Draft:Not for distribution
version: 0.90
processed: September 5, 1994
This is a working draft and contents are not fully debugged.
If any error is found, please inform it to the author.

FULL for JLC physics study

INTERNAL USE ONLY

Akiya Miyamoto*
KEK, National Laboratory for High Energy Physics
1-1 Oho, Tsukuba, Ibaraki 305, Japan

ABSTRACT

For JLC physics studies, an event generator and a simple detector simulator were developed using the TOPAZ FULL program. This paper summarize the DESCRIBE subroutines for JLC physics studies and the output data format.

*e-mail address: miyamoto@kekux1.kek.jp

Contents

1 Introduction **4**

2 DESCRIBES for JLC physics study. **4**

2.1 Names of DESCRIBE 4

2.2 SPRING 4

2.3 LUND_Frag. 6

2.4 Swimmer 7

2.5 Smear 7

2.6 SQUEZE_EVT and RECOVER_EVT 7

3 Banks in Begin_Run records **7**

3.1 DATA_TYPE 8

3.2 FULL_Analysis 8

3.3 SPRING 9

3.4 SWIMMER 9

3.5 SMEARING 10

4 Banks in the Event records **10**

4.1 Spring:Header 10

4.2 Spring:Parton_List 11

4.3 Generator:Header 11

4.4 Generator:Particle_List 12

4.5 Production:VTX;Space_Point 12

4.6 Production:CDC;Track_Parameter 12

4.7 Production:CDC_VTX;Track_Parameter 13

4.8 Production:EMC;Hit_Cell 14

4.9 Production:EMC;Cluster 14

4.10 Production:EMC;Cluster_to_Cell 15

4.11 Production:HDC;Hit_cell 15

4.12 Production:HDC;Cluster 16

4.13 Production:HDC;Cluster_to_Cell 16

4.14 G1 16

4.15 Z.Production:CDC_VTX;Track_Parameter 18

4.16 Z.Production:EMC;Hit_Cell 19
4.17 Z.Production:EMC;Cluster 19
4.18 Z.Production:EMC;Cluster_to_cell 20
4.19 Z.Production:HDC 21

1 Introduction

The TOPAZ FULL is a software system to organize the data analysis of high energy physics by analysis *module*[1]. Each module can be fairly independent, thus we can easily omit any module from the data analysis or include new analysis module. The module is defined by the subroutine called as *DESCRIBE* subroutine. In the DESCRIBE, we define the subroutine to be called at the input parameter read, the job initialization, the event analysis, and so on. Several subroutines are prepared for JLC physics study, which are discussed in the section 2. The TOPAZ FULL uses TBS[2] for data bank system. It should be also used for data communication among the FULL module. The data format are discussed in the section 3.

2 DESCRIBES for JLC physics study.

2.1 Names of DESCRIBE

The name of DESCRIBE for JLC physics study is summarized in the Table 1. The comments and input data for each module are described in the following subsections.

Table 1: JLC standard DESCRIBE and their function.

module name	Subroutine name	function
`SPRING`	`DSSPRG`	SPRING generator
`LUND_Frag.`	`DSLHAD`	Hadronization
`Swimmer`	`DSCSWM`	Swim particle in the detector
`Smearing`	`DSCSMR`	Smear detector signals
`SUQEZE_EVT`	`DSSQUZ`	Compress output data.
`RECOVER_EVT`	`DSRCOV`	Un-compress data.

2.2 SPRING

The many JLC physics event generators use the BASES/SPRING program[3] for the calculation of differential cross section and event generation of parton four momentum. In this program, the SPRING generate the parton four momentum reading the integrated result of BASES from a disk. Since the original subroutines of SPRING are not well fitted to the FULL structure[†] , we made some modification to the SPRING.

By including the DESCRIBE `DSSPRG` in the FLANLI program, SPRING is called within the frame work of FULL and the generated parton information stored in the TBS

[†]Recently, version 5.1 of BASES/SPRING[4] is released. We do not modify the original BASES/SPRING code to implement into FULL if we use the new version. The standard DESCRIBE to use this version is in preparation.

bank can be accessed by the subsequent module for hadronization, detector simulation, and analysis.

In the DESCRIBE `DSSPRG`, the subroutines `SPGPAR`, `SPRNIN`, `SPJBTM`, and `SPCALL` are defined for the parameter reading, run initialization, job termination and event generation, respectively.

The calling sequence of the subroutine SPGPAR is,

```
SUBROUTINE SPGPAR
```

The default `SPGPAR` is a dummy routine and does nothing, except to quit after reading the string "`EXIT'`". If user needs to read parameter, he should override the `SPGPAR`. The original SPRING requires two input data from FORTRAN unit number 5, namely number of events to generate and CPU time limit. Since the FULL steering takes care of these parameter, these two input data are not required any more.

The `SPRNIN` is called at the run initialization. It stores the run information in the bank `Spring` of the Begin_run record, then calls the `SPBEGN` for SPRING initialization. As a result, we can not generate events with multiple run number in a single job, but this will not hurt any one. During the SPRING initialization, the subroutine `SPINIT` is called. The calling sequence of `SPINIT` is same as the original BASES/SPRING. In the `SPINIT`, user should declare the function name of the integrand and maximum number of retry for event generation, by calling the subroutine SPFUNC. User can also store his own information at the beginning of the run in the `SPINIT`. The example of `SPINIT` is shown in the Fig. 1.

```
SUBROUTINE SPINIT
IMPLICIT REAL*8(A-H,O-Z)


EXTERNAL FUNC


MAXTRY=50
CALL SPFUNC(FUNC, MAXTRY)
store information to the bank, if necessary


RETURN
END
```

Figure 1: An example of the subroutine `SPINIT`

The subroutine `SPJBTM` is prepared to terminate SPRING at end run. Since the standard termination subroutine of SPRING , `SPTERM`, is called from `SPJBTM`, user can do their own termination task for SPRING in the `SPTERM`.

For every event, the subroutine `SPEVNT` is called. The `SPEVNT` should prepared parton four momentum according to the phase space information stored in the BASES commons, and the result should be stored in the bank "Spring:Parton_List" of the event record so that they can be hadronized and used for detector simulation by the other FULL module. The calling sequence of `SPEVNT` is,

```
SUBROUTINE SPEVNT(NRET)
```

where `NRET` is a `INTEGER*4` variable. If `NRET` is negative, the FULL modules following thereafter are not executed and the event is discarded. On the otherhand the `SPEVNT` in the library on the FACOM MSP system, namely those in a file `T#FP.GENLIB.LOAD`, does not require the argument `NRET`. Please note the difference in the calling sequence.

2.3 LUND_Frag.

The DESCRIBE LUND_Frag. is prepared for hadronize partons stored in the bank "Spring:Parton_List". The obtained stable and unstable particles are written in the bank "Generator:Particle_List". It uses LUND6.3 JETSET program for hadronization[5]. The default parameter of JETSET program can be modified as follows. The command syntax is,

array_name address value

where

array_name: The name of array where JETSET parameters are stored. As the array name `MST, PAR, MSTE, PARE, MASS, BRF`, and `IDB` is recognized (see. pp.375-382 or pp.395-399 of Ref.[6]).

address: The element address of the array of which parameter is overwrite.

value: New value for the parameter, namely *value* is stored in *array_name(address)*.

Input data should be terminated by `EXIT` to return to the FULL steering. The example to overwrite default JETSET6.3 parameters is shown in the Fig. 2, where we suppress π^0 and K_s decay and change the fragmentation parameters, $\sigma_q = 0.40$ and $a = 0.45$. Note that the all long lived unstable particles must be decayed properly prior to give the result to the detector simulator described in the following two subsections.

```
LUND_FRAG
IDB 23 0    ! Inhibit PI0 decay
IDB 37      ! Inhibit Ks decay
PAR 12 0.40 ! sigma-q for fragmentation
PAR 31 0.45 ! a for fragmentation
EXIT
```

Figure 2: Example of input data for LUND_Frag module.

2.4 Swimmer

The `Swimmer` module swim particles in the detector volume for the latter use by the Smear module to generate the detector signal. The format of the input data is

id value

where

id: The ID number of the parameter. 1 for the seed of smearing, 2 for the solenoid magnetic field strength in k-gauss. *ID* should be 0 to terminate the input.

value: The value for the parameter.

The typical example of input data is shown in the Fig. 3. When you change the magnetic field strength, make sure to use same field strength both for `Swimmer` and `Smear`.

```
Swimmer
 1   31415921    ! Initial seed for swimmer
 2   20.0        ! B field (kG)
 0
EXIT
```

Figure 3: Example of input data for Swimmer module.

2.5 Smear

The `Smear` module generates the detector signal using the track information prepared by `Swimmer` module. Make sure to use `Smear` module together with `Swimmer` module. The input data of `Smear` module overwrites the default definition of the detector geometry and performance. The input data format is same as those used for `Swimmer` module. The meanings of the parameter ID are too long to describe here, but you will be able to find the example in the file.

2.6 SQUEZE_EVT and RECOVER_EVT

The `SQUEZE_EVT` module read the bank data created by the module `SPRING`, `LUND_Frag` `Swimmer` and `Smear` and creates the new banks with compressed formats while deleting the original banks. Thus save the space of the output data. `RECOVER_EVT` module uncompresses the data compressed by `SQUEZE_EVT` module.

3 Banks in Begin_Run records

The bank data format created by the modules are described in this section and next section. This section describes the format in the Begin_Run record, and the next section

in the Event record. In the below, the elements should be 0, if their format is described explicitly.

3.1 DATA_TYPE

Element# : 1
TBPUT by : Subroutine `PRWDTP`.
Contents : To indicate type of data, contents are update in each job.
Format : I*4 , 9 words fixed length

IBUF(1)= 2 (indicate this is MC data).
(3)= 1 (indicate this is processed by Generator module).
(9)= Number of jobs which analyzed this data.

3.2 FULL_Analysis

The bank is created and put in the subroutine `PRWDTP`.

Element# : 1 + 100*(Nj-1) ; Nj=IBUF(9) of bank "Data_Type"
Contents : Contain the job informations.
Format : I*4 & Character mixed, 17 I*4 words length fixed.

IBUF(1-2)= Job start date(yymmnn) and time(msec from 0:00).
(3:6)= 16 characters, "FULL_ANALYSIS".
(7)= Version# *100 + Level#
(8:11)= 16 Characters of name of FLANLI, those set by the subroutine. `PRPDSC` in `FLANLI`.
(12)= Version number of FLANLI. Those set by subroutine. `PRPDSC` in `FLANLI`.
(13)= Version date of FLANLI. Those set by subroutine. `PRPDSC` in `FLANLI`.
(14-15)= 8 characters : CPU name who processed data. ('KEKFACOM')

Element# : 2 + 100*(Nj-1) ; Nj=IBUF(9) of bank "Data_Type".
Contents : Full module information.
Format : IBUF(7,i) I*4 & Character mixed, i=# of module.

IBUF(1-3,i)= 12 characters, module name.
(4,i)= Execution level.
(5,i)= Version level of module.
(6,i)= Version date of module.
(7,i)= Analysis quit level

Module name, version level, version date, should be set by sub. PRPVRS which should be called at job initialization module. Calling sequence of this routine is,

CALL PRPVRS(name, ivers, idate)

where

name: (C*12) Module name.

ivers: (I*4) for version # (100*version# + level)
idate: (I*4) for version date (yymmdd)

Element# : 3 + 100*(Nj-1) ; Nj=IBUF(9) of bank "Data_Type".
Contents : Constants version # (only for TOPAZ use.)

Element# : 4 + 100*(Nj-1) ; Nj=IBUF(9) of bank "Data_Type".
Contents : Input data read from FT05F001 Comment lines (@! lines) are not stored.
Format : Character, Length = 20 * (# of lines) I*4 words.

3.3 SPRING

This bank is TBCRTEed by subroutine SPRNIN (Run initialization of SPRING).

Element# : 1
TBPUT by : Subroutine SPRNIN.
Contents : Job information of SPRING.
Format : I*4 & Character 7 words fixed length

IBUF(1-2)= Date(yymmnn) and time(msec from 0:00) when SPRING was executed. .
(3-6)= 16 characters, ``SPRING''.
(7)= Version# *100 + Level#

Element# : 2
TBPUT by : Subroutine SPINIT (Generator module should store this data.).
Contents : Beam parameters.
Format : R*4 12 words fixed length

RBUF(1)= Beam energy (GeV)
(2)= 11./0. (Beamstrahlung in on/off).
(3)= version # of beam strahlung.
(4-8)= n, σ_x, σ_y, σ_z, and WIDTH parameters for beamstrahlung generation.

Element# : 3
TBPUT by : Subroutine SPINIT (Optional. Generator module should store this).
Contents : Store SPRING parameters specific to each processes.
Format : R*4

3.4 SWIMMER

The bank is TBCRTEed by subroutine SWMRIN (Run initialization of SWIMMER) and stores parameter values used by the `SWIMMER`.

Element# : 2
TBPUT by : Subroutine SWMRIN
Contents : SWIMMER job information.
Format : R*4 2 words, fixed length.

RBUF(1)= seed value for swimmer
(2)= magnetic field strength (k-gauss).

3.5 SMEARING

The bank is TBCRTEed by subroutine SMRRIN (Run initialization of SMEAR) and stores parameter values used by the `SMEAR`.

Element# : 2
TBPUT by : Subroutine SMRRIN
Contents : Smearing job information.
Format : R*4 4*NVX+77 variable length where NVX is the number of VTX layers + 1.

RBUF(1)= seed value for smearing.
(2)= magnetic field (kG).
(11-14)= R_{min}, R_{max}, Z_{min}, Z_{max}.
(15)= # sampling points.
(16)= cut on # sampling points.
(17-18)= $\sigma_{r\phi}$, σ_Z
(31-33)= Number of ϕ, θ(barrel) and r (endcap) segments of EM calorimeter.
(34-37)= R_{min}, R_{max}, Z_{min}, Z_{max}.
(38-39)= σ_E/E pedestal and at 1 GeV for (barrel EM).
(40-41)= σ_E/E pedestal and at 1 GeV for (endcap EM).
(51-53)= Number ϕ, θ(barrel), and r(endcap) segments of HD calorimeter.
(58-59)= σ_E/E pedestal and at 1 GeV for (barrel HD)
(60-61)= σ_E/E pedestal and at 1 GeV for (endcap HD)
(71)= Number of VTX layers + 1 = NVX
(72-73)= VTX resolution in ϕ and z(cm)
(74-76)= r(cm), Z-(cm), Z+(cm) of beampipe.
(77)= thickness in radiation length of beam pipe.
(78-80)= r(cm), Z-(cm), Z+(cm) of 1st VTX layer.
(81)= thickness in radiation length of 1st VTX layer.
(82-84)= r(cm), Z-(cm), Z+(cm) of 2nd VTX layer.
(85)= thickness in radiation length 2nd VTX layer.
.....
(4*NVX+74)= r(cm) of CDC inner cylinder
(+75)= Z-(cm) of CDC inner cylinder
(+76)= Z+(cm) of CDC inner cylinder
(+77)= thickness in radiation length of CDC inner cylinder

4 Banks in the Event records

4.1 Spring:Header

Element# : 1

TBPUT by : Subroutine SPEVNT
Contents : Event information..
Format : I*4, 10 words fixed.

RBUF(1)= Event number.
(2-3)= Date (yymmdd) and time (msec from 0:00).
(5-6)= Energy of e^- and e^+ after beamstrulung in MeV unit.
(7)= Ebeam MeV unit.
(8-9)= Polarization of e^- and e^+.

4.2 Spring:Parton_List

Element# : 1
TBPUT by : Subroutine SPEVNT
Contents : Parton information..
Format : R*4, 20 words fixed.

RBUF(1)= Particle serial number.
(2)= Particle ID
(3-4)= mass (GeV) and charge.
(5-8)= p_x, p_y, p_z, and E (GeV).
(12)= number of daughter partons.
(13)= particle serial number of the 1st daughter.
(14)= particle serial number of the parent daughter.
(17)= helicity.
(18)= colour single group ID. Partons with same group ID form colour singlet system.

4.3 Generator:Header

Element# : 1
TBPUT by : Hadronizer such as `LUHADR`.
Contents : Event information..
Format : R*4, 20 words fixed.

IBUF(1)= Hadronizer event number.
(2-3)= Date and time.
(4)= Generator ID (not defined yet.)
(5)= Start seed value.
(7)= Beam energy in MeV unit.

4.4 Generator:Particle_List

Element# : Equal to the track number.
TBPUT by : Hadronizer such as `LUHADR`.
Contents : Particles output by hadronizer, including stable and unstable particles depending on the case.
Format : R*4, 20 words fixed.

RBUF(1)= Particle serial number
(2)= Particle ID (PDG Format)[‡].
(3-4)= Particle mass (GeV) and charge.
(5-8)= p_x, p_y, p_z, and E (GeV)
(9-11)= Vertex coordinate, x, y, z (cm).
(12)= Number of daughter particle (=0 if stable.)
(13)= Pointer to the daughter
(14)= Pointer to the mother. If LT 0, equal to the element number of SPRING:Parton_List corresponding to this particle.
(15)= trigger timing(nsec).
(16)= particle life time (c×nsec, =0. or negative for stable.)
(17)= decay length (cm, =0. for stable.)

4.5 Production:VTX;Space_Point

Element# : Equal to the particle serial number in the bank "Generator:Particle_List".
TBPUT by : Smearing routine (`SMRVTX`).
Contents : VTX hit points data.
Format : R*4, variable length (3×number hit layers).

RBUF(1,i)= r(cm).
(2,i)= ϕ
(3,i)= z(cm)

, where $i = 1$ to number of hits layer.

4.6 Production:CDC;Track_Parameter

Element# : Equal to the track number.
TBPUT by : Smearing routine.
Contents : Smeared Charged track information.
Format : R*4, R*8, I*2, and I*4 mixed.
variable length (56+number of contributed generator tracks.)

[‡]Particle ID of unstable particle is not PDG Format. In that case, ID = 100000 + LUND6.3 Particle_ID.

RBUF(1-3)= p_x, p_y, p_z.
(4)= E assuming massless particle.
(5-7)= x, y, z of closest approach to the beam axis.
IBUF(9)= apparent charge (+/-1)
RBUF(11-15)= dρ_0(cm), ϕ_0(radian), κ(GeV^{-1}), dZ(cm), and tan λ of track parameter.
(16-18)= Pivot of track parameter, X_0, Y_0, and Z_0.
DBUF(10-24)= error matrix (lower triangle only)
RBUF(49)= not used (reserved for confidence level)
(50)= not used (reserved for χ^2)
HBUF(1,51)= Number of degrees of freedom.
(2,51)= not used (reserved for duplicate flag)
(1,52)= Number of track parameters (5 for helices)
(2,52)= not used (reserved for track finding type)
(1,53)= not used (reserved for pointer to 1-st hit)
(2,53)= not used (reserved for pointer to last hit)
RBUF(54-55)= radius of the innermost and the outmost hit
IBUF(56)= Number of contributing generator particle
(57)= Particle serial number of first generator particle.
.....

x, y, z, p_x, p_y, and p_z in RBUF(1-7) are those at the closest approach point to the beam axis regardless of where the track actually started. On the other hand, (X_0, Y_0, Z_0) in RBUF(16-18) is the innermost measured point in the CDC whose radius can be greater than the first wire layer radius for, for instance, a Ks which decayed in the midst of the CDC.

4.7 Production:CDC_VTX;Track_Parameter

Element# : Track number.
TBPUT by : Smearing routine.(SMRCVT)
Contents : CDC-VTX combined track information.
Format : R*4, R*8, I*2, & I*4 mixed fixed length (55)
RBUF(1-3)= p_x, p_y, p_z.
(4)= E assuming massless particle.
(5-7)= x, y, z of closest approach to the beam axis.
IBUF(9)= apparent charge (+/-1)
RBUF(11-15)= dρ_0(cm), ϕ_0(radian), κ(GeV^{-1}), dZ(cm), and tan λ of track parameter.
(16-18)= Pivot of track parameter, X_0, Y_0, and Z_0.
DBUF(10-24)= error matrix (lower triangle only)
RBUF(49)= not used (reserved for confidence level)
(50)= not used (reserved for χ^2)
HBUF(1,51)= Number of degrees of freedom.
(2,51)= not used (reserved for duplicate flag)
(1,52)= Number of track parameters (5 for helices)
IBUF(53)= Track type
(54)= pointer to "Production:CDC;Track_Parameter"
(55)= pointer to "Production:VTX;Space_Point"

The first 52 words have the same meanings as those of "Production:CDC;Track_Parameter". The track type is the number of linked VTX hits, if positive. Otherwise, it signals the failure of combined fitting.

track type = -1 to 0 : identical to the corresponding CDC track
= 2 : CDC-VTX combined track

Notice that there is a one-to-one correspondence between CDC tracks and CDC-VTX combined tracks: CDC track #i corresponds to CDC-VTX track #i. At present, no actual fitting is performed, instead, the track is assumed to pass exactly through the VTX hit positions and to have the momentum given by CDC.

4.8 Production:EMC;Hit_Cell

Element# : Hit cell number.
TBPUT by : Smearing routine.(SMREMC)
Contents : Smeared electro-magnetic calorimeter counter hit information.
Format : R*4 & I*4 variable length (3+number of contributed generator track).

RBUF(1)= Energy (MeV).
(2)= Cell ID (ICELL)
(3)= Number of contributed Generator track.
IBUF(4)= Generator particle number.
.....

The meaning of ICELL is as follows.

For barrel part: ICELL = 1000×ITHT + IPHI
ITHT ; Counter # in θ direction (0 < ITHT < NTHTEM-1).
IPHI ; Counter # in ϕ direction (0 < IPHI < NPHIEM-1).
For endcaps: ICELL = IZE×(1000000 + 1000×IRAD + IPHI)
IZE ; end cap sign (+/- 1).
IRAD ; Counter # in R direction (0< IRAD < NRADEM-1).
IPHI ; Counter # in ϕ direction (0< IPHI < NPHIEM-1).

where NTHTEM, NPHIEM, and NRADEM are the numbers of segments in θ, ϕ, and radial directions, respectively, which can be controlled by FT05 input stream.

4.9 Production:EMC;Cluster

Element# : Track(cluster) number.
TBPUT by : Smearing routine.(SMRCBK)
Contents : Smeared electro-magnetic calorimeter cluster information.
Format : R*4 & I*4 variable length (15+Number of contributed generator track)

RBUF(1-3)= E_x, E_y, E_z (GeV).

(4)= Energy (GeV)
(5-7)= X, Y, Z (cm)
IBUF(14)= ICELL
(15)= Number of contributed generator track.
(16)= First generator particle number
.....

4.10 Production:EMC;Cluster_to_Cell

Element# : Track(cluster) number.
TBPUT by : Smearing routine.(SMRCBK)
Contents : Cell number which forms the ele-mag cluster.
Format : I*4 variable length, equal to the number of cell belonging to the cluster.

IBUF(1)= Element number of first cell.

4.11 Production:HDC;Hit_cell

Element# : Hit cell number.
TBPUT by : Smearing routine.(SMRHDC)
Contents : Smeared hadron calorimeter counter hit information.
Format : R*4 & I*4 variable length (3+number of contributed generator track).

RBUF(1)= Energy (MeV).
(2)= Cell ID (ICELL)
(3)= Number of contributed Generator track.
IBUF(4)= Generator particle number.
.....

The meaning of ICELL is as follows.

For barrel part: ICELL = 1000×ITHT + IPHI
ITHT ; Counter # in θ direction (0 < ITHT < NTHTHD-1).
IPHI ; Counter # in ϕ direction (0 < IPHI < NPHIHD-1).
For endcaps: ICELL = IZE×(1000000 + 1000×IRAD + IPHI)
IZE ; end cap sign (+/- 1).
IRAD ; Counter # in R direction (0< IRAD < NRADHD-1).
IPHI ; Counter # in ϕ direction (0< IPHI < NPHIHD-1).

where NTHTEM, NPHIEM, and NRADEM are the numbers of segments in θ, ϕ, and radial directions, respectively, which can be controlled by FT05 input stream.

4.12 Production:HDC;Cluster

Element# : Track(cluster) number.
TBPUT by : Smearing routine.(SMRCBK)
Contents : Smeared hadron calorimeter cluster information.
Format : R*4 & I*4 variable length (15+Number of contributed generator track)

RBUF(1-3)= E_x, E_y, E_z (GeV).
(4)= Energy (GeV)
(5-7)= X, Y, Z (cm)
IBUF(14)= ICELL
(15)= Number of contributed generator track.
(16)= First generator particle number
.....

4.13 Production:HDC;Cluster_to_Cell

Element# : Track(cluster) number.
TBPUT by : Smearing routine.(SMRCBK)
Contents : Cell number which forms the hadron cluster.
Format : I*4 variable length, equal to the number of cell belonging to the cluster.

IBUF(1)= Element number of first cell.

4.14 G1

Compressed "Generator:Particle_List" bank. The subroutine `SQUZEV` creates the data. `DSSQUZ` is the describe for this routine.

Element# : 1
Contents : Number of valid words of corresponding "Generator:Particle_List" bank.
It depends on the number of daughters, etc.
Format : I*2, Number of I*2 words is equal to the number of
elements(particles) of "Generator:Particle_List".

IBUF(1)= Element number of first cell.

Meanings of the values are as follows:

=8 : There is no daughter particles and the creation coordinates of the particle is less than 10 μm from the interaction point(IP).

=9 : There are daughter particle(s) and the creation coordinates of the particle is less than 10 μm from the interaction point(IP).

=11 : There are no daughter particles and the creation coordinates of the particle is greater than 10 μm away from the interaction point(IP).

=12 : There are daughter particle(s) and the creation coordinates of the particle is greater than 10 μm away from the interaction point(IP).

=14 : There are daughter particle(s) and particle has finite life time.

Element# : 2

Contents : Compressed "Generator:Particle_List" information.

Format : I*2, I*4, R*4 mixed. All elements of "Generator:Particle_List" are packed in a single element data. The total sum of the values in the element number 1 is equal to the word length of this bank element. The original "Generator:Particle_List" information is compressed as follows.

NW=8 case:

HBUF(1,1)= RBUF(1):Particle serial number (RBUF(1))
(2,1)= RBUF(14):Pointer to the mother track. Negative when pointing the bank "Spring:Parton_List"
RBUF(2)= RBUF(2):Particle ID of PDG format.
(3-4)= RBUF(3-4):Particle mass (GeV) and charge.
(5-8)= RBUF(5-8):p_x, p_y, p_z, and E (GeV).

NW=9 case:

HBUF(1,1)= RBUF(1):Particle serial number (RBUF(1))
(2,1)= RBUF(14):Pointer to the mother track. Negative when pointing the bank "Spring:Parton_List"
RBUF(2)= RBUF(2):Particle ID of PDG format.
(3)= RBUF(3):Particle mass.
HBUF(1,4)= RBUF(4):Particle charge.
(2,4)= RBUF(12):number of daughter particles.
RBUF(5-8)= RBUF(5-8):p_x, p_y, p_z, and E (GeV).
HBUF(1,9)= RBUF(13):Pointer to the daughter particle.
(2,9)= RBUF(14):Pointer to the mother, same as the one stored in HBUF(2,1).

NW=11 case:

HBUF(1,1)= RBUF(1):Particle serial number (RBUF(1))
(2,1)= RBUF(14):Pointer to the mother track. Negative when pointing the bank "Spring:Parton_List"
RBUF(2)= RBUF(2):Particle ID of PDG format.
(3)= RBUF(3):Particle mass.
(4)= RBUF(4):Particle charge.
(5-8)= RBUF(5-8):p_x, p_y, p_z, and E (GeV).
(9-11)= RBUF(9-10):x, y, and z (cm) coordinates of particle creation point.

NW=12 case:

HBUF(1,1)= RBUF(1):Particle serial number (RBUF(1))
(2,1)= RBUF(14):Pointer to the mother track. Negative when pointing the bank "Spring:Parton_List"
RBUF(2)= RBUF(2):Particle ID of PDG format.
(3)= RBUF(3):Particle mass.
HBUF(1,4)= RBUF(4):Particle charge.
(2,4)= RBUF(12):number of daughter particles.
(5-8)= RBUF(5-8):p_x, p_y, p_z, and E (GeV).
(9-11)= RBUF(9-10):x, y, and z (cm) coordinates of particle creation point.
HBUF(1,12)= RBUF(13):Pointer to the daughter particle.
(2,12)= RBUF(14):Pointer to the mother, same as the one

NW=14 case:

HBUF(1,1)= RBUF(1):Particle serial number (RBUF(1))
(2,1)= RBUF(14):Pointer to the mother track. Negative when pointing the bank "Spring:Parton_List"
RBUF(2)= RBUF(2):Particle ID of PDG format.
(3)= RBUF(3):Particle mass.
HBUF(1,4)= RBUF(4):Particle charge.
(2,4)= RBUF(12):number of daughter particles.
(5-8)= RBUF(5-8):p_x, p_y, p_z, and E (GeV).
(9-11)= RBUF(9-10):x, y, and z (cm) coordinates of particle creation point.
HBUF(1,12)= RBUF(13):Pointer to the daughter particle.
(2,12)= RBUF(14):Pointer to the mother, same as the one.
RBUF(13)= RBUF(16):Particle life time (c$\times$nsec).
(14)= RBUF(17):Decay length(cm).

4.15 Z.Production:CDC_VTX;Track_Parameter

The banks "Production:CDC;Track_Parameter"(BANK1) and "Production:CDC_VTX;Track_Parameter"(BANK2) are compressed and merged into a single bank. Element # 1 contains the I*4 word number of each track information and Element #2 contains each track informations merged sequentially.

Element# : 1
Contents : Word count of each tracks.
Format : I*2

HBUF(i)= Number of words of i-th track.

Element# : 2
Contents : Merged track parameter information.
Format : I*2, R*4 mixed.

Each track informations are stored in a single element after data compressions as follows;

IBUF(1)(1:NW_1), IBUF(2)(1:NW_2), ... IBUF(n)(1:NW_n)

where index of IBUF corresponds to the track number, n is a total number of tracks which is equal to the number of element of the bank "Production:CDC_VTX;Track_Parameter", and NW_i is the number of words for i-th track. NW_i may be different track by track and it is stored in the element # 1 of this bank. Format of IBUF(i) is shown below.

RBUF(1-8)= Equal to RBUF(1-8) of BANK2 (p and $\mathbf{x}$) .
IBUF(9)= Equal to IBUF(9) of BANK2. (Charge).
RBUF(10-17)= Equal to RBUF(11-18) of BANK2
(Track parameters and pivot.)
IBUF(18-20)= Equal to IBUF(53-55) of BANK1
(Track type, and pointes to the bank CDC:Track_parameter and VTX:Space_Point).
IBUF(21)= Number of corresponding generator bank.IBUF(56) of bank1.
(22)= Track number of first generator track.IBUF(57) of bank1.
.....

4.16 Z.Production:EMC;Hit_Cell

Only the energy and Hit cell ID is stored in a single element. See subsection 4.8 for the meanings of cell ID.

Element# : 1.
Contents : Compressed EMC;Hit_Cell information.
Format : I*4, R*4 mixed.

RBUF(1)= Energy (MeV) of 1st cell.
IBUF(2)= Cell ID of first cell.
RBUF(3)= Energy (MeV) of 2nd cell.
IBUF(4)= Cell ID of first cell.

4.17 Z.Production:EMC;Cluster

The banks "Production:EMC;Cluster is compressed in to two elements; Element # 1 is the word count information and element #2 is cluster information.

Element# : 1
Contents : Word count of each cluster.
Format : I*2

HBUF(i)= Number of words of i-th cluster.

Element# : 2
Contents : Compressed cluster information.
Format : I*2, R*4 mixed.

Each cluster informations are stored in a single element after data compressions as follows;

IBUF(1)(1:NW$_1$), IBUF(2)(1:NW$_2$), ... IBUF(n)(1:NW$_n$)

where index of IBUF corresponds to the cluster number, n is a total number of cluster which is equal to the number of element of the bank "Production:EMC;Cluster", and NW$_i$ is the number of words for i-th cluster. NW$_i$ may be different cluster by cluster and it is stored in the element # 1 of this bank. Format of IBUF(i) is shown below. Note that number of words for i-th cluster is 9 when only one generator track contributes to the cluster.

RBUF(1-3)= RBUF(1-3):E_x, E_y, E_z (GeV).
(4)= RBUF(4):Energy (GeV)
(5-7)= RBUF(5-7):X, Y, Z (cm)
IBUF(8)= IBUF(14):ICELL
HBUF(17)= Number of contributed generator track.
(18)= First generator particle number
.....

4.18 Z.Production:EMC;Cluster_to_cell

The banks "Production:EMC;Cluster_to_cell" is compressed in to two elements; Element # 1 is the word count information and element #2 is cluster information.

Element# : 1
Contents : Word count of each cluster.
Format : I*2

HBUF(i)= Number of words of i-th cluster.

Element# : 2
Contents : Compressed cluster information.
Format : I*2, R*4 mixed.

Each cluster informations are stored in a single element after data compressions as follows;

HBUF(1)(1:NW$_1$), HBUF(2)(1:NW$_2$), ... HBUF(n)(1:NW$_n$)

where index of HBUF corresponds to the cluster number, n is a total number of cluster which is equal to the number of element of the bank "Production:EMC;Cluster", and NW$_i$ is the number of words for i-th cluster. NW$_i$ may be different cluster by cluster and it is stored in the element # 1 of this bank. HBUF(i) contains the cell number, namely the element number of "Production:EMC;Hit_Cell" contributing to the i-th cluster.

4.19 Z.Production:HDC

For HDC calorimeter, three banks "Z.Production:HDC;Hit_Cell", "Z.Production:HDC;Cluster", and "Z.Production:HDC;Cluster_to_cell" are created by the SQUEZE_EVT module from corresponding three banks (banks with their name without "Z."). Their formats are same as those for EMC cluster, just EMC in the bank name replaced by HDC.

References

1. "FULL for TOPAZ analysis", A. Miyamoto, TOPAZ Internal report, 1994.
2. "Tristan Bank System", TOPAZ Offline Group, KEK-TOPAZ-OFFLINE-86-4, April, 1986.
3. S. Kawabata, *Compt. Phys. Commun.* **41** (1986)127; T. Ishikawa, T. Kaneko, K. Kato, S. Kawabata, Y. Shimizu, and H. Tanaka, *KEK Report 92-19*, February, 1993.
4. S. Kawabata, to be published in *Comp. Phys. Commun.*
5. T. Sjöstrand and M. Bengtsson, *Compt. Phys. Commun.* **43** (1987) 367.
6. T. Sjöstrand, *Compt. Phys. Commun.* **39** (1986) 347.