MAGIC-TDAS 01-04
010419/OBlanch

How To Use the camera simulation program

O Blanch
<blanch@ifae.es>

19 April, 2001

Abstract

Here, I gave a user guide to run the Montecarlo simulation of the MAGIC camera.

Contents

1 Introduction

Simulation of Camera: This is done with the program camera. This program was originally written by Jose Carlos. Some modifications and improvements were introduced by Oscar Blanch, Dirk Petry and Harald Kornmayer. This program is able to read in files created with reflector. In addition files from the StarField adder can also be read in. The camera program simulates the photomultiplier and the behaviour of the Trigger and FADC system. The output of the program will in the same format as the real data will be later. So it is in principle possible to test the whole analysis software.

2 Installation

First you should get the current version of the program. The easiest way to get it is from the web page:

http://hegra1.mppmu.mpg.de/MAGICWeb/private/montecarlo.

This page is internal and therefore you will need a password. There you have to choose Download. You will find the current version of the camera program for MAGIC and you can get a zip file with all the needed code. Form here on, we will explain the procedure to get the camera program running from this zip file. If you got the code from somewhere else you would have to make sure that you got all the needed code (see Appendix B) and then follow the instructions from point 3.

1. Decompress the file using:

   gunzip camera*.gz

2. Unpack the tar file with:

   tar xvf camera*.tar

3. Change the directory to the ranlib:

   cd MagicProgs/Simulation/Detector/lib

4. Link the linranlib.a.[yoursystem] to libranlib.b

   ln -s libranlib.a.[yoursystem] libranlib.a

5. Select the right quantum efficiency file

   cd ../Data
   ln -s qe-emi.RFL.dat qe.dat

6. Change to Camera directory:

   cd ../Camera

7. Check the dependencies with the operating system. The program is supposed to run under osf and Linux.

   make depend

   If an error occurs this may be due to the fact, that on your system the system variable "$OSTYPE" is not "osf1" or "linux". If you have an other value for "$OSTYPE", check the system variable and create a link to the config.mk.$OSTYPE you need (i.e. ln -s config.mk.linux config.mk.linux-gnu).

8. Compile the program using:

   make all

   Perhaps you have to change some values in the Makefile to compile the code.

9. Then the program is ready to run but you might need some input files that are described in the next section. Check that you have all the needed input files at that they are right and run the program with:

   camera < input.card

3 Needed files

The camera program needs several input files to do its job.

1. The input card (called "input.card" in the directory of the camera program) is used to control the job. In the next section (see chapter 4) this input.card is extensively described.

2. A reflector file contains the information of the simulated photons that hit the camera plane. So this is the input for the processing of camera. The actual reflector file that ones is going to use can be chosen with the input card.

3. The quantum efficiency for the photo-detectors are read from a file called "qe.dat". It is located in the directory MagicSoft/Simulation/Detector/Data. In that directory are two "qe.dat"-like files:

   - "qe-emi.RFL.dat" for the EMI photomultipliers, which are supposed to be used in the first phase of MAGIC.
   - "qe-ipd.RFL.dat" for the photo detectors of the future using intevac photo detectors.

   To selected one of the two files above just link them to "qe.dat" in the same directory. (See also item 5 in the installation procedure (chapter 2).) One can introduce a new type of photo detectors by creating a new file (with the right format of course!) and creating the right link. At the moment on should use the standard install procedure (chapter 2).

4. If one wanted to simulate the Night Sky Background (NSB) it would be needed to have the right database files. It can be generated with the "starrespo" program, which is downloaded along with the camera program (chapter 7)

4 Input card

The input card sets all parameters and option of the camera program. It consists in a chain of commands that are read sequentially. It means that if one defines something twice or puts commands that are not compatible the last option will survive.

4.1 Mandatory commands

No steercard can miss the following commands:

camera 0.4

It should be the first line of the file. The camera program uses it to check if it is really an input file.

input_file /hd02/Maggi/RefData/Gamma/gam_15_on.rfl

The command input_file tells the code where the reflector file can be found. The file has to be an reflector file, otherwise the camera will stop. You can specify the file name by the absolute path (as above) or by the relative path compared to the acutal directory.

starfield_file /usr/users/magicsol/RefData/CrabNebula.rfl

It is path for the file with the star light field information. This file should be a reflector file generated by the starfield adder program. Any other reflector would also work but it most likely would give non-sense results. If this command is not in the input card the program will assume that there is no star light.

nsb_directory /hd02/Maggi/StarLight/

To speed up th simulation the response to the NSB is taken from a data base. This command tells the camera program where is the data base.

data_file MagicMC01.dat

This command gives to the program the name of the file where it has to store a brief summary of trigger statistics. The file name can use relative (as is done here) and absolute path.

root_file MagicMC01.root

The root_file commands indicates the name of the output file. This file will have the final information of the MAGIC simulation and it will be written in the root format that has been defined. The file name can use relative (as is done here) and absolute path.

ct_file ../Data/magic.def

The camera program simulates the general behaviour of a Cerenkov Telescope. The ct_file command tells the program where is the file with the specific characteristics for the Telescope thats behaviour we want to simulate.

end_file

The end_file command marks the end of the input card. Any thing after it will not exist for the camera program.

4.2 Optional commands

Some other commands set some paramters for the camera program. These commands are optionally. If they were not in the input card, default parameters would be used. They are the following:

nsb_mean 0.15 0

> To control the diffuse Night Sky Background (NSB) on uses this command.
> The first number sets the amplitude of the diffuse NSB. This is given in number of photoelectrons per one inner pixel of the camera per ns.
> The second value defines the minimum number of photoelectrons from the shower to simulate the NSB for this shower.

ana_pixels 577

> It is just the number of pixels that the camera analyse. The numbering begins from the central pixel and going spiralling out.

seeds 69184 10406

> It gives the seeds for the random number generators.

trigger_prop 6.0 0.25 1.0 2.0

> The trigger class needs several input about the behaviour of the simulated trigger. They are given with the "trigger_prop" input parameter.
> The first number is the length of the gate that the discriminator of one pixel opens each time the signal of this pixel passes form below to above the threshold.
> The next number means the minimum time (in ns) that two pixels have to be on to be assumed in coincidence.
> The shape of the single photo-electron response is also needed and it is given with the third and the last number. Currently, the shape is assumed to be a gaussian with amplitude in mv (third number) and FWHM in ns (last number).

fadc_prop 4.0 5.0

> The FADC simulation needs to know the single photoelectron response of the actual FADC. At this moment a gaussian with a given amplitude in FADC counts (first number) and FWHM in ns (second number) is used.

There are also few that gives some input parameters and, simultaneously, switch on or off some options of the camera program:

trigger_loop 4 20 2 10 0 2

> It switches the loop mode on. The three couple of numbers are the limits for the loop over threshold, multiplicity, and topology respectively. This trigger mode is meant to be used to perform trigger studies and, therefore, the camera program does not put the Raw Evt data in the root file.
> Be carefull to big ranges in the different parameters may crash the job due to a memory problem.

trigger_single 4 4 2

It switches the loop trigger mode off and gives the threshold, multiplicity, and topology for the single trigger mode.

select_energy 0 10000000000

This command selects the range of energy. Only the primary showers generated with energy inside this range will be analysed.

And finally several commands that switch one or off option of the camera program:

nsb_on

This switches the Night Sky Background (NSB) simulation on, both the diffuse light of the NSB and the star light.

nsb_off

This switches the Night Sky Background (NSB) simulation off, both the diffuse light of the NSB and the star light.

write_all_images

If this input parameter is in the input card, the information will be saved in the root file for all the simulated showers. Otherwise it is only saved for the ones that trigger.

nowrite_McEvt

The Monte Carlo information about the shower generation will not be inside the root file, if this input parameter exists.

write_McTrig

It tells the camera program that it has to write some information about the trigger simulation (threshold, multiplicity, and topology...) in the root file.

write_McFadc

It tells the camera program that it has to write some information about the fadc simulation (pedestals, single photoelectron response, ...) in the root file.

trigger_scan

It activates the interactive showing of the trigger signal for each event.

fadc_scan

It activates the interactive showing of the FADC signal for each event.

skip

This command works interactive, therefore it is not foreseen for a batch job. It allows skipping some of the generated showers.

data_from_stdin

If this input line is in the input card, the program will expect the data to be entered online. Otherwise, the program will use the input files that are defined in the input card and are already explained in this section. This options is only recommended for debugging mode and not for Monte Carlo production.

5 Output files

The camera program produces two output files, which are given with the input card, by the commands root_file and data_file.

The "data_file" is an ASCII file with a brief summary of the trigger statistics. Here is an example:

163851 event(s), with a total of 13923137 C.photons

Fraction of triggers:

Thres 4, Multi 4, Topo0: 0.00671342% (11 out of 163851)

Thres 4, Multi 4, Topo1: 0.00671342% (11 out of 163851)

Thres 4, Multi 4, Topo2: 0.00427217% (7 out of 163851)

The "root_file" is a root file, which can have several trees depending on with which option the camera program is executed. There are mainly three sets of data inside this file, which are spread over two trees and several branches (see figure 1).

The first set is the raw data. This is the actual output of the simulation, the data that simulates the real data of the MAGIC telescope. It would have one branch, which is called RawEvt, in the EvtTree. This branch is going to have the same structure than for the real data (see documentation about Raw Root Format).

The Monte Carlo information about the generation of the shower in the atmosphere can be also found. It contains information like the energy of the primary particles, the direction of the shower, ... And it can be found in the branch called McData inside the EvtTree.

Finally, there is a set of data with information about the simulation of the trigger. This set is split in two branches: the HeaderTrig, which is in the HeaderTree, and the McTrig, which is in the EvtTree. The former contains common information for the whole run, like the configuration of the studied trigger. The later has information related to the trigger studies for each event. It saves the pixel that triggers, the time of trigger, the number of first level trigger, the number of second level triggers...

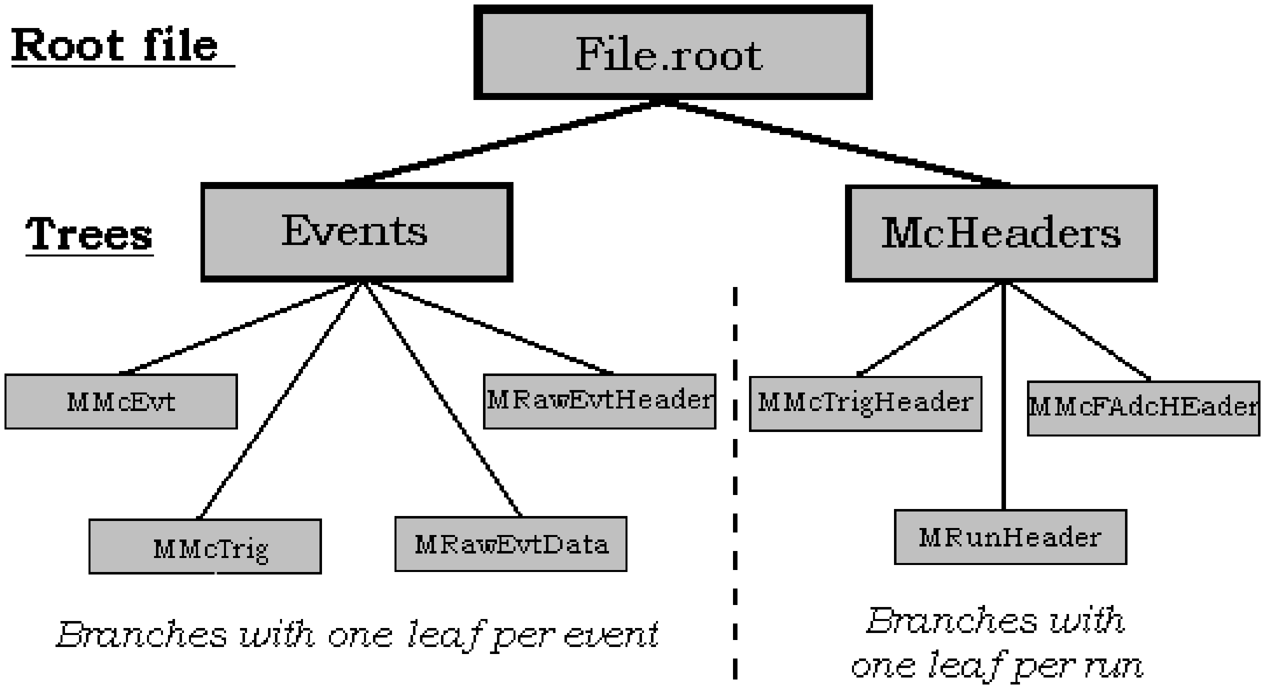

Figure 1: Diagram of root_file structure.

6 Most frequent problems

6.1 Compiling and linking

6.1.1 Library libranlib.a I

/usr/bin/ld: cannot open -lranlib: No such file or directory

collect2: ld returned 1 exit status

make: *** [camera] Error 1

Try to repeat the step 4 of the installation.

6.1.2 Library libranlib.a II

../lib/libranlib.a: could not read symbols: Archive has

no index; run ranlib to add one

collect2: ld returned 1 exit status

make: *** [camera] Error 1

The library libranlib.a most likely is not the one needed for your system. Check in the directory MagicProg/Simulation/Detector/lib if the library libranlib.a is linked to the one with the extension of your operation system.

6.2 Running

6.2.1 File qe.dat

ERROR in read_pixels: Cannot open "../Data/qe.dat". Exiting.

read_pixels: No such file or directory

Try to repeat step 5 of the installation.

6.2.2 New operator

Fatal in ¡operator new¿: storage exhausted aborting

It usually happens when the camera program is executed in loop mode. The new operator has a maximum space that can allocate. If one tries to analyses too many trigger configurations and wants to store the information of the Trigger simulation this space can be exceeded. Try to analyse less trigger conditions or to not write the trigger information in the root file (comment the "write_McTrig" command in the input card).

6.2.3 Unexpected mode

If the camera program does not work in the mode you expected, make sure that you did not leave some commands after the ones you put that cancel what you are asking to the program. The input card works sequentially and, therefore, the last command before the "end_file" has priority over the others.

6.2.4 Wrong data base for NSB

There can be several problems while trying to read the NSB data base. First of all one should check if the given path is right (becarefull, in should end with "/"). If it is, the problem is most likely due to a wrong data base. It can be either because the camera program cannot get the number of required photoelectrons from the existing files or because the trigger and FADC properties used to generate the data base are not the same than the ones that the camera program is checking. In the frist case, a larger data base should be generated. And in the second case, a new one with the right trigger and FADC properties has to be produced.

7 Appendix A: StarResponse program

The StarResponse program generates the data base that the simulation of the camera will use to get the FADC and trigger response due to the NSB. To install the program one should do the following:

1. Change to StarResponse directory:
   cd MagicProgs/Simulation/Detector/StarResponse
2. Check the dependencies with the operating system:
   make depend
3. Compile the program using:
   make all

4. Run the program:

   starrespo starresponse.par

The “starresponse.par” file has several parameters given through input commands, that are needed to run “starrespo”. They are the following:

1. Path where the outcoming data base will be stored.

   database_path /hd02/Maggi/StarLight/

2. Values of photoelectrons per ns for which the data base will be generated: lower limit, upper limit and precision step.

   simulated_phe 0.0 50.0 0.1

3. Shape for the single photoelectron trigger response: shape identification (0 means gaussian), amplitude and FWHM.

   trig_properties 0.0 1.0 2.0

4. Shape for the single photoelectron FADC response: shape identification (0 means gaussian), amplitude and FWHM.

   trig_properties 0.0 4.0 5.0

5. The “starrespo” program writes root files, which are not needed but have interesting histograms for NSB studies, if this command is in the input card.

   write_root

6. Finally, there is the end file command, which marks the end of the input card. Any thing after it will not exist for the starrespo program.

   end_file

8 Appendix B

Here are all the files of the Camera project. One can check before trying to compile or whatever, that there is the whole set of files.

MagicProgs/Simulation/Detector/Camera/LinkDef.h

MagicProgs/Simulation/Detector/Camera/MDiag.cxx

MagicProgs/Simulation/Detector/Camera/MDiag.h

MagicProgs/Simulation/Detector/Camera/MDiagLinkDef.h

MagicProgs/Simulation/Detector/Camera/MDiagdict.cxx

MagicProgs/Simulation/Detector/Camera/MDiagdict.h

MagicProgs/Simulation/Detector/Camera/Makefile

MagicProgs/Simulation/Detector/Camera/README

MagicProgs/Simulation/Detector/Camera/camera-v.h

MagicProgs/Simulation/Detector/Camera/camera.cxx

MagicProgs/Simulation/Detector/Camera/camera.h

MagicProgs/Simulation/Detector/Camera/config.mk.Linux
MagicProgs/Simulation/Detector/Camera/config.mk.linux-gnu
MagicProgs/Simulation/Detector/Camera/config.mk.osf1
MagicProgs/Simulation/Detector/Camera/creadparam.cxx
MagicProgs/Simulation/Detector/Camera/creadparam.h
MagicProgs/Simulation/Detector/Camera/input.card
MagicProgs/Simulation/Detector/Camera/moments.cxx
MagicProgs/Simulation/Detector/Camera/moments.h
MagicProgs/Simulation/Detector/ReflectorII/Makefile
MagicProgs/Simulation/Detector/ReflectorII/atm.c
MagicProgs/Simulation/Detector/ReflectorII/atm.h
MagicProgs/Simulation/Detector/ReflectorII/attenu.f
MagicProgs/Simulation/Detector/ReflectorII/config.mk.linux-gnu
MagicProgs/Simulation/Detector/ReflectorII/config.mk.osf1
MagicProgs/Simulation/Detector/ReflectorII/diag.c
MagicProgs/Simulation/Detector/ReflectorII/diag.h
MagicProgs/Simulation/Detector/ReflectorII/geometry.c
MagicProgs/Simulation/Detector/ReflectorII/geometry.h
MagicProgs/Simulation/Detector/ReflectorII/header.c
MagicProgs/Simulation/Detector/ReflectorII/header.h
MagicProgs/Simulation/Detector/ReflectorII/init.c
MagicProgs/Simulation/Detector/ReflectorII/init.h
MagicProgs/Simulation/Detector/ReflectorII/input
MagicProgs/Simulation/Detector/ReflectorII/lagrange.h
MagicProgs/Simulation/Detector/ReflectorII/parms.c
MagicProgs/Simulation/Detector/ReflectorII/parms.h
MagicProgs/Simulation/Detector/ReflectorII/ph2cph.c
MagicProgs/Simulation/Detector/ReflectorII/reflector.c
MagicProgs/Simulation/Detector/ReflectorII/version.h
MagicProgs/Simulation/Detector/StarResponse/MStarLight.cxx
MagicProgs/Simulation/Detector/StarResponse/MStarLight.hxx
MagicProgs/Simulation/Detector/StarResponse/Makefile
MagicProgs/Simulation/Detector/StarResponse/config.mk.Linux
MagicProgs/Simulation/Detector/StarResponse/config.mk.linux-gnu
MagicProgs/Simulation/Detector/StarResponse/config.mk.osf1
MagicProgs/Simulation/Detector/StarResponse/srreadparam.cxx
MagicProgs/Simulation/Detector/StarResponse/srreadparam.h
MagicProgs/Simulation/Detector/StarResponse/starresponse.cxx
MagicProgs/Simulation/Detector/StarResponse/starresponse.hxx

MagicProgs/Simulation/Detector/StarResponse/starresponse.par
MagicProgs/Simulation/Detector/include-GENERAL/DllImport.h
MagicProgs/Simulation/Detector/include-GENERAL/RConfig.h
MagicProgs/Simulation/Detector/include-GENERAL/Rtypes.h
MagicProgs/Simulation/Detector/include-GENERAL/jcdebug.h
MagicProgs/Simulation/Detector/include-GENERAL/jcmacros.h
MagicProgs/Simulation/Detector/include-GENERAL/lagrange.h
MagicProgs/Simulation/Detector/include-GENERAL/ranlib.h
MagicProgs/Simulation/Detector/include-CORSIKA/COREvent.cxx
MagicProgs/Simulation/Detector/include-CORSIKA/COREvent.hxx
MagicProgs/Simulation/Detector/include-CORSIKA/COREventHeader.cxx
MagicProgs/Simulation/Detector/include-CORSIKA/COREventHeader.hxx
MagicProgs/Simulation/Detector/include-CORSIKA/CORParticle.cxx
MagicProgs/Simulation/Detector/include-CORSIKA/CORParticle.hxx
MagicProgs/Simulation/Detector/include-CORSIKA/CORStatfile.cxx
MagicProgs/Simulation/Detector/include-CORSIKA/CORStatfile.hxx
MagicProgs/Simulation/Detector/include-MC/MCCphoton.cxx
MagicProgs/Simulation/Detector/include-MC/MCCphoton.hxx
MagicProgs/Simulation/Detector/include-MC/MCEventHeader.cxx
MagicProgs/Simulation/Detector/include-MC/MCEventHeader.hxx
MagicProgs/Simulation/Detector/include-MLons/MLons.cxx
MagicProgs/Simulation/Detector/include-MLons/MLons.hxx
MagicProgs/Simulation/Detector/include-MTrigger/MGTriggerSignal.cxx
MagicProgs/Simulation/Detector/include-MTrigger/MGTriggerSignal.hxx
MagicProgs/Simulation/Detector/include-MTrigger/MTrigger.cxx
MagicProgs/Simulation/Detector/include-MTrigger/MTrigger.hxx
MagicProgs/Simulation/Detector/include-MTrigger/MTriggerDefine.h
MagicProgs/Simulation/Detector/include-MTrigger/TABLE_NEXT_NEIGHBOUR
MagicProgs/Simulation/Detector/include-MTrigger/TABLE_PIXELS_IN_CELLS
MagicProgs/Simulation/Detector/include-MFadc/MFadc.cxx
MagicProgs/Simulation/Detector/include-MFadc/MFadc.hxx
MagicProgs/Simulation/Detector/include-MFadc/MGFadcSignal.cxx
MagicProgs/Simulation/Detector/include-MFadc/MGFadcSignal.hxx
MagicProgs/Simulation/Detector/lib/README
MagicProgs/Simulation/Detector/lib/libranlib.a.linux
MagicProgs/Simulation/Detector/lib/libranlib.a.osf1
MagicProgs/Simulation/Detector/Data/ct1-perfect.def
MagicProgs/Simulation/Detector/Data/ct1.def
MagicProgs/Simulation/Detector/Data/focals.dat

MagicProgs/Simulation/Detector/Data/magic-perfect.def
MagicProgs/Simulation/Detector/Data/magic.def
MagicProgs/Simulation/Detector/Data/magic.def.jc
MagicProgs/Simulation/Detector/Data/pixels.dat
MagicProgs/Simulation/Detector/Data/qe-emi.RFL.dat
MagicProgs/Simulation/Detector/Data/qe-emi.dat
MagicProgs/Simulation/Detector/Data/qe-ipd.RFL.dat
MagicProgs/Simulation/Detector/Data/qe-ipd.dat
MagicProgs/Simulation/Detector/Data/reflectivity-100.dat
MagicProgs/Simulation/Detector/Data/reflectivity-90.dat
MagicProgs/include-Classes/MBase/BaseIncl.h
MagicProgs/include-Classes/MBase/BaseLinkDef.h
MagicProgs/include-Classes/MBase/MAGIC.h
MagicProgs/include-Classes/MBase/MArray.cc
MagicProgs/include-Classes/MBase/MArray.h
MagicProgs/include-Classes/MBase/MArrayB.cc
MagicProgs/include-Classes/MBase/MArrayB.h
MagicProgs/include-Classes/MBase/MArrayS.cc
MagicProgs/include-Classes/MBase/MArrayS.h
MagicProgs/include-Classes/MBase/MEvtLoop.cc
MagicProgs/include-Classes/MBase/MEvtLoop.h
MagicProgs/include-Classes/MBase/MInputStreamID.cc
MagicProgs/include-Classes/MBase/MInputStreamID.h
MagicProgs/include-Classes/MBase/MParContainer.cc
MagicProgs/include-Classes/MBase/MParContainer.h
MagicProgs/include-Classes/MBase/MParList.cc
MagicProgs/include-Classes/MBase/MParList.h
MagicProgs/include-Classes/MBase/MReadTree.cc
MagicProgs/include-Classes/MBase/MReadTree.h
MagicProgs/include-Classes/MBase/MTask.cc
MagicProgs/include-Classes/MBase/MTask.h
MagicProgs/include-Classes/MBase/MTaskList.cc
MagicProgs/include-Classes/MBase/MTaskList.h
MagicProgs/include-Classes/MBase/MTime.cc
MagicProgs/include-Classes/MBase/MTime.h
MagicProgs/include-Classes/MBase/Makefile
MagicProgs/include-Classes/MRawFormat/MRawCrateArray.cc
MagicProgs/include-Classes/MRawFormat/MRawCrateArray.h
MagicProgs/include-Classes/MRawFormat/MRawCrateData.cc

MagicProgs/include-Classes/MRawFormat/MRawCrateData.h
MagicProgs/include-Classes/MRawFormat/MRawEvtData.cc
MagicProgs/include-Classes/MRawFormat/MRawEvtData.h
MagicProgs/include-Classes/MRawFormat/MRawEvtHeader.cc
MagicProgs/include-Classes/MRawFormat/MRawEvtHeader.h
MagicProgs/include-Classes/MRawFormat/MRawEvtPixelIter.cc
MagicProgs/include-Classes/MRawFormat/MRawEvtPixelIter.h
MagicProgs/include-Classes/MRawFormat/MRawFileRead.cc
MagicProgs/include-Classes/MRawFormat/MRawFileRead.h
MagicProgs/include-Classes/MRawFormat/MRawFileWrite.cc
MagicProgs/include-Classes/MRawFormat/MRawFileWrite.h
MagicProgs/include-Classes/MRawFormat/MRawRunHeader.cc
MagicProgs/include-Classes/MRawFormat/MRawRunHeader.h
MagicProgs/include-Classes/MRawFormat/Makefile
MagicProgs/include-Classes/MRawFormat/MerppLinkDef.h
MagicProgs/include-Classes/MRawFormat/RawIncl.h
MagicProgs/include-Classes/MRawFormat/RawLinkDef.h
MagicProgs/include-Classes/MMcFormat/MFadcDefine.h
MagicProgs/include-Classes/MMcFormat/MMcEvt.cxx
MagicProgs/include-Classes/MMcFormat/MMcEvt.h
MagicProgs/include-Classes/MMcFormat/MMcFadcHeader.cxx
MagicProgs/include-Classes/MMcFormat/MMcFadcHeader.hxx
MagicProgs/include-Classes/MMcFormat/MMcTrig.cxx
MagicProgs/include-Classes/MMcFormat/MMcTrig.hxx
MagicProgs/include-Classes/MMcFormat/MMcTrigHeader.cxx
MagicProgs/include-Classes/MMcFormat/MMcTrigHeader.hxx
MagicProgs/include-Classes/MMcFormat/MTrigerDefine.h
MagicProgs/include-Classes/MMcFormat/Makefile
MagicProgs/include-Classes/MMcFormat/McIncl.h
MagicProgs/include-Classes/MMcFormat/McLinkDef.h
MagicProgs/include-Classes/MMcFormat/Mdefine.h
MagicProgs/include-Classes/MMcFormat/readme.txt