8 :nstrip    : nstrip/F                                               *
*Entries :      140 : Total  Size=       1204 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br    9 :qtot      : qtot/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
*Br   10 :ncore     : ncore/F                                                *
*Entries :      140 : Total  Size=       1198 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.............................................................................*
```

```
*Br   11 :qcore     : qcore/F                                                *
*Entries :      140 : Total  Size=       1198 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   12 :impx      : impx/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   13 :impy      : impy/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   14 :tanx      : tanx/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   15 :tany      : tany/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   16 :nint      : nint/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   17 :ncyl      : ncyl/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   18 :qcyl      : qcyl/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   19 :qtrack    : qtrack/F                                               *
*Entries :      140 : Total  Size=       1204 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   20 :qmax      : qmax/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   21 :nx22      : nx22/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   22 :qx22      : qx22/F                                                 *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   23 :qq        : qq[4]/F                                                *
*Entries :      140 : Total  Size=       2866 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   24 :qtrackx   : qtrackx/F                                              *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   25 :qtracky   : qtracky/F                                              *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   26 :dxtrack   : dxtrack/F                                              *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
```

```
*.....................................................................*
*Br   27 :dytrack   : dytrack/F                                        *
*Entries :      140 : Total  Size=       1210 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   28 :qlast     : qlast/F                                          *
*Entries :      140 : Total  Size=       1198 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   29 :nlast     : nlast/F                                          *
*Entries :      140 : Total  Size=       1198 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   30 :qpre      : qpre/F                                           *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   31 :npre      : npre/F                                           *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   32 :qpresh    : qpresh/F                                         *
*Entries :      140 : Total  Size=       1204 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   33 :npresh    : npresh/F                                         *
*Entries :      140 : Total  Size=       1204 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   34 :qlow      : qlow/F                                           *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   35 :nlow      : nlow/F                                           *
*Entries :      140 : Total  Size=       1192 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   36 :qtr       : qtr/F                                            *
*Entries :      140 : Total  Size=       1186 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   37 :ntr       : ntr/F                                            *
*Entries :      140 : Total  Size=       1186 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   38 :cibar     : cibar[22][2]/I                                   *
*Entries :      140 : Total  Size=      25292 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   39 :tibar     : tibar[22][2]/I                                   *
*Entries :      140 : Total  Size=      25292 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   40 :cbar      : cbar[22][2]/F                                    *
*Entries :      140 : Total  Size=      25286 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   41 :tbar      : tbar[22][2]/F                                    *
*Entries :      140 : Total  Size=      25286 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   42 :planetot  : planetot/F                                       *
*Entries :      140 : Total  Size=       1216 bytes  One basket in memory    *
```

```
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br   43 :qmean     : qmean/F                                           *
*Entries :      140 : Total  Size=       1198 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
```

the class is CalorimeterLevel2 (defined in CaloFunctions.h)
the tree name is "CaloLevel2"
branch name is "Event"

The rootple contains also another branch with one entry only:

```
*********************************************************************************
*Tree    :Software  : Software used to generate data                             *
*Entries :        1 : Total =            1663 bytes  File  Size =        534 *
*        :          : Tree compression factor =   1.00                          *
*********************************************************************************
*Br    0 :swcode    : swcode/I                                                   *
*Entries :        1 : Total  Size=        644 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
*Br    1 :swtrkcode : swtrkcode/I                                               *
*Entries :        1 : Total  Size=        662 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*.....................................................................*
```

the tree name is “Software”

swcode is the software version of the calorimeter LEVEL2 program which produced the rootple. It is an integer coded this way: swcode = version + 100*subversion + 10000*subsubversion + ...
For example, software version 2.09 has swcode = 902 .

swtrkcode is the same as above but for tracker code used to produce track and rigidity informations used by the calorimeter LEVEL2 program.

The ntuple format is:

```
 ******************************************************************
 * Ntuple ID = 1      Entries = 140       Pamela Calo
 ******************************************************************
 * Var numb * Type * Packing *    Range     *  Block   *  Name    *
 ******************************************************************
 *      1   * I*4  *         *              * CALO     * OBT
 *      2   * I*4  *         *              * CALO     * pkt_num
 *      3   * I*4  *         *              * CALO     * pro_num
 *      4   * R*4  *         *              * CALO     * trigty
 *      5   * I*4  *         *              * CALO     * good
 *      6   * I*4  *         *              * CALO     * perr(4)
 *      7   * I*4  *         *              * CALO     * swerr(4)
 *      8   * I*4  *         *              * CALO     * crc(4)
 *      9   * R*4  *         *              * CALO     * nstrip
 *     10   * R*4  *         *              * CALO     * qtot
 *     11   * R*4  *         *              * CALO     * ncore
 *     12   * R*4  *         *              * CALO     * qcore
 *     13   * R*4  *         *              * CALO     * impx
 *     14   * R*4  *         *              * CALO     * impy
 *     15   * R*4  *         *              * CALO     * tanx
 *     16   * R*4  *         *              * CALO     * tany
 *     17   * R*4  *         *              * CALO     * nint
 *     18   * R*4  *         *              * CALO     * ncyl
```

```
 *     19   * R*4  *         *              * CALO     * qcyl
 *     20   * R*4  *         *              * CALO     * qtrack
 *     21   * R*4  *         *              * CALO     * qmax
 *     22   * R*4  *         *              * CALO     * nx22
 *     23   * R*4  *         *              * CALO     * qx22
 *     24   * R*4  *         *              * CALO     * qq(4)
 *     25   * R*4  *         *              * CALO     * qtrackx
 *     26   * R*4  *         *              * CALO     * qtracky
 *     27   * R*4  *         *              * CALO     * dxtrack
 *     28   * R*4  *         *              * CALO     * dytrack
 *     29   * R*4  *         *              * CALO     * qlast
 *     30   * R*4  *         *              * CALO     * nlast
 *     31   * R*4  *         *              * CALO     * qpre
 *     32   * R*4  *         *              * CALO     * npre
 *     33   * R*4  *         *              * CALO     * qpresh
 *     34   * R*4  *         *              * CALO     * npresh
 *     35   * R*4  *         *              * CALO     * qlow
 *     36   * R*4  *         *              * CALO     * nlow
 *     37   * R*4  *         *              * CALO     * qtr
 *     38   * R*4  *         *              * CALO     * ntr
 *     39   * I*4  *         *              * CALO     * cibar(2,22)
 *     40   * I*4  *         *              * CALO     * tibar(2,22)
 *     41   * R*4  *         *              * CALO     * cbar(2,22)
 *     42   * R*4  *         *              * CALO     * tbar(2,22)
 *     43   * R*4  *         *              * CALO     * planetot
 *     44   * R*4  *         *              * CALO     * qmean
 ****************************************************************
 *  Block   *  Entries  * Unpacked * Packed *   Packing Factor    *
 ****************************************************************
 * CALO     *  140      * 912      * 912    *       1.000         *
 * Total    *    ---    * 912      * 912    *       1.000         *
 ****************************************************************
 * Blocks = 1            Variables = 44          Columns = 228   *
 ****************************************************************
```

- ntuple ID is 1

- lrec is 8190

the ntuple file contains another ntuple ID:

```
 ****************************************************************
 * Ntuple ID = 2      Entries = 1         Software version
 ****************************************************************
 * Var numb * Type * Packing *    Range     *  Block   *  Name    *
 ****************************************************************
 *      1   * I*4  *         *              * VERSIONS * swcode
 *      2   * I*4  *         *              * VERSIONS * swtrkcode
 ****************************************************************
 *  Block   *  Entries  * Unpacked * Packed *   Packing Factor    *
 ****************************************************************
 * VERSIONS *  1        * 8        * 8      *       1.000         *
 * Total    *    ---    * 8        * 8      *       1.000         *
 ****************************************************************
 * Blocks = 1            Variables = 2           Columns = 2     *
 ****************************************************************
```

this ntuple contains software version informations (see above).

4) BRIEF DESCRIPTION OF VARIABLES

This description is partly taken from Jens Lund's PhD thesis (Appendix A2, page 126). All energies are measured in MIP.

* OBT - On Board Time of the event

* pkt_num - PSCU counter

* pro_num - YODA event counter for the processed file

* trigty - type of trigger which generated the event (taken from the trigger board information):
      trigty = 0 is a TOF trigger
      trigty = 1 is a S4/pulser trigger
      trigty = 2 is a calorimeter trigger

* good - integer, true if there were no error in the event (CRC or processing errors or alarms)

* perr[4] - integer, one for each section of the calorimeter, if true there was a processing error for the relative section in the selected event.

* swerr[4] - integer, one for each section, if true there was an error detected by the DSP(s) of the calorimeter for the selected event.

* crc[4] - integer, one for each section, if true there was a CRC error in data transmitted by the relative section in the selected event.

* nstrip - the total number of strip hit.

* qtot - the total measured energy in the calorimeter.

* n(q)core - SUM(j=1,2)SUM(i=1,PLmax) Nhit(i,j)*i , where Nhit(i,j) is the number of hits in a cylinder of radius 2 Rm (Moliere radius) around the track in the i-th plane (where the top plane is number 1 and the sum runs up to plane number PLmax, closest to the calculated electromagnetic shower maximum of the j-th view). "qcore" is similar but uses the measured energy instead of the number of strip hit.

* impx(y) - the x (y) impact position on the first plane as determined by the track fitted in the calorimeter.

* tanx(y) - the tangent of the angle in the x (y) direction as determined by the track fitted in the calorimeter.

* nint - SUM(j=1,2)SUM(i=1,22) TH(i,j)*i , where TH(i,j) = 1 if the i-th plane of the j-th view has a cluster along (less than 4 mm away) the track with a deposited energy typical of a proton (order of one MIP), otherwise TH(i,j) = 0.

* n(q)cyl - the measured energy deposited (number of strip hit) in a cylinder of radius 8 strips around the shower axis.

* qtrack - the energy deposited in the strip closest to the track and the neighbouring strip on each side.

* qmax - the maximum energy detected in a strip.

* n(q)x22 - the number of strip hit (energy) in the last silicon plane of the calorimeter (x view number 22).

* qq(4) - the energy released in the first half of each of the four calorimeter sections.

* qtrackx(y) - measured energy in clusters along the track in the x(y)-view.

* dx(y)track - measured energy outside the clusters along the track in the x (y)-view.

* q(n)last - the same as "q(n)cyl" but only for the last four planes.

* q(n)pre - the same as "q(n)cyl" but only for the first three planes.

* q(n)presh - the same as "q(n)cyl" but with radius 2 strips and only in the first four planes.

* q(n)low - the same as "qtot(nstrip)" but below the calculated electromagnetic shower maximum.

* q(n)tr - the same as "q(n)cyl" but with radius 4 strips.

* cibar(2,22) - for each view and each plane the strip traversed by the particle (or by the shower axis) according to the fit of the track performed by the calorimeter.

* tibar(2,22) - the same but using the tracking information.

* cbar(2,22) - for each view and each plane the position in millimeters traversed by the particle (or by the shower axis) according to the fit of the track performed by the calorimeter.

* tbar(2,22) - the same but using the tracking information.

* planetot - number of planes used to calculate the energy truncated mean “qmean”.

* qmean - the energy truncated mean that is the average energy deposit for the five planes with the smaller energy deposit of the whole calorimeter.

5) KNOWN BUGS

No known bugs. Write to Emiliano (Emiliano.Mocchiutti@ts.infn.it) to report any problem.