M. Guidal
July 1993
CEBAF/University of South Carolina

CLAS NOTE-93-013

CLAS315
User Manual

Contents

1 INTRODUCTION

CLAS315 is the ULTRIX-GEANT315 simulation package for the CLAS detector. It allows the digitized data (detector components ID and their corresponding parameters (ADC, TDC)) to be written in CODA format on disk (if desired).
All the files needed necessary to compile and run CLAS315 are available at CEBAF on CLAS01 in the directory "/clas01/usr/local/classw/GEANT". The directory will be regularly updated to be compatible with the latest versions of GEANT.
This document describes the operation of CLAS315 on "CLAS01" at CEBAF.

2 GENERAL ORGANIZATION

The compilation uses YPATCHY and the sources of the code which are modifiable and accessible to the user are in the files 'glas.car', 'ch.cra', 'cc.cra', 'sc.cra', 'ec.cra' and 'my.cra' : this choice has been made because it appears that it is sometimes (rarely) necessary to make modifications in the "deep GEANT" subroutines (beginning by a "G") which are not directly accessible to the user (they are contained in the 'gean(xxx)315.car' files in 'cernlib') and this can be done only by this way.
For those who are not familiar with this, there is no real difference with a usual fortran code, just note that the user 'COMMON' blocks are represented by '+SEQ,...' in the subroutines and are defined by a '+KEEP,...' at the beginning of the file 'glas.car'.
This latter file contains all the general routines : the initialisation and termination routines (UGINIT, UGLAST), the geometry definition of the "non-active" volumes ("mother" volumes, magnets, shielding,...) and various routines which are called by GEANT in the course of an event generation (GUKINE, GUTREVE, GUSTEP,...).
'ch.cra', 'cc.cra', 'sc.cra' and 'ec.cra' are 4 files specific to the 4 "active" detector packages : respectively, the drift chambers, the cerenkovs, the time-of-flight counters (scintillators) and the electromagnetic calorimeter. Each of these files contains 4 subroutines : GExx defines the geometry, xxSET defines the package as a sensitive set and the "hit" and "digitization" parameters, xxHITS stores the hits during the tracking process and xxDIGI calculates and stores the hits.
The code is organized so it can easily be modified for personnal applications : but make sure you make the changes in the '.cra' and '.car' files rather than in the '.f' (for *FORTRAN*) files subsequently created for the *FORTRAN* files will be overwritten the next time *YPATCHY* is run.
The file 'codageant.car' contains the subroutines which allow GEANT to write the digitized events in the CODA format : it is a convenient format to store the information as it can easily be read by other programs ('SDA' for the reconstruction, 'CED' for the event display, 'EC' for the electromagnetic showers analysis, 'xcefdmp' for event browsing). Another word concerning the file ('gxint.cra') : normally it should not need modifications but through it you can modify the "deep GEANT" subroutines I previously mentioned ; this is needed to correct bugs in GEANT : you will find there per example a correction for the subroutine GNSPHR (designed to compute a distance up to the border of a "spherical" volume...).

'mf.coda' is the makefile and links with "cernlib93c", lX11, the "coda-output" libraries (IOLIB),...
You will find a list of all those files in appendix A.
If you have an account on CEBAF9, CLAS01 or USCLAS, no changes are needed to create CLAS315 : just copy the directory and type 'make -f mf.coda' to recompile after a change in any file...
If transported to another site, obviously the different logical paths ("$(LIB)" for "cernlib", "$(IOLIB)" for the coda routines) must be redefined and the adresses of the different GEANT source files ("geanxxx315.car") figuring after the YPATCHY instruction "PAM" in the "cradle" files modified to match their new site location.

3 RUNNING CLAS315

Two modes are possible : batch and interactive. Both will prompt you at the beginning of the execution to read the "DATA CARDS" which are switches allowing you to turn 'ON' or 'OFF' various options or features of the program. You then have to type "read 33" as the 'data cards' are contained in the 'fort.33' file.
I will describe here the main ones, a more complete description is provided in appendix B :

- The card KINE allows you to choose the kinematics of the triggered particle. The parameters are the following :

  KINE *ikine pkine(1) pkine(2)...pkine(7)...*

  If '*ikine*' is '3' then the meaning of the '*pkine*'s array elements are as follows :

  KINE *3 id p_min p_max th_min th_max ph_min ph_max*

  id : GEANT code for the particle (ex : 1=photon, 3=electron, cf. appendix C)
  p_xxx : total momentum of the particle
  th_xxx : theta and *ph_xxx* : phi

  ikine = 99 corresponds to a "background" event generator, it is more precisely described in section 5.
  The user is free to define new event generators ; he should then assign them a corresponding *ikine*, when writing the code.

- You can choose if you want a magnetic field or not :
  - **MAGN** 'OFF ' will turn it off
  - **MAGN** 'TABL' will read the file(s) containing the components of the field(s). If you select this option, you have the choice between 3 files (corresponding to 3 kinds of magnetic field) :

    - **SWIT 9** = 0 : Only Big Torus
      ⇒ file 'fort.35'

    - **SWIT 9** = 1 : Mini-Torus + Big Torus
      ⇒ file 'fort.36'

    - **SWIT 9** = 2 : Pol. Targ + Big Torus
      ⇒ files 'fort.40' & 'fort.41' (resp. big & small grid)

- You can write the output of the events in CODA format with :

$$\textbf{CODA} = 1$$

You will then be prompted, during the initialization phase of the execution for the name of the output file ; the convention is to give it a name with a '.evt' extension.

3.1 BATCH JOBS

To run a batch job, you have to type : CLAS315 -b nameprog.kumac where 'nameprog.kumac' is a file containing the instructions you want to execute. Per example,

kine 3 3 1. 1. -45. 0. 0. 0.
trig 1000
exit

would trig 1000 electrons of 1GeV at random Theta and Phi=0.
The standard ULTRIX commands can help pipe in and out the different input parameters. For instance :

CLAS315 -b batch < *clas.in* > *clas.out*

would execute the file *'batch.kumac'* (in typing the command the extension '.kumac' can be omitted) taking the input from the file *'clas.in'* and redirecting the output to *'clas.out'*. The file *'clas.in'* would look like :

read 33
output.evt (optional)

'output.evt' stands for the name of the file CLAS315 will write the digitized parameters to (if requested by the data card CODA).

3.2 INTERACTIVE VERSION

The GEANT interactive version code is contained in the file 'gxint.f', which is created by 'gxint.cra'. It controls the interactive execution of most of the GEANT subroutines (definition of volumes, matrices, media, display of structures, cross-section tables, tracks, kinematics generation, tracking,...). A complete menu and description of the commands is available by typing 'help'.
A typical session would be :

clas01>*CLAS315*

The prompt then asks you for :

- the type of terminal you are using,

- the file containing the DATA cards (ex : type 'read 33' to read unit 33, i.e. the file named 'fort.33'),

- if selected, the name of the file containing the outputs of GEANT in the CODA format.

The prompt **GEANT**> then appears and waits for your instructions ; several useful ones could be :

- dcut las- 2 0 10 10 .015 .015 *(cut view of CLAS).*
- trig *(triggers a particle with the kinematics specified in the data cards).*
- dxyz *(draws the track on the screen).*
- dhits *(draws the hits in the different detectors).*
- dpart *(prints the name of the particle).*
- pdigi *(print the id of the hit detectors and their digitized output).*
- next *(clears the screen).*
- dtree las- 0 111 *(draws the structural 'tree' of the volumes).*
- exit.

Numerous other commands and operations are possible (zoom, drawings with hidden parts, "on-line" histogram display,...), type 'help' to have a list of them and know how to use them. I just want to stress here a bit the command 'meta' which allows the creation of metafiles because it might seem a bit tricky ; you have to follow these specific steps :

GEANT>meta *id* -111 *picture*
GEANT>next
GEANT> *"command(s) drawing your picture"*
GEANT>close *id*

where *id* can be in principle any integer superior to 10 : the name of the created file containing the picture will then be 'fort.*id*'. There is therefore a little constraint : *id* must be different from the numbers already used for the 'fort.xx' files figuring in the current directory (i.e., units 33, 35, 36,...). *picture* is a dummy character variable (apparently useless but needed internally).

4 DESCRIPTION OF THE DIGITIZED OUTPUTS

The digitized outputs of the different packages follow the CODA standards ("id" of the detectors, number of ADCs and TDCs per package,...) ; Though, a lot remains to be done regarding the digitization and in general the ADC corresponds to the energy loss (in keV or MeV according to the detector) of the particle in the considered "volume" and the TDC to the time of flight (in ns) at the entrance of this volume. A bit has been done for the drift chambers where the drift time is taken into account (via a calculus of the distance of closest approach to the wire) and for the time of flight counters where the time of propagation to the PMs at the ends of the scintillators is calculated :

'CH ' 1 ADC (Eloss in keV), 1 TDC (Drift time in ns)

'CC ' 1 ADC (Eloss in keV), 1 TDC (Time of flight in ns)

'SC ' 2 ADCs (Right and Left, Eloss in keV), 2 TDCs (Right and left, Time of flight + propagation time in ns)

'EC ' 1 ADC (Eloss in MeV), 1 TDC (Time of flight in ns)

5 OVERVIEW OF SELECTED SUBROUTINES

5.1 "MYSTUFF"

The file 'my.cra' contains 8 subroutines which allow the user to add his personnal applications to the existing program. For them to be called, it is necessary to add 'MY ' to the data card "GEOM", "SETS", "DIGI" in the 'fort.33' file. Those subroutines are :

- 'MYINIT' : called at the beginning of the program ; user may define his own 'data cards', initialize his own histograms (HBOOK),...
- 'MYEVNT' : called at the end of each event ;
- 'MYLAST' : termination routine called at the end of the run ;

In the same way, you can add some 'volumes' to the preexisting ones :

- 'MYGEOM' : add your own geometry ;
- 'MYSET': define the active part of the volumes defined in 'MYGEOM' ;
- 'MYHITS' : store the hits of the active parts defined in 'MYGEOM' ;
- 'MYDIGI' : store the digitized data of 'MYHITS'.

The user may 'enter' his own event generator through the subroutine 'KIUSER'. The program will call this subroutine if the argument of the data card **KINE** is 99 (i.e., type '**KINE** 99' in the 'fort.33' file).
A subroutine 'MYPKG_ADD' also allows the user to store in CODA format "his" digitized events. This particular subroutine is in the file 'codageant.car' since this file is meant to contain all the code specific to CODA.

5.2 "BACKGR"

An electromagnetic background event generator has been included in 'glas.car'. It is called if the argument *ikine* of the previously described data card KINE is negative. In this case, for each event, both the background and the event generator corresponding to KINE abs(*ikine*) are called, per example :

KINE -3 3 1 1 30 0 0 0

will trigger the electromagnetic background along with the event corresponding to KINE 3 3 1 1 30 0 0 0 (i.e., a 1 GeV electron at theta=30° and phi=0°).
The background is obviously dependent on the luminosity of the beam and the target. For that purpose, 3 data cards, LUMI, ATAR, ZTAR (the two latter ones corresponding to the "A" and "Z" of the target) have been created. According to the target, subroutine BACKGR opens a file ('fort.21' per example for liquid hydrogen) which contains

histograms of the angular and energy distributions of photons, positrons and electrons generated by EGS4. BACKGR then generates the electromagnetic background according to those distributions.

5.3 "ch_geom"

This subroutine is contained in the file *ch_geom.f*. It is called at initialization time and defines the drift chambers wire parameters (position, length,...). Each single wire evidently cannot be represented by a "physical" GEANT volume (too small, too many wires) and their parameters are therefore just stored in arrays (ch_r(i), ch_dlen(i), ... see their meaning in *chgeom.cmn*) which will be used in subroutine CHDIGI to calculate the distances of closest approach from the 'hits' in the drift chambers. The subroutine *ch_geom* is also used in SDA (the Standard Data Analysis Program), that's why it is not included with the other subroutines. This is part of a long-range view where they will be a common data base for all the software codes (SDA, CED, EC,...).

6 ONE LAST WORD

CLAS315 is the first released version of CLAS-UNIX-GEANT315. Some checking and debugging has been made, but there might still be some errors here and there. I'd be glad to have some 'feedback' from users ; may you have suggestions, comments or problems, please contact me :

Tel : (804) 249-74-32
Email : GUIDAL@CEBAF2.GOV

DESCRIPTION OF THE FILES

- **CLAS315** : the executable file
- **EVGEN.PAR** : parameters needed for the EVGEN routines
- **batch.kumac** : contains the commands to be executed for a batch job
- **xx.cra** : 'xx' stands for 'ch', 'cc', 'sc', 'ec' ; contains the subroutines GExx, xxSET, xxHITS & xxDIGI in the YPATCHY form
- **xx.f** : contains the subroutines GExx, xxSET, xxHITS & xxDIGI in FORTRAN (created automatically from xx.cra)
- **ch_geom.f** : subroutine 'ch_geom'
- **chgeom.cmn** : common blocks for 'ch_geom.f'
- **codageant.car** : contains all the subroutine dealing with the output event buffer (fetch the digitized events from GEANT and stores them in the CODA format).
- **fort.21** : contains 3 2d-histograms (angular-energy distributions of background electrons, photons, positrons). Needed when subroutine BACKGR is called.
- **fort.33** : the data cards
- **fort.35** : magnetic field table (TORUS)
- **fort.36** : magnetic field table (TORUS+MINITORUS)
- **fort.40** : magnetic field table (Pol. Tar. Field+TORUS, big grid)
- **fort.41** : magnetic field table (Pol. Tar. Field+TORUS, small grid)
- **fort.42** : magnetic field table (Pol. Tar. Field)
- **glas.car** : common blocks definition & general subroutines in YPATCHY form
- **glasexe.cra** : creates 'glasexe.f' from 'glas.car' & 'codageant.car' via YPATCHY
- **glasexe.f** : FORTRAN file created by 'glasexe.car'
- **gxint.cra** : creates 'gxint.f' from the CERNLIB 'gean...315.car' files (not accessible)
- **gxint.f** : FORTRAN file created by 'gxint.cra' ; contains the MAIN program and all the "INTERACTIVE" subroutines
- **mf.coda** : the makefile

DESCRIPTION OF THE DATA CARDS

The following 14 data cards describe the physics processes :

0 = turns them off

1 = turns them on

Other values besides 0 and 1 are possible ; it is not possible to describe them here, cf GEANT315 manual. The following values in parenthesis are the recommended values if you want the physics "on".

DCAY : control decay mechanism

ANNI : control bremstrahlung

BREM : control bremstrahlung

COMP : control Compton scattering

DRAY : control delta rays mechanism

HADR : control hadronic interactions

LOSS : control energy loss

MULS : control multiple scattering

MUNU : control muon nuclear interactions

PAIR : control pair production mechanism

PFIS : control photo fission mechanism

PHOT : control Photo effect

RAYL : control Rayleigh scattering

AUTO : automatic computation of some parameters for MULS, should always be 1 for the non-familiar users

ERANG 1.E-6 50.0 100. : range of the cross-section tables (Emin,Emax (in GeV), nbin)

CUTS 12=1.E-6 13=1.E-6 14=1.E-4 15=1.E-4 : energy cuts (in GeV) (resp. for photons, electrons, neutrons and hadrons)

The 2 following *switches* (5 & 6) "go together" :

SWIT 5 : 1 = on-line drawing of the track

2 = on-line printing (*step by step*) of the parameters of the track(energy, position, destep,...)

SWIT 6 : number of the track to be drawn (**SWIT 5 = 1**) or printed (**SWIT 5 = 2**) ; **SWIT 6 = 0** acts on *all* the tracks of the event

The 2 following data cards (**MAGN** & **SWIT 9**) "go together" :

MAGN : 'TABL' = reads the magnetic field table corresponding to *swit* 9
'OFF ' = no magnetic field (**SWIT** 9 has then no meaning

SWIT 9 : 0 = TORUS field
1 = TORUS + MINITORUS field
2 = Pol. Tar. field + TORUS field
3 = Pol. Tar. field

VERT *id x y z* : defines the position of the vertex (source) *id.* **VERT** 1 is by default 0. 0. 0.

KINE *ikine pkine(1) pkine(2) ...* : defines the kinematics of the event to be trigged ; several meanings possible for the *pkine* according to *ikine*

GEOM : selects the volumes to be taken into account ; 'MG ' 'SHI ' 'CH ' 'CC ' 'SC ' 'EC ' 'MY ' are possible

SETS : selects the active volumes ; 'CH ' 'SC ' 'EC ' 'CC ' are possible

DIGI : selects the hits to be digitized ; 'CH ' 'SC ' 'EC ' 'CC ' are possible

RNDM : number of the random generator ; any number is possible. If not changed, it will reproduce the same events (for the same initial parameters) for each run

PREC : allows a greater precision in the tracking (via a more accurate definition of the parameters controlling **MULS**) ; 'CH ' 'CC ' 'SC ' 'EC ' are possible

CODA : 1 = stores the digitized events in the raw event buffer
0 = no saving !

The three following parameters are meaningful only if the background event generator has been called (i.e., *ikine* of **KINE**<0) :

LUMI 1.E33 : luminosity of the beam

ATAR 1. : atomic number of the target

ZTAR 1. : charge of the target