The Event Generator

P. Avery, L. Garren

March 26, 1999

1 Introduction

Evgen provides as consistent a wrapper as possible around the Isajet, Pythia, and Herwig Monte Carlo event generators. Each example program contains the option of decaying the particles via QQ. The example code can be found under $EVGEN_DIR/example. All examples fill the HEPEVT common block and write events to disk using the stdhep xdr (mcfio) I/O routines. The examples are designed to generate b events at collider energies.

2 Directory Structure

Simple executables are in $MCFBIN. The source code is in $EVGEN_DIR/src and $EVGEN_DIR/inc. Input files are in $EVGEN_DIR/input. A scratch area is availble for large event files in /afs/fnal/files/scratch/bphysN, where N = 1-5.

The products cern, lund, herwig, isajet, qq, and stdhep will need to be defined. Products are defined with the command "setup $[-t|-d]$ <product> $[version]$". The flags -t and -d signify test and development versions. Use the command "ups list -a <product>" to check the status of a product.

See http://www-pat.fnal.gov/stdhep.html for a table of stdhep and related product versions. Because common blocks and particle numbers can change between event generator releases, it is important to use the correct set of products.

Evgen directory structure			
$EVGEN_DIR			
	README		(evgen overview)
	GNUmakefile		(build evgen library)
	evgen_arch		(build evgen library)
	doc		
		examples.doc	(short explanation)
		evgen.tex	(long explanation)
	example		(example area)
		b_psiks	(B to psi Ks example)
		d_kpi	(D to K pi example)
		min_bias	(minimum bias example)
		simple	(simple example)
	inc		
		file_names.inc	
		herwig_cmd.inc	
		pythia_user.inc	
		qq_flag.inc	
	input		
		isa_nobc_decay.dat	(isajet decay file if QQ used)
		jetset.dcy	(Pythia decay table if QQ used)
		user.dec	(sample user decay file for QQ)
	src		
		amass_isa.F	
		bookhistos.F	
		herwig_exam.F	
		hwg_command.F	
		hwg_open.F	
		hwg_user_init.F	
		hwg_user_reset.F	
		isajet_exam.F	
		main_args.c	
		main_fortran.F	
		pyt_command.F	
		pyt_get_seed.F	
		pyt_save_seed.F	
		pyt_user_init.F	
		pythia_exam.F	
		read_xdr.F	
		usr_end_event.F	
		usr_end_job.F	
		usr_filter.F	
		usr_filter_postqq.F	
		usr_filter_preqq.F	
$MCFBIN			
	herwig_exam		(generate herwig events)
	isajet_exam		(generate isajet events)
	pythia_exam		(generate pythia events)
	read_xdr		(example to read event file)
$MCFLIB			
	libevgen.a		
	amass_isa.o		
	herwig_exam.o		
	isajet_exam.o		
	pythia_exam.o		
	read_xdr.o		

3 Conventions

We use the following file extension conventions.

File Extension Conventions	
.hwg	Herwig input file
.isa	Isajet input file
.pyt	Pythia input file
.std	input file for read_xdr example
.lpt	line printer output file
.evt	xdr event files written by stdhep and mcfio
.f	standard fortran
.F	fortran which will be passed through the c preprocessor
.rz	hbook histogram or ntuple file

4 Running the Executables

Before running the examples, define the environmental variables QQ_USER_FILE, and ISAJET_DECAY. To run an executable, type, for instance, $MCFBIN/pythia_exam. You may define the input and output file names in the command line. Alternatively, you may use the input file to define the file names. For instance, you might type "$MCFBIN/pythia_exam -f pythia.pyt -o pythia_1.evt -l pythia_1.lpt" Note that the structure of the Isajet input file is defined by Isajet. The Herwig and Pythia input files are defined locally. Sample input files are available in $EVGEN_DIR/example.

Command Line Arguments		
-f	cmd_file	input command file
-l	lpt_file	line printer output file
-o	event_output_file	output event file
-i	event_input_file	input event file
-hb	hbook_file	output hbook histogram file
-qq		use QQ for decay
-qu	user_decay_file	QQ user decay file
-h		type help display

5 Sample Input files

Sample Herwig input file		
!B production: 1800 GeV pbar on p		
file_lpt	herwig_exam.lpt	!Output file
file_evt_wrt	herwig_exam.evt	!Output event file
file_hbk	herwig_exam.rz	!hbook output
maxev	1000	!Max events to generate
maxpr	2	!Max number of events to print
ranseed	0 0	!Random seed set to time of day
iproc	1705	!bbar quark production
part1	P	!Particle 1 ==> Proton
part2	PBAR	!Particle 2 ==> Antiproton
pbeam1	900.	!Beam energy for proton
pbeam2	900.	
ptmin	0.	!Pt range
ptmax	900.	
other possible commands		
maxer		!
iprint		!
yjmin		!
yjmax		!
q2min		!
q2max		!
emmin		!
emmax		!
qcdlam		!
rmass		!
file_evt_rd		!input event file
qq		!use QQ for hadron decay
ranseed		!two numbers
exit		!stop reading the input file
stop		!stop reading the input file

Sample Pythia input file		
!B production: 900 GeV pbar on p		
file_lpt	pythia_test.lpt	!Output print file
file_evt_wrt	pythia_test.evt	!File to write events
file_hbk	pythia_test.rz	!Hbook output
maxev	1000	!# of events to generate
frame	CMS	!Select center of mass frame
beam	PBAR	!Beam particle 1
target	P	!Beam particle 2
cms_energy	1800.0	!Center of mass energy
ranseed	0	!Set random seed using time of day
pygive	MSEL=5	!B production with massive matrix elements
other possible commands		
maxpr	1	!# of events to print
file_evt_rd	pythia_prev.evt	!input event file
qq		!use QQ for hadron decay
exit		!stop reading the input file
stop		!stop reading the input file

Sample Isajet input file
Example b bbar job
1800.,1000,2,10/
TWOJET
BEAMS
'P','AP'/
JETTYPE1
'BT', 'BB'/
JETTYPE2
'BT', 'BB'/
P
0. , 900., 0. , 900./
PT
0.01 , 850., 0.01 , 850./
NTRIES
5000/
NSIGMA
400/
END
STOP
See the Isajet manual for an explanation of options

6 StdHep

The Particle Data Group has established a standard particle numbering scheme. A standard event common block, HEPEVT, has also been established. The StdHep product is designed to convert various Monte Carlo event generator output events to a standard version of the output event common block.

The particle ID numbers between 1 and 80 are for elementary particles: quarks, gluons, leptons, gauge and higgs bosons and their supersymmetric partners. The PDG numbering algorithm for composite particles uses a signed 7 digit number for each particle: $\pm n n_r n_L n_{q_1} n_{q_2} n_{q_3} n_J$. $n_{q_{1-3}}$ are quark numbers used to specify the quark content. The rightmost digit, n_J = 2J + 1, gives the spin of the composite particle. The scheme does not cover particles of spin $J > 4$. The fifth digit, n_L, is reserved to distinguish mesons of the same total (J) but different spin (S) and orbital (L) angular momentum quantum numbers. The sixth digit, n_r, is used to label mesons radially excited above the ground state. The numbering scheme does not extend to baryons with $n > 0$, $n_r > 0$, or $n_L > 0$. Digits n_{q_2} and n_{q_3} are used for mesons, with $n_{q_1} = 0$. Digits n_{q_1}, n_{q_2}, and n_{q_3} are used for baryons. Digits n_{q_1} and n_{q_2} are used for diquarks, with $n_{q_3} = 0$. A negative number indicates an antiparticle. The states are generally listed in order of increasing mass. K_L^0 and K_S^0 are exceptions. Their assigned identification numbers are 130 and 310, respectively. A complete explanation can be found in $STDHEP_DIR/doc/PDG_numbers_98.ps or the 1998 PDG (C. Caso et. al., The European Physical Journal **C3**).

HEPEVT common block	
PARAMETER	(NMXHEP=4000)
COMMON/HEPEVT/	NEVHEP,NHEP,ISTHEP(NMXHEP),IDHEP(NMXHEP), JMOHEP(2,NMXHEP),JDAHEP(2,NMXHEP), PHEP(5,NMXHEP),VHEP(4,NMXHEP)
NEVHEP	event number
NHEP	number of entries in this event record
ISTHEP(..)	status code
IDHEP(..)	particle ID, P.D.G. standard
JMOHEP(1,..)	position of mother particle in list
JMOHEP(2,..)	position of second mother particle in list
JDAHEP(1,..)	position of first daughter in list
JDAHEP(2,..)	position of last daughter in list
PHEP(1,..)	x momentum in GeV/c
PHEP(2,..)	y momentum in GeV/c
PHEP(3,..)	z momentum in GeV/c
PHEP(4,..)	energy in GeV
PHEP(5,..)	mass in GeV/c**2
VHEP(1,..)	x vertex position in mm
VHEP(2,..)	y vertex position in mm
VHEP(3,..)	z vertex position in mm
VHEP(4,..)	production time in mm/c

ISTHEP convention	
0	null
1	final state particle
2	intermediate state
3	documentation line
4-10	reserved for future use
11-200	reserved for specific model use
201-...	reserved for users

Event I/O is handled by the routines STDXWRT(ILBL,ISTR,LOK) and STDXRD(ILBL,ISTR,LOK). LOK is an integer variable which returns the status of the I/O attempt. ILBL is an integer which defines the type of record to read or write. ISTR is a label used by MCFio to identify the event file. STDXRD and STDXWRT are interfaces from StdHep to MCFio xdr binary files, which can be transported to other platforms. See the StdHep manual for a more complete explanation.

Supported Values of ILBL	
ILBL = 1	standard HEPEVT common block
ILBL = 2	standard HEPEVT common block plus multiple interaction info
ILBL = 100	stdhep begin run record
ILBL = 200	stdhep end run record
ILBL = anything else	use I/O buffer

StdHep writes the information in common block STDCM1 as the begin and end run records. You must call STDFLHWXSEC, STDFLISXSEC, or STDFLPYXSEC(N1) to fill common block STDCM1 before calling HEPWRT with ILBL of 100 or 200. N1 is the number of events to be generated.

STDCM1 Common Block	
COMMON /STDCM1/	STDECOM,STDXSEC,STDSEED1,STDSEED2,
1	NEVTREQ,NEVTGEN,NEVTWRT
STDECOM	center-of-mass energy
STDXSEC	cross-section
STDSEED1	random number seed
STDSEED2	random number seed
NEVTREQ	number of events to be generated
NEVTGEN	number of events actually generated
NEVTWRT	number of events written to output file

The routine STDXRDM(ILBL,ISTR,LOK) is provided to allow users the option of reading events from multiple input streams. STDXRDM adds each event to the end of the existing data in HEPEVT. The user must call STDZERO to zero the arrays in HEPEVT. This is intended as a tool to fake multiple interactions.

See $STDHEP_DIR/doc/stdhep.ps for a list of stdhep translation, I/O, and utility routines.