

The PAMELA Event Viewer package v. 8.00 README
Emiliano Mocchiutti
5th December 2005

In this README file there is a description of how to install and use the PAMELA event viewer. Some informations about the visualization of the detectors are also reported in the "GENERAL INFORMATIONS" section.

INDEX:

- 1) INSTALLATION
- 2) STANDARD USE
- 3) FEATURES
- 4) GENERAL INFORMATIONS
- 5) RUNNING OPTIONS
- 6) KNOWN BUGS
- 7) CHANGELOG

1) INSTALLATION:

Please refer to the calorimeter COMMON package (required by this package!) for the installation.

2) STANDARD USE:

The EventViewer has been compiled. The standard use is displayed giving the

```
EventViewer --help
```

command.

To use the interpreted event viewer version cd to the macros directory and start ROOT.

If you have not changed the \$HOME/.rootrc file (or you don't know what I am talking about) the PAMELA environment will be automatically loaded and you must see this printout on the terminal:

```
Welcome to the PAMELA environment!
```

If you don't see this printout or if you start your ROOT session from another directory than the \$PAM_MACROS one you must first execute the rootlogon.C file by hand:

```
bash> root  
root[] .x /mydirectory/pamela/macros/rootlogon.C
```

At this point you should be welcomed to the PAMELA environment. Once you are inside the PAMELA environment you just have to load the EventViewer.c file (mind the change in the name!) and start the macro. The only input needed is the path to the directory "filename" created by YODA for the data file you want to analyze. Hence the standard use is:

```
root[] .L EventViewer.c
root[] ShowEvent("/home/data/filesFromYoda/DW_050112_00600/");
```

3) FEATURES:

The event viewer accepts as input variables the following data:

```
ShowEvent(TString filename, TString selfile = "", Int_t ctrlword = 509,
Int_t FORCELEV = -1, Int_t fromevent = 0, Int_t toevent = 0,
TString outDir = "")
```

Input variables:

- * filename = path to the YODA directory for a file.
- * selfile = selection file. Can be used to select events.
Default is "" (empty) that is no selection is applied. If a selection filename is given the selection will applied.
The selection file must have the same format as the template selection file called "filter.c". Notice that different levels can be loaded and selection must refer to the correct level.
- * ctrlword is an integer which can be used as a mask to customize the figure:

ctrlword bit meaning is the following:

ctrlword bit:	10	9	8	7	6	5	4	3	2	1
detector :	special	TOF	AC	TRK	CALO	S4	ND	infos	names	legenda

Examples:

```
ctrlword = 509 = 0111111101 : all detectors+infos are printed(DEFAULT)
ctrlword = 511 = 0111111111 : show all
ctrlword = 505 = 0111111001 : don't print infos on figure
ctrlword = 507 = 0111111011 : show detector names+legenda (no infos)
ctrlword = 8 = 0000001000 : show only ND
ctrlword = 1019 = 1111111011 : show all but infos without signals
```

When a bit is set to 1 the detector is shown in figure, when it is set to 0 the detector is NOT shown.

When the "special" bit (10) is set NO SIGNAL will be shown in the detectors. It can be used to draw a draft of the PAMELA apparatus without any event.

If set the "names" bit will print additional information on figure (detectors names).

- * FORCELEVEL = to force a certain level visualization.
For the moment only level0 can be forced. This option can be used if you want to see differences between level0 and higher level data.
All detectors will be forced to LEVEL0. The default value is -1, the program will try to show the highest level data for each detector.
- * fromevent = the first event to be shown. The number entered

here (and in "toevent") is the progressive number inside the YODA file you are looking at. To see a certain "event number" (number in the header of the physic packe) read below in the "Running options" section. fromevent = 0 means "show all" (default value).

- * toevent = the last event to be shown.
NB: if fromevent differs from zero and toevent is equal zero only the progressive event number "fromevent" will be shown.

- * outDir = directory where to place figures (if you want to print them).
Default is empty, any figure will be put in the "filename" directory.

5) RUNNING OPTIONS:

If the program is started to show more than one event the following dialog lines will appear after each event:

Press <enter> to continue, b<enter> to go backward, j<enter> to jump, p<enter> to save the figure, o<enter> for more options, q<enter> to quit:

- pressing enter the program will display the next event;
- pressing b+enter the program will show the previous event;
- j+enter will allow you to input an event or progressive number of a certain event. After giving j+enter it will ask you

Do you want to jump to a progressive number [p] or to an event number [e]?

press p+enter if you want to give the relative progressive number of event, that is the number of event with respect to the first of the file you are looking.

Press e+enter if you want to give an event number, the number of event written in the header of physics packet.

Then the program will ask to input the number and it will display that event.

- p+enter allows you to print the figure to a file. As input is required the extension of the type of figure you want to save (eps, ps, gif,...). Any ROOT recognized format is allowed. The name of the figure will be formatted as follow:

```
ev_1000_dw_050323_011.eps
  ^      ^_ filename
  |_ progressive number
```

if not outDir input has been given the figure will be saved in the "filename" directory.

In the case of B/W visualization the name will be of the form

```
ev_1000_dw_050323_011bw.eps
  ^         ^         ^_ black and white
  |         |_ filename
  |_ progressive number
```

- o+enter open the following sub-menu:

Press <enter> to go back to previous menu, c<enter> to toggle B/W visualization, t<enter> to toggle track visualization in S4 and ND:

= enter you will send you back to the previous menu without any action;

= c+enter will toggle between black and white and colour visualization; the event shown is redisplayed (note you will loose the delta time information). You can toggle between the two visualization types as many times you want;

= t+enter will toggle the visualization of the track (if any) in S4 and ND, since they have a different scale the track seem "broken" and you may want not to display it. The event shown will be redisplayed. You can toggle this configuration as many times you want.

- q+enter (or .q+enter) quit the event viewer (not ROOT).

4) GENERAL INFORMATIONS:

The program will try to use the highest level data existing for each detector.

In case they are missing it will try to display events using LEVEL0 data.

- CALORIMETER:

LEVEL0: when no calibration is available the program will try to show LEVEL0 calorimeter data. This will work only if the calorimeter acquired data in FULL or COMPRESS mode and in that case all the transmitted strip will be shown with light blue colour (in this case obviously it doesn't mean 0-2 MIP).

TRY TO GENERATE LEVEL1: the program will look for a calibration and will use a simplified algorithm to calibrate data event by event basis. The standard calibration procedure is in the script CaloLEVEL1.c and differs from the procedure used here in the search of calibrations and in the data correction (in the case of cross-talk or similar effects).

LEVEL1: the program will use the level1 rootple generated by the CaloLEVEL1.c script.

- TRACKER:

LEVEL0: the LEVEL0 information shown is the position of the cluster with maximum energy for each silicon detector. No pulse height signal is reported.

LEVEL1: when level1 information are available (only rootples generated by the root2paw package) all strip hit are shown with height proportional to the signal.

LEVEL2: when level2 data are available (only rootples generated by the root2paw package) the fitted track will be shown in red colour. If more than one track has been fitted they will be shown in different colours.

The Y-view is read with an ambiguity hence each cluster is reported twice in two different colours, light blue and green. In the plane view (in the center) it is shown the tracker information as seen from above. Six crosses are plotted one for each plane. The bigger is the cross the closer to S2 is the tracker plane relative to that cross, the smaller is the cross the closer to S3 is the tracker plane of that cross. The cross represent the position of the x and y clusters with the maximum energy release. The division into silicon detectors of each tracker plane is shown with a small black dot in the plane.

- TIME OF FLIGHT:

LEVEL0: the level0 information comes from the trigger board and represent the map of the PMTs which gave the trigger. The colour of the hit does not represent an energy release. The scintillators are divided into two half representing the A (upper triangles, corner to the right) and B (lower triangles, corner to the right) phototubes.

TRY TO GENERATE LEVEL1: TDC and ADC informations from the tof board are used.

Paddle are shown as hit if at least one phototube has collected any signal.

In the not segmented view the position as reconstructed by the TDC infos is reported; if no ADC signal is present the colour will be yellow.

The width of the signal in this case will represent the error in the position determination. The program uses a default first order calibration of data.

LEVEL1: TDC and ADC informations from the tof board are used.

Paddle are shown as hit if at least one phototube has collected any signal.

In the not segmented view the position as reconstructed by the TDC infos is reported; if no ADC signal is present the colour will be yellow.

The width of the signal in this case will represent the error in the position determination.

- ANTICOINCIDENCE: AC detector hit are shown in two different colours depending if they were inside the trigger time window or outside. This will work only with LEVEL1 data, in the case of LEVEL0 data all hit will be shown as being within the time window (i.e. in red colour).

- S4: S4 data are shown in MIP using a preliminar calibration

- NEUTRON DETECTOR: LEVEL0 data shown are the number of neutrons recorded during a trigger. The background information is shown as text.

Detectors are in scale a part X and Y dimension of ND and S4 reduced to fit in the figure.

Notice that when going backward with event visualization the time between two events cannot be calculated.

6) KNOWN BUGS:

- Positioning of detectors (expecially CARDS) is not accurate yet.

7) CHANGELOG:

```
// 7.03 - 8.00 (2005-11-24): many changes to compile the macro. The code
// is still very redundant and heavy but some
// bugs were fixed and now it should be more
// stable.
// Do not require anymore the AcFunctions.h
// header, a clone has been written
// (aclevlclass.h).
//
// 7.02 - 7.03 (2005-10-14): load library containing CalorimeterLevel1
// class definition.
//
// 7.01 - 7.02 (2005-10-12): changed neutron detector color scale to a
// more realistic one.
//
// 7.00 - 7.01 (2005-10-10): simplyfied input (less parameters), added
// support for TOF level1 data.
//
// 6.05 - 7.00 (2005-09-21): changes to work with tracker software
// version 2.00.
//
// 6.04 - 6.05 (2005-08-03): changes to fix bugs related to 64 bit arch.
//
// 6.03 - 6.04 (2005-07-22): small bug in signal visualization in TOF
// S11 fixed.
// 6.02 - 6.03 (2005-07-19): use clone routines of yodaUtility.c that
// can be found in CaloFunctions.h and don't
// load anymore yodaUtility.c
//
// 6.01 - 6.02 (2005-07-12): small changes due to correction for the
// compilation of CaloFunctions.h.
//
// 6.00 - 6.01 (2005-07-07): AC, now it reads LEVEL1 output (hit are
// shown with different colour if they are not
// in time with trigger). Some AC related
// small bugs fixed.
// Added synchronization routines between
// "YODA" output and tracker output (I am
// assuming that everything comes out from
// YODA is synchronized).
//
// 5.21 - 6.00 (2005-07-06): Now it can read tracker rootples generated
// by the GroundDataConvert software. Some
// minor bugs related to tracker visualization
// fixed.
// Notice that tracker data are not
// synchronized yet to the "YODA" data.
//
// 5.20 - 5.21 (2005-06-29): Added some options in the interactive menu
// (B/W toggling, track in S4/ND toggling).
// Created a new include file with structures
// for the event viewer, changed the WhatToDo
// routine, now renamed in WhatToDoEV.
// Bug: cannot run program twice when showing
// track without quitting ROOT, fixed. Changed
// "CALO" to "Calorimeter" and "TRK" to "Tracker"
// in verbose infos mode. Bug: don't show trigger
```

```
//          configuration on the text output when no
//          infos are shown on the figure, fixed.
//          Exit the event viewer also with ".q".
//
// 5.19 - 5.20 (2005-06-28): Bug: S4 z dimension is wrong! corrected.
//          New filename and changes needed to upgrade
//          to the new software architecture.
//
// 5.18 - 5.19 (2005-06-16): Bug: S4/PULSER flag is not always correct
//          since sometimes the trigger is given by S4
//          but the flag is not set. Fixed (?) with a
//          workaround, to be checked that is the best
//          we can do.
//
// 5.17 - 5.18 (2005-06-14): Added black and white visualization (use
//          input variable "BW" as flag). Wrong
//          position of S4 in level0 visualization,
//          fixed.
//          Saved PS files do not contain the TOF
//          information from TDC position, fixed.
//
// 5.16 - 5.17 (2005-06-08): S4 level0 and level1 splitted. Level0
//          shows the patterntrig variable
//          (NB: NOT ALWAYS CORRECT!!), Level1 shows
//          calibrated info from S4.
//
// 5.15 - 5.16 (2005-06-07): Unload libraries when exiting and change
//          back to the starting directory. Check if
//          the file is older than 050515_007.
//
// 5.14 - 5.15 (2005-05-17): Crashes if ADC to MIP calorimeter
//          conversion value is zero, fixed.
//
// 5.13 - 5.14 (2005-05-13): Force level0 visualization also for the
//          calorimeter.
//
// 5.12 - 5.13 (2005-05-10): Calorimeter Z dimension is WRONG!!! fixed.
//
// 5.11 - 5.12 (2005-05-02): OBT is On Board Time and not Orbital Time!
//
// 5.10 - 5.11 (2005-04-19): Bug fixed in the new subroutine
//          ShowTOFL1 and changed to the old behaviour the calorimeter
//          calibration search (it could took long time
//          to find a good calibration with the new method especially in
//          the case of a directory with a single file).
//
// 5.09 - 5.10 (2005-04-18): Added ShowTOFL1 subroutine to show TOF
//          informations instead of trigger infos. Still it possible to
//          see trigger infos forcing level0 data.
//
// 5.08 - 5.09 (2005-04-11): Changes in the CaloShowL0 subroutine:
//          do not calibrate more than once if there is a jump in OBT.
//
// 5.07 - 5.08 (2005-04-07): Changes in the CaloShowL0 subroutine.
//          Changed tree name of level1 and adc2mip calorimeter data.
//
// 5.06 - 5.07 (2005-04-05): Bug in the rigidity and chi2 report in
//          the figure, fixed.
//
// 5.05 - 5.06 (2005-03-30): Small changes in the whattodo
//          behaviour.
//
// 5.04 - 5.05 (2005-03-23): Introduced the input variable FORCELEV
```

```
//      to force the data level to load. For now it is possible only
//      to force level0 data reading of calorimeter and tracker.
//      News: the selection file name now can be anyone and must be
//      used as input if you want to select events. Hence the
//      "selection" integer input has become the string "selfile".
//
// 5.03 - 5.04 (2005-03-18): Small changes in the print out.
//
// 5.02 - 5.03 (2005-03-17): Cannot go backward in selection mode,
//      fixed. Change in ShowS4 the S4 calibration parameter.
//      New: now it is possible to jump to a certain event number
//      and not only to progressive number.
//
// 5.01 - 5.02 (2005-03-16): Pulser information is now printed in
//      the INFOS and S4 MIP signal (first order calibration) is
//      represented with colours (same palette as for calorimeter).
//
// 5.00 - 5.01 (2005-03-14): Information text size increased.
//
// 4.11 - 5.00 (2005-03-11): Track visualization offset bug fixed.
//      Reload magnetic field maps if back to the first event,
//      fixed. Added some comments in the code. Added level2 tracker
//      data to filter.c.
//
// 4.10 - 4.11 (2005-03-10): Tracker LEVEL1 visualization
//      implemented. Changed the tracker directory from "magnet" to
//      "tracker". Show LEVEL1 information even if LEVEL2 data are
//      not present.
//
// 4.09 - 4.10 (2005-03-08): Read the calorimeter calibration file
//      if trying to generate level 1 data.
//
// 4.08 - 4.09 (2005-03-03): S4 always shown as hit, fixed.
//
// 4.07 - 4.08 (2005-02-25): In filter mode added a progress
//      viewer. CAS1 plane view was filled with full red colour
//      instead of pinkish colour, fixed.
//
// 4.06 - 4.07 (2005-02-25): crashes when exiting on empty files.
//      Fixed.
//
// 4.05 - 4.06 (2005-02-24): Changed order of input variables.
//
// 4.04 - 4.05 (2005-02-24): Changed variable definition from C/C++
//      style to ROOT style (int->Int_t).
//
// 4.03 - 4.04 (2005-02-23): Added the "selection" input variable.
//      If set greater than zero the program will look for events
//      satisfying the condition in the filter.c program. Notice
//      that running in this configuration will not let you to go
//      backward with events.
//
// 4.02 - 4.03 (2005-02-23): Sometimes the calorimeter was
//      repeating an event to infinity. Fixed.
//
// 4.01 - 4.02 (2005-02-22): ROOT 4.03/02 version bug fixed
//      (calorimeter only).
//
// 4.00 - 4.01 (2005-02-18): Calorimeter compress/full raw mode
//      visualization didn't work. Fixed. Neutron detector data
//      format changed, fixed accordingly.
//
// 3.04 - 4.00 (2005-02-14): Small changes in the calorimeter x/y
```



```
//      view and in the tracker plane view geometries. Added the
//      track visualization code once you have tracker LEVEL2 data.
//
// 3.03 - 3.04 (2005-02-10): Show not only the YODA number
//      (progressive number) but also the EVENT NUMBER from the
//      packet header!
//
// 3.02 - 3.03 (2005-02-08): TOF scintillators divided into A and B
//      PMTs. Upper triangles are A PMT, lower triangles are B PMTs.
//      S4 readout implemented: the information comes from the
//      trigger board and the energy level is drawn out as LEVEL0
//      information. Some bugs in the scintillators order fixed.
//      Changed color of shadow hit in the tracker Y-view; added
//      small black dot on tracker ladders to indicate the detectors
//      edges.
//
// 3.01 - 3.02 (2005-02-04): Tracker hit shown in the plane view.
//      CAS shown in the plane view. Level0 TOF information shown in
//      the scintillators. Bug for which it is not possible to see
//      only the last event, fixed.
//
// 3.00 - 3.01 (2005-01-31): AC CARD view from above added.
//      Printing bug fixed, now it calls for the WhatToDo routine in
//      CaloFunctions.h.
//
// 2.01 - 3.00 (2005-01-21): Major changes in the program
//      philosophy: now it will check first for higher order LEVEL
//      data and it will use low level data only as last resource.
//
// 2.00 - 2.01 (2005-01-18): AC cat changed, added PMTs, AC hits
//      out of triggers are shown now in a different colour. Added
//      "go backward" capability.
//
// 1.01 - 2.00 (2005-01-14): many changes, some cleanup of the
//      code, detectors are in scale.
//
// 1.00 - 1.01 (2004-12-15): Include also yodaUtility.c .
//
```