

Monte Carlo Status

- Functional modularized Monte Carlo- RCP files providing data-driven geometry.
- Well documented on the Web.

E907 MC documentation

- **E907MC-DATA DRIVEN GEANT for the experiment E907**
 - **Rajendran Raja**
 - 11/26/2001 2:56:03 PM
- E907MC is a version of GEANT 3.21 that is data driven. The program contains no hard-coded geometry constants. The Geometry and Detectors systems are fed to the program using RCP files that are structured text files. The main routine is called hanuman.F. The program works on SGI/UNIX and LINUX. The program is built using Makefiles. All fortran routines are in the style of CERNLIB (i.e. *.F). They can contain compile time switches and common blocks are introduced via the #include construction.
 - • • [Geant Documentation](#)
 - • • [RCP files](#)
 - • • [RCPE files](#)
 - • • [Declaring RCP and RCPE files to the program](#)
 - • • [Fortran Coding Standards](#)
 - • • [Preparing your area](#)
 - • • [Building the needed libraries](#)
 - • • [Building and running the E907MC executable](#)
 - • • [Library Linking Order and the ldd routines](#)
 - • • [Analyzing and Playing Back events](#)
- - [**GEANT Documentation**](#)

RCP files

- RCP files
- RCP files are inherited from the D0 experiment Run I code. These are structured ASCII files that can be used to input data to programs in an orderly form. Fortran programs can read in variables from RCP files from any subroutine without use of common blocks. The constants are kept in ZEBRA banks and are made available to the user by means of a fast binary search among the bank contents. RCP files can handle all common data types, Integer *4, Character ,Real*4 and Logicals. The general format of the file is a variable name (up to 32 characters long) followed by a value. Character values are enclosed in single quotes. Exclamation point (!) signifies that the remainder of the line is a comment. In addition to single characters, one can also specify arrays of variables by the declaration
- `\array real_nums 0.7 3.9 5.5 ! This line contains an array of 3 real numbers`
`\end` The primary RCP file controlling the program is called HANUMAN.RCP. Please browse through it by following the hyperlink.
- This file has an array called
- `\ARRAY DETECTOR_SYSTEMS 'MOTHERS' !idtype 0-999`
- `'JGGIANT' !idtype 1000-1999`
- `'BEAM' !idtype 2000-2999`
- `'TARGET' !idtype 3000-3999`
- `'TPC' !idtype 4000-4999`
- `'CHAMBERS' !idtype 5000-5999`
- `'CKOV' !idtype 6000-6999`
- `'ROSY' !idtype 7000-7999`
- `'TOF' !idtype 8000-8999`
- `'RICH' !idtype 9000-9999`
- `'CALORIMETER' !idtype 10000-10999`
- `\END`

RCP usage

- In any Fortran program, can use the following construct.— No need for common blocks.

```
LOGICAL FIRST
  DATA FIRST/.TRUE./
  SAVE FIRST
C-----
  IF(FIRST)THEN
C   ALL EZPICK('HANUMAN_RCP')
    CALL EZGET('PRINT_MATERIALS',PRMAT,IER)
    CALL EZGET('DO_INTERACTION_LENGTHS',DLAM,IER)
    CALL EZRSET
  ENDIF
```

- Will handle Arrays, logicals, reals and integers. Comments are easy.
- Great to control programs with (including Offline)

Detector description

Structure of detector RCP files

Each detector RCP file must contain the following RCP file structures.

\\ARRAY ROTATION_MATRICES !Interrupt [here](#) to read more about this structure

\\ARRAY MATERIAL_LIST !Interrupt [here](#) to read more about this structure

\\ARRAY MIXTURE_LIST !Interrupt [here](#) to read more about this structure

\\ARRAY MEDIA_LIST !Interrupt [here](#) to read more about this structure

\\ARRAY VOLUME_LIST !Interrupt [here](#) to read more about this structure

\\ARRAY DETECTOR_LIST !Interrupt [here](#) to read more about this structure\\

ARRAY CERENKOV_LIST !Interrupt [here](#) to read more about this structure

These arrays define the rotation matrices, materials, mixtures, tracking media, geometrical volumes and detector sets belonging to the detector subsystem in question.

RCPE files

RCP files can become somewhat large. Very often one wants to keep most of the RCP file but may want to change a few of the variables as the running conditions of the program change. This is done using RCPE files (stands for RCP Edit). As an example, the RCP file [hanuman.rcp](#) can be edited by a file called [hanuman.rcpe](#). The variables in hanuman.rcp will be unaffected if they are not over-ridden by their being declared in hanuman.rcpe.

Coding Standards

- **Fortran Coding standards**

1. Every subroutine must have a header with the Author name, Date of Creation, Modification, Purpose of routine, Input and Output.
2. All include files must be of the form `#include 'name.inc'`
3. All code must be properly indented. The editor xemacs has an automatic way of doing this. Select the area you want to indent, and type in `'esc'` followed by `'ctrl-\'`. It will make your code readable.
4. All code must be single operation per line. No Semi-colons.
5. Comments are encouraged strongly. Comments may be added at the end of a line by `!comment`.
6. All source code should be in upper case. Comments may be in lower case.
7. There should be no hard-coded constants in routines. All constants should be read in from RCP files. The programmer should familiarize him/herself with the routines EZPICK, EZGET and EZRSET.
8. The use of `IMPLICIT NONE` is mandatory in all routines. This forces declaration of all variables as `logical,real` etc.
9. The use of the construction
 - `-----`
 - `logical first`
 - `data first/.true.`
 - `save first`
 - `-----`
 - `if(first)then`
 - `first = .false.`
 - `endif`
 - Is encouraged when wanting to initialize things in routines.
10. The use of common blocks is discouraged. If one must have common blocks, they should come in as include files as described above. The three main places the include files are found are shown in the example below.
 - `#include "inc/gcphys.inc" !from $d0library/inc. Geant common blocks in $d0library/inc have variables type declared #include "listvol.inc" !from current directory (in this case dd_geant).`
 - `#include "geant321/gcdraw.inc" !from the directory $GEANT_DIR/src/geant321/geant321`
 - These paths are defined in the files Makefile and config.mk
11. All released code should reside in the CVS library.

MC To Do list

- Complete Geant control by RCP switches-
Got rid of FFREAD.
- RICH digitization– Needs putting in.
- TOF geometry and digitization need to go in.
- Wire Chamber geometry needs to be better described.
- Calorimeter needs to go in.
- RAW hits to mimic data.