INSTALLING AND RUNNING
DARTHVADER (v2r01)

Emiliano Mocchiutti

(Emiliano.Mocchiutti@ts.infn.it)

2006/11/28

1) Installing DarthVader

DarthVader requires YODA, ROOT and YodaProfiler to be already installed in the system. The package yodaUtility is not needed anymore.

To install the DarthVader program follow these step:

– set up the PAMELA environment, as example you can use my setting (it is done for the BASH shell):

```
export PAM_DBHOST=mysql://srv-g2-01.ts.infn.it/pamelaflightnew
export PAM_DBUSER=root
export PAM_DBPSW=CaloTs
export PAM_YODA=/wizard3/pamela/sw/slc4
export PAM_BIN=/wizard3/pamela/sw/slc4/bin
export PAM_LIB=/wizard3/pamela/sw/slc4/lib
export PAM_SRC=/wizard3/pamela/sw/slc4/src
export PAM_INC=/wizard3/pamela/sw/slc4/inc
export PAM_MACROS=/wizard3/pamela/sw/slc4/macros
export PAM_DOC=/wizard3/pamela/sw/slc4/docs
export PAM_CALIB=/wizard3/pamela/sw/calib
export ARCH=`uname`
```

the PAM_DB* variables can be omitted, their use is explained below.

– download the DarthVader source code
– enter the directory DarthVader
– give the command

```
make distclean all upgrade
```

– set up the GL_PARAM table in the DB, to do so enter the directory "docs" and give the command:

```
./install_GL_PARAM.sh --user=root --host=mysql://localhost/pamelaprod --psw=CaloTs
```

where as input you must give username and password of a MySQL user who have write permissions on the DB. The host must be the computer running the MySQL server the name of the DB ("pamelaprod" in this example) can be any.

Wait 30 seconds and the GL_PARAM table will be filled.
If you have set up the PAM_DB* variables you can just give the command:

```
./install_GL_PARAM.sh
```

if you give the input variables the environmental variables will be overridden.
WARNINGS:
*1) if the GL_PARAM table already contains data it will be cleaned and repopulated, **all table data will be lost**! You will have 30 seconds to interrupt the procedure before cleaning the table.*
*2) **NO ERROR OR WARNINGS MUST be issued by this procedure**. If you experience any problem first check that you are using a MySQL version newer or equal to 4.1.20, than check that you have the InnoDB engine active (look in the file /etc/my.cnf , the "skip-innodb" must be commented or absent).*

The installation is completed.

2) Running DarthVader

DarthVader is the program which calibrates the unpacked data and generates the "level2" output which can be used for the analysis.
DarthVader needs the informations contained in the DB created by YodaProfiler and creates a .root file for each run listed in the GL_RUN table. The file contains a TTree for each detector (Calorimeter, Tracker, Trigger, Anticounters, S4, ND, ToF, OrbitalInfo). An additional TTree called "Run" is created which is the copy of the corresponding row in the GL_RUN table of the DB.
The program by default process all detectors data. If a detector has been excluded by the acquisition DarthVader will exit with error requiring to explicitly exclude that detector from the analysis.
For online help give the --help option which will print:

```
Usage:

DarthVader [ options ] -idRun ID_RUN [+-all] [+-detector [ detector options ] ]

Options are:

--version         print informations about compilation and exit
-h | --help       print this help and exit
-v | --verbose    be verbose [default]
-s | --silent     print nothing on STDOUT
-c | --clean      remove file if exiting with errors
-b | --benchmark  perform and print a benchmark test
-auto | -AUTO     exclude from processing detector which are NOT in the
                  acquisition [default]
-zerofill         if a detector is not in the acquisition the routine is
                  called anyway but all detector's data will be marked as bad
-tedious          exit with error if a detector is not in the acquisition and
                  it has not been excluded from processing
-host             name for the host [default = $PAM_DBHOST
                  or mysql://localhost/pamelaprod]
```

```
-user               username for the DB [default = $PAM_DBUSER or "anonymous"]
-psw                password for the DB [default = $PAM_DBPSW or ""]
-processFile file   output filename [default ID_RUN.Level2.root]
-idRun              ID_RUN: ID number of the run to be processed
+all | +ALL         call all detectors software [default]
-all | -ALL         call nothing
+detector           process detector; detector can be: TOF,TRK,CAL,TRG,ORB,S4,ND,AC,RUN
-detector           do not process detector (as above)
                    detector options must be included in square parenthesis with
                    spaces, for example:
                    +CAL [ --verbose -g ] +TRK [ -v --level1 ]

Examples:

Standard call: DarthVader -idRun 1085

Process only RunInfo and Tracker (be verbose for tracker):
DarthVader -idRun 1085 -all +RUN +TRK [ --verbose ]

Process all and be verbose for calorimeter:
DarthVader -idRun 1085 +CAL [ --verbose ]
```

The standard call will be like the one in the example. Notice that if you have set the PAM_DB* environmental variables they will be used for the DB connection. If they are missing and no input is given, the default will be host=mysql://localhost/pamelaprod, user=anonymous, psw="". The DB connection can be controlled using the input flags -host, -user and -psw. Notice that these flags will override the environmental variables if set.

NOTICE: DarthVader does not modify the DB, hence a user with readonly permissions on the DB can be used.

The DarthVader standard output looks like:

```
>DarthVader -idRun 2773

 Welcome to the PAMELA LEVEL2 flight software, version v2r01


 Processing run number 2773

 RunInfo called
 RunInfo: opening RunInfo tree
 TrackerLevel2 called
 ToFLevel2 called
 CalorimeterLevel2 called
 TriggerLevel2 called
 AnticounterLevel2 called
 S4Level2 called
 NDLevel2 called
 OrbitalInfo called


 Finished, exiting...
```