Fermilab

StdHep 3.01
PM0091

StdHep 3.01

Monte Carlo Standardization at FNAL

Lynn Garren

November 17, 1995

DISCLAIMER

The software is licensed on an "as is" basis only. Universities Research Association, Inc. (URA) makes no representations or warranties, express or implied. By way of example but not of limitation, URA makes no representations or WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, that the use of the licensed software will not infringe any patent, copyright, or trademark, or as to the use (or the results of the use) of the licensed software or written material in terms of correctness, accuracy, reliability, currentness or otherwise. The entire risk as to the results and performance of the licensed software is assumed by the user. Universities Research Association, Inc. and any of its trustees, overseers, directors, officers, employees, agents or contractors shall not be liable under any claim, charge, or demand, whether in contract, tort, criminal law, or otherwise, for any and all loss, cost, charge, claim, demand, fee, expense, or damage of every nature and kind arising out of, connected with, resulting from or sustained as a result of using this software. In no event shall URA be liable for special, direct, indirect or consequential damages, losses, costs, charges, claims, demands, fees or expenses of any nature or kind.

This material resulted from work developed under a Government Contract and is subject to the following license: The Government retains a paid-up, nonexclusive, irrevocable worldwide license to reproduce, prepare derivative works, perform publicly and display publicly by or for the Government, including the right to distribute to other Government contractors. Neither the United States nor the United States Department of Energy, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

For futher information about this program, contact Lynn Garren, Fermi National Accelerator Laboratory (FNAL), (708) 840-2061, garren@fnal.gov.

Contents

1 Introduction

A basic particle numbering scheme has been proposed by the Particle Data Group. This scheme has been agreed to by most people writing Monte Carlo packages. A standard output common block has also been agreed to. Because the P.D.G. convention allows some room for interpretation, implementation tends to have minor variations among packages. Routines have been written to convert Isajet, Jetset, Herwig, and QQ output to the standard, agreed-upon, format, as interpreted at FNAL. The StdHep library also contains various utility routines for use with standard format events.

2 Particle Numbering Scheme

The standard particle numbering scheme is explained in references [2] and [5]. The particle ID numbers between 1 and 99 are for elementary particles: quarks, gluons, leptons, gauge and higgs bosons and their supersymmetric partners. A complete list has not been agreed to for the non-standard particles. Since both Isajet and Jetset include different sets of non-standard particles, I have attempted to reconcile the list. Numbers from 1-40 are for normal elementary particles and numbers 41-80 are for their supersymmetric partners. Numbers 1-18 and 21-25 match the PDG recommendation, however there are too many unseen theoretical particles used between Isajet and Jetset to do more than just put them in the remaining slots. The list of elementary particles is in Appendix A.

The PDG numbering algorithm for composite particles allows a signed 5 digit number for each particle: ±INMKL. The first digit, L, contains the J-spin as L = 2J + 1. The fifth digit, I, is used to differentiate particles with the same quark content. The digits N, M, and K contain the quark content. Digits M and K are used for mesons, with N = 0. Digits N, M, and K are used for baryons. Digits N and M are used for diquarks, with K = 0. (A list of diquark states is in Appendix B.) 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. Jetset, Herwig, and QQ use the PDG algorithm in addition to internal compressed numbering schemes.

The Isajet particle identification algorithm uses a signed four digit number: ±MLKJ. M, L, and K are quarks and J is the spin. A negative number indicates the antiparticle, and is meant to associate with the lightest quark. For mesons, M = 0, and for diquarks, K = 0.

Jetset, Herwig, and QQ have some differences from the actual numbers listed in the 1994 Review of Particle Properties. Discrepancies arise because Monte Carlo generators include states not in the PDG list, which covers only known states. Since any reasonable Monte Carlo generator must include states not yet seen and should be fairly stable, I have extended the list of possible particles. Appendix C contains a full list of meson states and their ID numbers, up through the top quark states. Both Isajet and Jetset define fourth family states. Note that in cases where the meson numbers do not agree with the list in the 1994 Review of Particle Properties, only the fifth digit, differentiating particles with the same quark content, is changed.

With StdHep 2.02 and later, the confusion about numbering baryon Ξ and Ω states for charmed and heavier quarks has been resolved. Three spin 1/2 states are recognized for cxy, bxy, etc., where x and y are lighter, non-identical quarks. The non-primed states are antisymmetric under interchange of the lighter quarks. and the primed states are symmetric. This is the same as the Jetset and Herwig definitions. Also, the Isajet states were translated incorrectly. This has been fixed. Appendix D contains a full list of the baryon states.

An ad-hoc numbering scheme for ions has been added to StdHep 3.01. Ion numbers are FAAAZZZ000L. F = 1 and flags this as an ion. AAA, and ZZZ are the ion's A and Z respectively. The first digit, L, contains the J-spin as L = 2J + 1.

3 Event Structure

In principle, a standard Monte Carlo event structure has been agreed to. Using this standard means that you only need one interface to use events from many different generators. Since no one is anxious to rewrite their code to use the new structure, an interface routine is generally provided. Also, it should be noted that some generator-specific information is usually lost in translating the output to standard format. Therefore the recommended proceedure at this point is either to write both the generator-specific information and the standard information for each event or to make the translation to the standard common block when the events are input for further processing. In either case, the standard does not attempt to address initialization or final information, but does provide a flag for such "events".
The standard output common block has the following form:

```
      integer nmxhep
      parameter (nmxhep=4000)
      integer nevhep,nhep,isthep,idhep,jmohep,jdahep
      double precision phep,vhep
      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

Note that phep and vhep have been changed from real to double precision as of StdHep 3.00 and Jetset 7.0407. This is the emerging standard for LEP 2 [1].

Nevhep is -1 for initialization records and -2 for final records. Jetset does not use event numbers, so nevhep is initialized the first time you call the Jetset translation routine and incremented by one for each call thereafter. In order to keep Isajet information available, jmohep(2,..) contains the jet identification. Neither Herwig nor Isajet have any vertex information, so vhep is zero. The HEPEVT common block may be included with the include file stdhep.inc or stdhep.h from the $STDHEP_DIR/src/inc subdirectory.

The isthep convention is as follows:

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

I have defined isthep = 11, 12, and 13 as the initial jets, W jet, and pair jet information from the Isajet /PJETS/ common block. In addition, isthep = 21 or 22 for "final" or decaying partons from the Isajet /JETSET/ common block.

4 Doing the Translation

To convert Monte Carlo events to standard output, call the routines lunhep(mconv), hwghep(mconv), isahep(mconv), or qqhep(mconv), where mconv = 1 converts from Isajet, Jetset, Herwig, or QQ event format to standard output format and mconv = 2 converts standard output format to Isajet, Jetset, Herwig, or QQ event format. Please note that some information is lost if you convert back to generator format.

Doug Wright of LLNL has supplied hep2geant, which takes undecayed HEPEVT particles and puts them on the Geant KINE stack. Vertex positions are recorded properly via calls to GSVERT. The Geant particles in the stack are cross referenced to the line numbers in StdHep through the user words in GSKINE. HEPEVT particles unknown to Geant are entered as geantino's and a warning message is printed.

The particle ID translation is done by the functions gtran(id,mconv) for Geant, hwtran(id,mconv) for Herwig, istran(id,mconv) for Isajet, lutran(id,mconv) for Jetset, and qqtran(id,mconv) for QQ. Id is the particle identification number to be translated to or from StdHep.

5 QQ interface

QQ currently contains the best known routines for decaying b particles. Therefore, a set of routines have been written to allow users to decay particles with QQ. In general, you should call stdqquset to initialize QQ and stddecayqq for each event. Stddecayqq will search the HEPEVT list for particles which QQ recognizes and decay them one by one, adding the results to HEPEVT. Optionally, you may wish to call stdqqdcy(it) to decay only particle it via QQ.

The QQ particles and their decay properties are defined in $QQ_DIR/decay.dec. To redefine any of these decays or particle properties, provide a user decay file, which is read after decay.dec. Define the environmental variable QQ_USER_FILE to point to your user decay file.

See the source code in $QQ_DIR/example for examples of using QQ with Isajet, Pythia, or Herwig. It is necessary to turn off appropriate decays in the generator if QQ is to be called.

6 Particle Properties

The Particle Data Group provides a table of particle masses and widths for known particles. This table is distributed with the StdHep package, and is defined as either logical name or environmental variable PDG_MASS_TBL. Two routines are supplied to read (pdgrdtb) or print (pdgprtb) the contents of the table. The table can also be read directly, since it is in ascii format. Because this table only contains known particles, it is of limited use. The user should check the mass and width information in the context of the Monte Carlo generator being used. Pdgrdtb uses lnhdcy from the HEPLUN common block as the unit number for the PDG table. The file is opened and closed within pdgrdtb. PDG_MASS_TBL points to the PDG table and may be redefined by the user.
The routine pdgrdtb fills the STDTBL common block:

```
      integer nmxln
      parameter (nmxln=1600)
      integer idt
      real stmerr,stwerr
      double precision stmass,stwidth
      character*21 stname
      common/stdtbl/ idt(nmxln),stmerr(2,nmxln),stwerr(2,nmxln),
     1               stmass(nmxln),stwidth(nmxln),stname(nmxln)
```

idt(..)	-	particle ID, P.D.G. standard
stmass(..)	-	particle mass in GeV/c^2
stmerr(1,..)	-	positive error on particle mass in GeV/c^2
stmerr(2,..)	-	negative error on particle mass in GeV/c^2
stwidth(..)	-	particle width in GeV/c^2
stwerr(1,..)	-	positive error on particle width in GeV/c^2
stwerr(2,..)	-	negative error on particle width in GeV/c^2
stname(..)	-	particle name

The routine pdgprtb(i,lun) will print either the contents of STDTBL (i=1) or the complete list of particles supplied by the Particle Data Group (i=2). The complete list contains particles like N(1650), which are not defined by StdHep. Lun is the unit number of the output file for the list.

7 Utility Routines

If you wish to print an event from the standard common block, call heplst(mlst), where mlst = 1 for output without vertex information and mlst = 2 for output with vertex information. If heplst is called with mlst equal to anything other than 1 or 2, it changes mlst to 1.

Three different event I/O options are provided: zebra, native mode binary, and machine independent binary (xdr). All I/O routines use lok. The native mode binary and xdr routines also use ilbl. 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. It is expected that the serious user will include the generator as part of a much larger Monte Carlo, and therefore write his/her own I/O routines.

Stdxrd(ilbl,istream,lok) and stdxwrt(ilbl,istream,lok) are interfaces to the xdr I/O protocol, which is machine independent. Istream is used to keep track of the xdr output stream. It is the index for array ixdrstr(15), which is filled by stdxrinit or stdxwinit. The xdr implementation used is MCFio, which is included as part of the StdHep release. The routine stdxrinit(filename,ntries,istream,lok) or stdwrinit(filename,title,ntries,istream,lok) must be called to initialize the xdr stream and open the file, filename. Ntries is the expected number of events. Filename and title are character variables. There is also a C binding for this interface. The C routines are in libstdhepC.a.

```
      call stdxwinit('stdtstpx.io','StdHep/Pythia example',
     1               nevt,istr,lok) ! open output stream
      call stdflpyxsec(nevt) ! get run information from Pythia
      call stdxwrt(100,istr,lok) ! write begin run record
      ....
      do i=1,nevt
        call pyevnt ! generate Pythia event
        call lunhep(1) ! fill HEPEVT common block
        call stdxwrt(1,istr,lok) ! write event record
      enddo
      ....
      call stdflpyxsec(nevt) ! get run information from Pythia
      call stdxwrt(200,istr,lok) ! write end run record
      call stdxend(istr) ! close output stream
```

Heprd(ilbl,lok) and hepwrt(ilbl,lok) are simplistic routines which use a sequential, unformatted binary file, that cannot be transported to other platforms. Heprd writes a record consisting of ilbl and nbuf (buffer length). This is followed by an output record of length nbuf.

```
      open(unit=lnhwrt,file='stdtstp.io',status='new',
     1     form='unformatted') ! open output file
      call stdflpyxsec(nevt) ! get run information from Pythia
      call hepwrt(100,lok) ! write begin run record
      ....
      do i=1,nevt
        call pyevnt ! generate Pythia event
        call lunhep(1) ! fill HEPEVT common block
        call  hepwrt(1,lok) ! write event record
      enddo
      ....
```

```
call stdflpyxsec(nevt) ! get run information from Pythia
call hepwrt(200,lok) ! write end run record
close(unit=lnhwrt) ! close output file
```

Hepzrd and hepzwrt use the zebra structure on native mode FZ files. The routine hepzinit must be called once before using hepzwrt or hepzrd. You must include your own OPEN and FZFILE calls. It is recommended that you use the C I/O option for zebra on unix platforms. The routines hpzflb and hpzflc, called by hepzwrt and hepzrd, respectively, are provided as an option if you want to call FZOUT and FZIN yourself.

```
call cfopen(iquest(1),0,900,'w',2,'stdtstjz.fz',ierr) ! open output file
call fzfile(lnhwrt,0,'XLO') ! initialize the output file for zebra
....
do i=1,nevt
  call lu2ent(0.,2,-2,20.) ! generate Jetset event
  call lunhep(1) ! fill HEPEVT common block
  call hepzwrt(lok) ! write event record
enddo
....
call fzendo(lnhwrt,'et') ! terminate zebra output
close(unit=lnhwrt) ! close output file
```

The supported values of ilbl are:

ilbl = 1	-	standard HEPEVT common block	
ilbl = 2	-	HEPEVT and HEPEV2 common blocks	
ilbl = 100	-	StdHep begin run record	
ilbl = 101	-	Pythia begin run record	(heprd/hepwrt only)
ilbl = 102	-	Jetset begin run record	(heprd/hepwrt only)
ilbl = 103	-	Herwig begin run record	(heprd/hepwrt only)
ilbl = 104	-	Isajet begin run record	(heprd/hepwrt only)
ilbl = 105	-	QQ begin run record	(heprd/hepwrt only)
ilbl = 200	-	StdHep end run record	
ilbl = 201	-	Pythia end run record	(not implemented)
ilbl = 202	-	Jetset end run record	(not implemented)
ilbl = 203	-	Herwig end run record	(not implemented)
ilbl = 204	-	Isajet end run record	(heprd/hepwrt only)
ilbl = 205	-	QQ end run record	(not implemented)
ilbl = anything else	-	use I/O buffer	(heprd/hepwrt only)

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. StdHep writes the information in common block STDCM1 as the begin and end run records. Common block STDCM1 may be included as include file stdcm1.inc or stdcm1.h.

The routines heprdm(ilbl,lok,iun) and stdxrdm(ilbl,istream,lok) are provided to allow users the option of reading events from multiple input streams. Iun is the index to the unit number array lnhrdm(15). Heprdm and stdxrdm add each event to the end of the existing data in HEPEVT. The user must call stdzero to initialize the arrays in HEPEVT. This is intended as a tool to fake multiple interactions.

The logical unit numbers used by StdHep are found in the include file stdlun.inc (or stdlun.h):

```
integer lnhwrt,lnhrd,lnhout,lnhdcy,lnhrdm,ixdrstr
common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy,lnhrdm(15)
common/stdstr/ixdrstr(15)
```

lnhwrt	-	output unit for standard format events
lnhrd	-	input unit for standard format events
lnhout	-	lineprinter output unit
lnhdcy	-	input unit for standard decay table (not yet implemented)
lnhrdm	-	array of input unit numbers for reading multiple files
ixdrstr	-	array of input streams for use by MCFio

The routine, isaprtlst(jtlis), is no longer necessary and is declared obsolete. Use the Isajet routine prtlst instead.

Many new routines have been created to interrogate the standard event common block. Routines exist to return lists of descendants, lists of anncestors, lists of particles containing a specific quark, etc. Section 10 contains a list of the StdHep routines. More extensive details on the StdHep routines can be found in Section 10 or in $STDHEP_DIR/doc/stdhep.doc.

8 Linking

The routines are in the StdHep library and may be included by linking STDHEP_OLB on the vax machines or $STDHEP_DIR/lib/libstdhep.a on unix machines after setting up stdhep. Jetset provides a similar translation routine, luhepc, which does not give identical results. A version of the Isajet translation routine has been included in the Isajet library by Frank Paige. Be sure to link StdHep before Isajet if you want the routines described in this document. The StdHep test routines, in the example subdirectory, provide working examples.

To use the xdr I/O interface, link with libFmcfio.a (disk) or libFmcfioRbio.a (tape). If you use the C interface routines for MCFio, link with libstdhepC.a. The MCFio documentation is included in $STDHEP_DIR/doc.

9 Event Display

A motif-based event display has been written by Paul Lebrun, using the Nirvana spin widget. Aliases Space and Phase are defined upon setup of StdHep to point to StdHep Space Display and StdHep Phase Display, respectively. StdHep Space Display is a tool to select, display, and browse tracks and vertices from the HEPEVT common block, in order to understand the kinematics, spatial relationships among these vertices, and the decay chain. StdHep Phase Display is a tool to select, display, and browse StdHep events, in order to understand the kinematics, topology, and the decay chain. PhaseD can be invoked from Space, but Space cannot be invoked from Phase.

After invoking Space or Phase, open a file containing StdHep events by means of the File pull-down menu.

After specifying the file to browse, specify the event to view. Use the arrow buttons on the top of the panel to select the event you wish to see. At the outset, the first event in the file is shown. The length of the track is mapped to the norm of the 3-momentum vector. Sliders are available to manipulate the event. Note that Space has a transverse magnification factor since most, if not all, event generators have the beam axis along the z (longitudinal) direction. The color is mapped to the particle identification. The Color Code is available in the Event pull-down menu.

To relate the kinematics to the decay chain in either Space or Phase, select the Event Tree option under the Event pull-down menu. Help on node or track selection in this display is available in the Help pull down menu. Further information about tracks or nodes in the tree is available. In fact, all the information in the HEPEVT common block can be accessed by interacting with this display.

You can pick a selected track on the drawing area by pointing the cursor to the middle of the track segment and depressing the central button (the left button controls the motion of the display). The selected track (or a portion of it) is highlighted on the display, and a small text box should appear, indicating the particle identity, its place in the HEPEVT common block, and various kinematic quantities. If the Event Tree is also displayed, the node to which the tracks belong is highlighted, indicating the position of the selected particle in the hierarchy. You can also get more information about a given node in the tree by pressing the left mouse button on a tree element. The corresponding tracks are highlighted on the event display. The highlight may not work properly if the main panel is resized. If this occurs, resize the window to readjust graphical context.

10 StdHep Routines

10.1 StdHep Translation Routines

hep2geant Put undecayed HEPEVT particles into the Geant KINE stack. Particles unknown to GEANT are entered as geantino's.

hwghep(mconv) Translate to or from Herwig event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Herwig common blocks

isahep(mconv) Translate to or from Isajet event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Isajet common blocks

lunhep(mconv) Translate to or from Jetset event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Jetset common blocks

qqhep(mconv) Translate to or from QQ event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the QQ common blocks
	=	3	fill the QQ common blocks for decay of "stable" particles

pdgrdtb Read particle mass and width information from the PDG table. Fill the PDGTBL and STDTBL common blocks.

stdflhwxsec Fill common block STDCM1 from Herwig. STDCM1 contains center-of-mass energy, cross-section, random number seeds, number of events to be generated, number of events generated, and number of events written.

stdflisxsec Fill common block STDCM1 from Isajet. STDCM1 contains center-of-mass energy, cross-section, random number seeds, number of events to be generated, number of events generated, and number of events written.

stdflpyxsec(n1) Fill common block STDCM1 from Pythia. STDCM1 contains center-of-mass energy, cross-section, random number seeds, number of events to be generated, number of events generated, and number of events written.

n1	=	number of events to be generated

hptrlsth Print full translation list for Herwig.

hptrlsti Print full translation list for Isajet.

hptrlstj Print full translation list for Jetset.

hptrlstq Print full translation list for QQ.

isaprtlst(jtlis) OBSOLETE - replaced by Isajet PRTLST routine.

pdgprtb(ityp,lun) Print information from the PDG mass and width table.

ityp	=	1	print particle properties for particles known to StdHep
	=	2	print complete list of particle properties from PDG table
lun	=		output unit for writing table

gtran(id,mconv) Translate to or from Geant particle id numbering.

gtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Geant to StdHep numbering scheme
	=	2	convert from StdHep to Geant numbering scheme

istran(id,mconv) Translate to or from Isajet particle id numbering.

istran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Isajet to StdHep numbering scheme
	=	2	convert from StdHep to Isajet numbering scheme

lutran(id,mconv) Translate to or from Jetset particle id numbering.

lutran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Jetset to StdHep numbering scheme
	=	2	convert from StdHep to Jetset numbering scheme

hwtran(id,mconv) Translate to or from Herwig particle id numbering.

hwtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Herwig to StdHep numbering scheme
	=	2	convert from StdHep to Herwig numbering scheme

qqtran(id,mconv) Translate to or from QQ particle id numbering.

qqtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from QQ to StdHep numbering scheme
	=	2	convert from StdHep to QQ numbering scheme

pdgtran(id,mconv) Translate to or from PDG particle id numbering.

pdgtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG to StdHep numbering scheme
	=	2	convert from StdHep to PDG numbering scheme

10.2 StdHep Routines for QQ Decay

stddecayqq Search the HEPEVT list for particles which QQ can decay. One by one, decay these particles and add the results to the HEPEVT common block.

stdqqdcy(it) Decay particle it via QQ and add the results to the HEPEVT common block.

it = the index for this particle within HEPEVT

stdzeroqq Zero elements of QQ common blocks before performing the QQ decays. This routine is called by stddecayqq.

stdqqadd(ihep) Add tracks generated by QQ to the HEPEVT common block.

ihep = the first QQ particle corresponds to HEPEVT particle ihep

stdqquset Call QQ initialization routines. This routine must be called by the user at initialization time.

10.3 Basic StdHep Routines

heplst(mlst) List an event from the HEPEVT common block.

mlst	=	1	print all information from HEPEVT except the vertex
	=	2	print all information from HEPEVT, including the vertex

heptree Print an event in simple tree format.

hepnam(id,chau) Return the particle name as a character string.

chau	=	20 character particle name
id	=	particle ID

hepcmp(id) Compress the StdHep particle ID for use in mass and decay arrays of size 1600. Also check to be sure the given ID is defined.

hepcmp	=	compressed particle ID
id	=	StdHep particle ID

hepchg(id) Return an integer charge which is 3 times the charge of the particle.

hepchg	=	3 times the charge of the particle
id	=	StdHep particle ID

stdchg(id) Return the charge of particle id as a real number.

stdchg	=	the charge of the particle
id	=	StdHep particle ID

stdexpname(id,lsz) Return an expanded particle name for particle id. Fill and pad the variable with empty space if necessary.

stdexpname	=	expanded particle name
id	=	StdHep particle ID
lsz	=	length of the character variable

stdlst List the particles known to StdHep.

10.4 StdHep Utility Routines

stdversn Print the StdHep version number and date.

stdquarklst(iq,nqrk,lsize,lqrk) Search the HEPEVT common block for all mesons and baryons with quark content iq. Fill array lqrk(lsize).

iq	=	quark identification
nqrk	=	number of particles found
lsize	=	length of the array
lqrk	=	array of particle indices in HEPEVT

stdparentlst(ip,nprnt,lsize,lprnt) Search the HEPEVT common block for all ancestors of particle ip. Fill lprnt(lsize).

ip	=	index of particle in HEPEVT
nprnt	=	number of particles found
lsize	=	length of the array
lprnt	=	array of particle indices in HEPEVT

stddautrlst(ip,ndau,lsize,ldtr) Search the HEPEVT common block for the immediate daughters of particle ip. Fill ldtr(lsize).

ip	=	index of particle in HEPEVT
ndau	=	number of particles found
lsize	=	length of the array
ldtr	=	array of particle indices in HEPEVT

stddscndlst(ip,ndsc,lsize,ldsc) Search the HEPEVT common block for all the descendants of particle ip. Fill ldsc(lsize).

ip	=	index of particle in HEPEVT
ndsc	=	number of particles found
lsize	=	length of the array
ldsc	=	array of particle indices in HEPEVT

stdstdsclst(ip,ndsc,lsize,ldsc) Search the HEPEVT common block for all the stable descendants of particle ip. Fill ldsc(lsize).

ip	=	index of particle in HEPEVT
ndsc	=	number of particles found
lsize	=	length of the array
ldsc	=	array of particle indices in HEPEVT

stdchgdsclst(ip,ndsc,lsize,ldsc) Search the HEPEVT common block for all the charged, stable descendants of particle ip. Fill ldsc(lsize).

ip	=	index of particle in HEPEVT
ndsc	=	number of particles found
lsize	=	length of the array
ldsc	=	array of particle indices in HEPEVT

stdparent(ip,ipc,lyn) Check to see if particle ipc is a parent of particle ip.

ip	=	index of particle in HEPEVT
ipc	=	index of prospective parent in HEPEVT
lyn	=	true or false

stddautr(ip,ipc,lyn) Check to see if particle ipc is a descendant of particle ip.

ip	=	index of particle in HEPEVT
ipc	=	index of prospective descendant in HEPEVT
lyn	=	true or false

stdquarkcnt(ip,iq,iret,ipar) Look to see if particle it or any of its parents contain quark iq.

ip	=	index of particle in HEPEVT	
iq	=	quark identification	
iret	=	0	no match found
	=	1	this particle contains quark iq
	=	2	parent ipar contains quark iq
ipar	=	0	unless iret is 2

stdrotboost(the,phi,bex,bey,bez) Rotate and boost the frame of reference for particles in the HEPEVT common block. This is a modified version of LUROBO from Jetset.

the	=	polar coordinate giving the rotated direction
phi	=	polar coordinate giving the rotated direction
bex	=	give direction and size of a Lorentz boost
bey	=	give direction and size of a Lorentz boost
bez	=	give direction and size of a Lorentz boost

stddbrotb(imi,ima,the,phi,dbex,dbey,dbez) Rotate and boost the frame of reference for the HEPEVT common block. Stddbrotb performs a specific range and double precision boost. This is a modified version of LUDBRB from Jetset.

imi	=	begin range of entries affectec by transformation
ima	=	end range of entries affectec by transformation
the	=	polar coordinate giving the rotated direction
phi	=	polar coordinate giving the rotated direction
dbex	=	give direction and size of a Lorentz boost
dbey	=	give direction and size of a Lorentz boost
dbez	=	give direction and size of a Lorentz boost

stdsort Sort the HEPEVT common block by daughters. This routine is called by qqhep.

stdquarks(ip,kq1,kq2,kq3,kqx,kqj) Get the constitutent quarks of particle ip. All numbers are positive.

ip	=	index of particle in HEPEVT
kq1	=	quark L of IJKLM
kq2	=	quark K of IJKLM
kq3	=	quark J of IJKLM
kqx	=	differentiates otherwise identically named particles
kqj	=	2*Jspin + 1

stdcquarks(ip,kq1,kq2,kq3,kqx,kqj) Get the signed constitutent quarks of particle ip.

ip	=	index of particle in HEPEVT
kq1	=	quark L of IJKLM
kq2	=	quark K of IJKLM
kq3	=	quark J of IJKLM
kqx	=	differentiates otherwise identically named particles
kqj	=	2*Jspin + 1

stdspin(ip,xjsp) Return the J spin of particle ip as a real number.

ip	=	index of particle in HEPEVT
xjsp	=	J spin

10.5 StdHep I/O Routines

10.5.1 Xdr Binary Files

stdxrd(ilbl,istream,lok) Read the next record from an xdr input stream, returning ilbl and lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxrdm(ilbl,istream,lok) Read the next record from an xdr input stream, returning ilbl and lok. Stdxrdm handles multiple input streams to fake multiple interactions.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxwrt(ilbl,istream,lok) Write record type ilbl to an xdr output stream, returning lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxrinit(filename,ntries,istream,lok) Initialize xdr and open an input stream, returning ntries and lok. This routine should never be called more than once.

filename	=		name of the xdr input file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxropen(filename,ntries,istream,lok) Open an xdr input stream, returning ntries and lok. This routine is called by stdxrinit.

filename	=		name of the xdr input file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxwinit(filename,title,ntries,istream,lok) Initialize xdr and open an output stream, returning lok. This routine should never be called more than once.

filename	=		name of the xdr output file
title	=		title for the xdr output file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxwopen(filename,title,ntries,istream,lok) Open an xdr output stream, returning lok. This routine is called by stdxwinit.

filename	=		name of the xdr output file
title	=		title for the xdr output file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxend(istream) Close the xdr output stream.

istream	=	index to array ixdrstr holding stream pointer

10.5.2 Native Mode Binary Files

heprd(ilbl,lok) Read the next record from a native mode binary file, returning ilbl and lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	101	Pythia begin run record
ilbl	=	102	Jetset begin run record
ilbl	=	103	Herwig begin run record
ilbl	=	104	Isajet begin run record
ilbl	=	105	QQ begin run record
ilbl	=	200	StdHep end run record
ilbl	=	204	Isajet end run record
lok	=	0	if no problems were encountered

heprdm(ilbl,lok,iun) Read the next record from a native mode binary file, returning ilbl and lok. Heprdm handles multiple input streams to fake multiple interactions.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	101	Pythia begin run record
ilbl	=	102	Jetset begin run record
ilbl	=	103	Herwig begin run record
ilbl	=	104	Isajet begin run record
ilbl	=	105	QQ begin run record
ilbl	=	200	StdHep end run record
ilbl	=	204	Isajet end run record
lok	=	0	if no problems were encountered
iun	=		index to lnhrdm to specify which input file to use

hepwrt(ilbl,lok) write out a record Write the specified record to a native mode binary file, returning lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	101	Pythia begin run record
ilbl	=	102	Jetset begin run record
ilbl	=	103	Herwig begin run record
ilbl	=	104	Isajet begin run record
ilbl	=	105	QQ begin run record
ilbl	=	200	StdHep end run record
ilbl	=	204	Isajet end run record
lok	=	0	if no problems were encountered

10.5.3 Zebra Files

hepzrd(lok) Read in an event from a zebra FZ file, returning lok.

lok	=	0	if no problems were encountered

hepzwrt(lok) Write an event to a zebra FZ file, returning lok.

lok	=	0	if no problems were encountered

hepzinit Initialize the zebra data structures.

hpzflb Create and fill the zebra data banks.

hpzflc Fill the HEPEVT common block from the zebra data banks.

10.5.4 I/O Utilities

stdzero Set the contents of the HEPEVT common block to zero. This routine must be called by the user if heprdm or stdxrdm is used.

10.6 StdHep I/O Routines - C Binding

10.6.1 Xdr Binary Files

int StdHepXdrRead(ilbl,istream) Read the next record from an xdr input stream, returning ilbl. The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

int StdHepXdrReadMulti(ilbl,istream) Read the next record from an xdr input stream, returning ilbl. StdHepXdrReadMulti handles multiple input streams to fake multiple interactions. The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

int StdHepXdrWrite(ilbl,istream) Write record type ilbl to an xdr output stream. The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

int StdHepXdrReadInit(filename,ntries,istream) initialize input Initialize xdr and open an input stream, returning ntries. The return argument is 0 if no problems were encountered. This routine should never be called more than once.

filename	=	name of the xdr input file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

int StdHepXdrReadOpen(filename,ntries,istream) initialize input stream Open an xdr input stream, returning ntries. This routine is called by stdxrinit. The return argument is 0 if no problems were encountered.

filename	=	name of the xdr input file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

int StdHepXdrWriteInit(filename,title,ntries,istream) Initialize xdr and open an output stream. The return argument is 0 if no problems were encountered. This routine should never be called more than once.

filename	=	name of the xdr output file
title	=	title for the xdr output file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

int StdHepXdrWriteOpen(filename,title,ntries,istream) Open an xdr output stream. This routine is called by stdxwinit. The return argument is 0 if no problems were encountered.

filename	=	name of the xdr output file
title	=	title for the xdr output file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

StdHepXdrEnd(istream) Close the xdr output stream at ixdrstr(istream).

istream = index to array ixdrstr holding stream pointer

10.6.2 I/O Utilities

StdHepZero Set the contents of the hepevt_ structure to zero. Hepevt_ is a mirror of the HEPEVT common block. This routine must be called by the user if StdHepXdrReadMulti is used.

11 History

3.01
- StdHep 3.01 was built with Jetset 7.0408, Herwig 5.8, Isajet 7.13, and QQ 9.1.
- A C binding library for the MCFio interface routines was added.
- Phase and Space display programs now accept MCFio files.
- An ad-hoc ion numbering scheme has been added.

3.00
- StdHep 3.00 was built with Jetset 7.04, Herwig 5.8, Isajet 7.13, and QQ 9.1.
- The phep and vhep arrays are now double precision. This provides consistency with Jetset 7.0407 (and higher) and Herwig. It is the emerging standard [1].
- MCFio I/O routines have been included.
- Translation done for QQ 9.1.
- Fix a bug in heprd/hepwrt (affects backward compatibility).
- As of PDG 1994 [2], the particle $f_0(1710)$ has become $f_J(1710)$. It retains particle ID 60221 in StdHep.

2.03
- StdHep 2.03 was built with Jetset 7.04, Herwig 5.8, Isajet 7.13, and QQ 9.0.

2.02
- StdHep 2.02 was built with Jetset 7.04, Herwig 5.7, Isajet 7.06, and QQ 9.0.
- StdHep 2.02 is not backwards compatible. It contains modifications to heprd and hepwrt and to the definitions of heavy baryons.
- Modify hep2geant, stdquarks, and stdspin
- Stdrotboost and stdcquarks added
- Allow more than one event to be read into the HEPEVT common block to mimic multiple interactions.
- Cross-section written upon request
- Add Pythia example
- Fix a minor bug in istran, which translated states not known to Isajet.

2.01
- Modify stddecayqq to spawn another routine
- Clean up begin-run and end-run output options

- First attempt to get cross-sections, etc. in a standard common block
- Expand read/write options
- Resolve Ξ and Ξ' translation issue
- Add more utility routines
- QQ translation routines
- read/write routine changes
- Version 2.01 is not backwards compatible.

2.00

- QQ translation and utility routines are now functional.
- Add routine to print StdHep version number.
- Add more diffractive particles for Jetset 7.4.
- Replace local routines isflavr and islabl with Isajet routines.
- Replace local routine isaprtlst with Isajet routine prtlst.
- Add utility routines
- Version 2.00 is not backwards compatible.

1.06

- Add utility routines
- Modify native mode output routines to use keyword for identification
- Add QQ translation routines
- Add Z'^0 for Herwig 5.7
- Use Herwig include file

1.05

- Make StdHep compatible with new SUSY particles in Isajet 7.0. (see Appendix A).
- Add hptrlst subroutines to stdhep library.
- Add routines to read and print the particle info supplied by PDG.

1.04

- As of PDG 1992 [3], the particle $f_2(1720)$ has become $f_0(1710)$. The particle ID was changed from 10225 to 60221.
- Add new Jetset particles

1.03

- Add zebra I/O routines
- Add flag HEPDBG for extra debugging code
- Add flag HPZTST to select zebra output in test jobs

1.02

- Add simple I/O routines to write HEPEVT common block.
- Add test programs
- Add routines to convert Herwig events.
- Improve hepnam and hepcmp

1.01 • First version of StdHep translation package.

• Convert Isajet and Jetset events to the HEPEVT standard.

References

[1] T. Sjostrand, "Interfacing four-fermion generators with QCD generators", Workshop on Physics at LEP2, (Jan. to Oct. 1995).

[2] Particle Data Group, L. Montanet, et al., Phys. Rev. D50, 1325-1326, (1994).

[3] Particle Data Group, K. Hikasa, et al., Phys. Rev. D45, III.73-III.74, (1992).

[4] B. Bambah et al., "QCD Generators", CERN-TH.5466/89, in "Z Physics at LEP 1", (1989).

[5] T. G. Trippe and G. R. Lynch, "Particle I.D. Numbers, Decay Tables, and Other Possible Contributions of the Particle Data Group to Monte Carlo Standards", LBL-24287, in "Proceedings of the Workshop on Detector Simulation for the SSC", (August 1987).

A Elementary Particle Identification Code Listing

Quarks and Leptons						
Particle	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
d	2	1	1	-2,-8	1	1
u	1	2	2	-1,-7	2	2
s	3	3	3	-3,-9	3	3
c	4	4	4	-4,-10	4	4
b	5	5	5	-5,-11	5	5
t	6	6	6	-6,-12	6	6
l	7 (y)	7	7 ($b\prime$)			7
h	8 (x)	8	8 ($t\prime$)			8
e^-	12	11	11	7,8	11	11
ν_e	11	12	12	9,10	12	12
μ^-	14	13	13	11,12	13	13
ν_μ	13	14	14	13,14	14	14
τ^-	16	15	15	15, 16	15	15
ν_τ	15	16	16	17,18	16	16
χ^-		17				17
ν_χ		18				18
reggeon		28				19
pomeron		29				20

Gauge and Higgs Bosons						
Particle	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
g	9	21	21	-13	21 (9)	21 (9)
γ	10	22	22	1	22	22
Z^0	90	23	23	2	23	23
W^+	80	24	24	3,4	24	24
H_1^0	81 (H^0)	25 (H^0)	25		25	25
η^0_{tech}		38				27
LQ^0		39				28
R^0		40				29
Z_2^0		32 (Z'^0)	32 (Z'^0)			30
W_2^+		34 (W'^+)				31
H_2^0	82 $(H_1^0)^{NOTE}$	35 (H'^0)			35 (H_2^0)	32
H_2^+	86 (H^+)	37 (H^+)			37 (H^+)	33
H_2^{++}	88 (H_1^{++})					34
Z_3^0		33 (Z''^0)				35
H_3^0	83 $(H_2^0)^{NOTE}$	36 (A^0)			36 (H_3^0)	36
H_3^+	87 (H_2^+)					37
H_3^{++}	89 (H_2^{++})					38
H_4^0	84 $(H_3^0)^{NOTE}$					39
H_5^0	85 (H_4^0)					40

NOTE: In Isajet 7.06, H_1^0, H_2^0, and H_3^0 are removed.

Supersymmetric Particles and Miscellaneous I						
Particle	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
left susy d	22					41
left susy u	21					42
left susy s	23					43
left susy c	24					44
left susy b	25					45
left susy t	26					46
gluino	29					47
photino						48
left susy e^-	32					51
left susy ν_e	31					52
left susy μ^-	34					53
left susy ν_μ	33					54
left susy τ^-	36					55
left susy ν_τ	35					56
H_L^0	82					57
H_H^0	83					58
H_A^0	84					59
right susy d	42					61
right susy u	41					62
right susy s	43					63
right susy c	44					64
right susy b	45					65
right susy t	46					66

Supersymmetric Particles and Miscellaneous II						
Particle	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
zino	30 (Z_1^{ss})					67
zino_2	40 (Z_2^{ss})					68
zino_3	50 (Z_3^{ss})					69
zino_4	60 (Z_4^{ss})					70
right susy e^-	52					71
right susy ν_e	51					72
right susy μ^-	54					73
right susy ν_μ	53					74
right susy τ^-	56					75
right susy ν_τ	55					76
wino	39 (W_1^{+ss})					77
wino_2	49 (W_2^{+ss})					78
internal code		81-99	91,94	82-93	81-99	81-99
ρ^0_{diffr}		110^{NOTE}				110
π^+_{diffr}		210				210
ω^0_{diffr}		220				220
ϕ^0_{diffr}		330				330
ψ^0_{diffr}		440				440
n^0_{diffr}		2110				2110
p^+_{diffr}		2210				2210
remnant photon			9998			9998
remnant nucleon			9999			9999
K^0_S	20	310	310	57	310	310
K^0_L	-20	130	130	58	130	130
NOTE: In Jetset 7.04, π^0_{diffr} becomes ρ^0_{diffr}.						

B List of diquark states

Diquarks						
Diquark	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
dd_0	2200					1101
ud_0	1200	2101	2101		2101	2101
uu_0	1100					2201
sd_0	2300	3101	3101		3101	3101
su_0	1300	3201	3201		3201	3201
ss_0	3300					3301
cd_0	-2400	4101				4101
cu_0	-1400	4201				4201
cs_0	-3400	4301				4301
cc_0	4400					4401
bd_0	2500	5101				5101
bu_0	1500	5201				5201
bs_0	3500	5301				5301
bc_0	4500	5401				5401
bb_0	5500					5501
td_0	-2600	6101				6101
tu_0	-1600	6201				6201
ts_0	-3600	6301				6301
tc_0	4600	6401				6401
tb_0	-5600	6501				6501
tt_0	6600					6601

Diquarks						
Diquark	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
dd_1		1103	1103		1103	1103
ud_1		2103			2103	2103
uu_1		2203	2203		2203	2203
sd_1		3103			3103	3103
su_1		3203			3203	3203
ss_1		3303	3303			3303
cd_1		4103				4103
cu_1		4203				4203
cs_1		4303				4303
cc_1		4403				4403
bd_1		5103				5103
bu_1		5203				5203
bs_1		5303				5303
bc_1		5403				5403
bb_1		5503				5503
td_1		6103				6103
tu_1		6203				6203
ts_1		6303				6303
tc_1		6403				6403
tb_1		6503				6503
tt_1		6603				6603

C Complete Meson Particle Identification Code Listing

Light Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
ρ^0_{diffr}		110^{NOTE}				110
π^+_{diffr}		210				210
π^0	110	111	111	51	111	111
π^+	120	211	211	21,22	211	211
$a_0(980)^0$		10111		251	10111	10111
$a_0(980)^+$		10211		252,253	10211	10211
$\pi(1300)^0$					20111	20111
$\pi(1300)^+$					20211	20211
$\rho(770)^0$	111	113	113	91	113	113
$\rho(770)^+$	121	213	213	61, 62	213	213
$b_1(1235)^0$		10113		265	10113	10113
$b_1(1235)^+$		10213		266,267	10213	10213
$a_1(1260)^0$		20113	20113 (A_1^0)	108	20113	20113
$a_1(1260)^+$		20213	20213 (A_1^+)	107,109	20213	20213
$\rho(1450)^0$					40113	30113
$\rho(1450)^+$					40213	30213
$\rho(1700)^0$					30113	40113
$\rho(1700)^+$				205,206	30213	40213
$a_2(1320)^0$		115	115 (A_2^0)	254	115	115
$a_2(1320)^+$		215	215 (A_2^+)	255,256	215	215
$\pi_2(1670)^0$					10115	10115
$\pi_2(1670)^+$					10215	10215
$\rho_3(1690)^0$					117	117
$\rho_3(1690)^+$					217	217

NOTE: In Jetset 7.04, π^0_{diffr} becomes ρ^0_{diffr}.

Light Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
ω^0_{diffr}		220				220
η	220	221	221 (η & η_s)	52	221	221
$f_0(975)$		10221		257	10221	10221
$\eta(1295)$					20221	20221
$f_0(1400)$				258	30221	30221
$\eta(1440)$					40221	40221
$f_0(1590)$					50221	50221
$f_J(1710)^{NOTE}$					60221	60221
$\omega(783)$	221	223	223	92	223	223
$h_1(1170)$		10223		263	10223	10223
$f_1(1285)$		20223	20223 (D^0)	259	20223	20223
$\omega(1420)$					50223	30223
$f_1(1420)$				260	30223	40223
$f_1(1510)$					40223	50223
$\omega(1600)$					60223	60223
$f_2(1270)$		225	225 (f^0)	261	225	225
$f_2(2010)$					20225	20225
$f_2(2300)$					30225	30225
$f_2(2340)$					40225	40225
$\omega_3(1670)$					227	227
$f_4(2050)$					229	229

NOTE: In PDG 1992 [3], $f_2(1720)$, which was 10225, became $f_0(1710)$.
In PDG 1994 [2], $f_0(1710)$ became $f_J(1710)$.

Strange Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
K_L^0	-20	130	130	58	130	130
K_S^0	20	310	310	57	310	310
K^0	230	311	311	25,26	311	311
K^+	130	321	321	23,24	321	321
$K_0^*(1430)^0$		10311		268,269	10311	10311
$K_0^*(1430)^+$		10321		270,271	10321	10321
$K^*(892)^0$	231	313	313	65,66	313	313
$K^*(892)^+$	131	323	323	63,64	323	323
$K_1(1270)^0$		10313		59,60	10313	10313
$K_1(1270)^+$		10323		201,202	10323	10323
$K_1(1400)^0$		20313	20313 (Q_A^0)		20313	20313
$K_1(1400)^+$		20323	20323 (Q_A^+)		20323	20323
$K^*(1410)^0$				19,20	30313	30313
$K^*(1410)^+$				203,204	30323	30323
$K^*(1680)^0$					40313	40313
$K^*(1680)^+$					40323	40323
$K_2^*(1430)^0$		315	315 (K^{**0})	272,273	315	315
$K_2^*(1430)^+$		325	325 (K^{**+})	274,275	325	325
$K_2(1770)^0$					10315	10315
$K_2(1770)^+$					10325	10325
$K_2(1820)^0$					20315	20315
$K_2(1820)^+$					20325	20325
$K_3^*(1780)^0$				278,279	317	317
$K_3^*(1780)^+$				280,281	327	327
$K_4^*(2045)^0$					319	319
$K_4^*(2045)^+$					329	329

Strange Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
ϕ^0_{diffr}		330				330
$\eta'(958)$	330	331	331 (η'_s & η'_r)	53	331	331
f'_0		10331				10331
$\phi(1020)$	331	333	333	93	333	333
$h_1(1380)$		10333 (h'_1)		264 (h'_1)		10333
$f_1(1420)$			20333 (E^0)			20333
$\phi(1680)$					10333	30333
f'_1		20333				40333
$f'_2(1525)$		335	335 (f'^0_r)	262	335	335
$\phi_3(1850)$					337	337

Charmed Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
D^+	-240	411	411	29,30	411	411
D^0	-140	421	421	27,28	421	421
D_0^{*+}		10411		221,225		10411
D_0^{*0}		10421		229,233		10421
$D^*(2010)^+$	-241	413	413	69,70	413	413
$D^*(2007)^0$	-141	423	423	67,68	423	423
$D_1(2420)^+$		10413	20413 (D_A^+)	223,227		10413
$D_1(2420)^0$		10423	20423 (D_A^0)	231,235	10423	10423
D_1^{*+}		20413		222,226		20413
D_1^{*0}		20423		230,234		20423
$D_2^*(2460)^+$		415	415 (D^{**+})	224,228	415	415
$D_2^*(2460)^0$		425	425 (D^{**0})	232,236	425	425
D_s^+	-340	431	431 (F^+)	31,32	431	431
D_{0s}^{*+}		10431		237,241		10431
D_s^{*+}	-341	433	433 (F^{*+})	71,72	433	433
$D_{1s}(2536)^+$		10433	20433 (F_A^+)	239,243	10433	10433
D_{1s}^{*+}		20433		238,242		20433
D_{2s}^{*+}		435	435 (F^{**+})	240,244		435

Charmed Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
ψ^0_{diffr}		440				440
$\eta_c(1S)$	440	441	441	54	441	441
$\chi_{c0}(1P)$		10441	10441 (χ)	110	10441	10441
$\eta_c(2S)$						20441
J/ψ	441	443	443	94	443	443
$h_c(1P)$		10443		215		10443
$\chi_{c1}(1P)$		20443	10441 (χ)	111	10443	20443
$\psi(2S)$		30443 (ψ')	20443 (ψ')	219	20443	30443
$\psi(3770)$			30443 (ψ'')	211	30443	40443
$\psi(4040)$				212	40443	50443
$\psi(4160)$				213	50443	60443
$\psi(4415)$				214	60443	70443
$\chi_{c2}(1P)$		445	10441 (χ)	112	445	445

Bottom Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
B^0	250	511	511	36,35	511	511
B^+	150	521	521	34,33	521	521
B_0^{*0}		10511		479,483		10511
B_0^{*+}		10521		471,475		10521
B^{*0}	251	513		76,75	513	513
B^{*+}	151	523		74,73	523	523
B_1^0		10513		481,485		10513
B_1^+		10523		473,477		10523
B_1^{*0}		20513		480,484		20513
B_1^{*+}		20523		472,476		20523
B_2^{*0}		515		482,486		515
B_2^{*+}		525		474,478		525
B_s^0	350	531	531	38,37	531	531
B_{s0}^{*0}		10531		487,491		10531
B_s^{*0}	351	533		78,77		533
B_{s1}^0		10533		489,493		10533
B_{s1}^{*0}		20533		488,492		20533
B_{s2}^{*0}		535		490,494		535
B_c^+	450	541	541	40,39		541
B_{c0}^{*+}		10541				10541
B_c^{*+}	451	543		80,79		543
B_{c1}^+		10543				10543
B_{c1}^{*+}		20543				20543
B_{c2}^{*+}		545				545

Bottom Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
$\eta_b(1S)$	550	551		55		551
$\chi_{b0}(1P)$		10551		100	551	10551
$\eta_b(2S)$						20551
$\chi_{b0}(2P)$				103	10551	30551
$\eta_b(3S)$						40551
$\Upsilon(1S)$	551	553	553	95	553	553
$h_b(1P)$		10553 (h_{1b})		218		10553
$\chi_{b1}(1P)$		20553		101	10553	20553
$\Upsilon(2S)$		30553 (Υ')		97	20553	30553
$h_b(2P)$						40553
$\chi_{b1}(2P)$				104	70553	50553
$\Upsilon(3S)$				98	30553	60553
$\Upsilon(4S)$				99	40553	70553
$\Upsilon(10860)$				216	50553	80553
$\Upsilon(11020)$				217	60553	90553
$\chi_{b2}(1P)$		555		102	555	555
$\chi_{b2}(2P)$				105	10555	10555

Top Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
T^+	-260	611	611	43,44		611
T^0	-160	621	621	41,42		621
T_0^{*+}		10611				10611
T_0^{*0}		10621				10621
T^{*+}	-261	613		83,84		613
T^{*0}	-161	623		81,82		623
T_1^+		10613				10613
T_1^0		10623				10623
T_1^{*+}		20613				20613
T_1^{*0}		20623				20623
T_2^{*+}		615				615
T_2^{*0}		625				625
T_s^+	-360	631	631	45,46		631
T_{s0}^{*+}		10631				10631
T_s^{*+}	-361	633		85,86		633
T_{s1}^+		10633				10633
T_{s1}^{*+}		20633				20633
T_{s2}^{*+}		635				635
T_c^0	460	641	641	47,48		641
T_{c0}^{*0}		10641				10641
T_c^{*0}	461	643		87,88		643
T_{c1}^0		10643				10643
T_{c1}^{*0}		20643				20643
T_{c2}^{*0}		645				645

Top Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
T_b^+	-560	651	651	49,50		651
T_{b0}^{*+}		10651				10651
T_b^{*+}	-561	653		89,90		653
T_{b1}^+		10653				10653
T_{b1}^{*+}		20653				20653
T_{b2}^{*+}		655				655
η_t	660	661		56		661
χ_{t0}		10661				10661
θ	661	663	663 (T/ψ)	96 (ψ_t)		663
h_{1t}		10663				10663
χ_{1t}		20663				20663
χ_{2t}		665				665

D List of baryon states

Light Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
p^+_{diffr}			2210				2210
n^0_{diffr}			2110				2110
p	duu	1120	2212	2212	137,138	2212	2212
n	ddu	1220	2112	2112	143,144	2112	2112
Δ^{++}	uuu	1111	2224	2224	193,194	2224	2224
Δ^+	duu	1121	2214	2214	169,170	2214	2214
Δ^0	ddu	1221	2114	2114	175,176	2114	2114
Δ^-	ddd	2221	1114	1114	195,196	1114	1114

Strange Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Λ	dus	2130	3122	3122	121,122	3122	3122
Σ^+	uus	1130	3222	3222	139,140	3222	3222
Σ^0	dus	1230	3212	3212	129,130	3212	3212
Σ^-	dds	2230	3112	3112	145,146	3112	3112
Σ^{*+}	uus	1131	3224	3224	171,172	3224	3224
Σ^{*0}	dus	1231	3214	3214	161,162	3214	3214
Σ^{*-}	dds	2231	3114	3114	177,178	3114	3114
Ξ^0	uss	1330	3322	3322	149,150	3322	3322
Ξ^-	dss	2330	3312	3312	151,152	3312	3312
Ξ^{*0}	uss	1331	3324	3324	181,182	3324	3324
Ξ^{*-}	dss	2331	3314	3314	183,184	3314	3314
Ω^-	sss	3331	3334	3334	197,198	3334	3334

Charmed Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Λ_c^+	duc	2140	4122	4122	123,124	4122	4122
Σ_c^{++}	uuc	1140	4222	4222	141,142	4222	4222
Σ_c^+	duc	1240	4212	4212	131,132	4212	4212
Σ_c^0	ddc	2240	4112	4112	147,148	4112	4112
Σ_c^{*++}	uuc	1141	4224	4224	173,174		4224
Σ_c^{*+}	duc	1241	4214	4214	163,164		4214
Σ_c^{*0}	ddc	2241	4114	4114	179,180		4114
Ξ_c^+	usc	3140	4232	4232	125,126	4322	4232
Ξ_c^0	dsc	3240	4132	4132	127,128	4312	4132
$\Xi_c^{\prime +}$	usc	1340	4322	4322 (SS+)	133,134		4322
$\Xi_c^{\prime 0}$	dsc	2340	4312	4312 (SS0)	135,136		4312
Ξ_c^{*+}	usc	1341	4324	4324 (SS*+)	165,166		4324
Ξ_c^{*0}	dsc	2341	4314	4314 (SS*0)	167,168		4314
Ω_c^0	ssc	3340	4332	4332	153,154	4332	4332
Ω_c^{*0}	ssc	3341	4334	4334 (T^{*0})	185,186		4334
Ξ_{cc}^+	dcc	2440	4412		157,158		4412
Ξ_{cc}^{++}	ucc	1440	4422		155,156		4422
Ξ_{cc}^{*+}	dcc	2441	4414		189,190		4414
Ξ_{cc}^{*++}	ucc	1441	4424		187,188		4424
Ω_{cc}^+	scc	3440	4432		159,160		4432
Ω_{cc}^{*+}	scc	3441	4434		191,192		4434
Ω_{ccc}^{*++}	ccc	4441	4444		199,200		4444

Bottom Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Λ_b^0	dub	2150	5122	5122	401,402	5122	5122
Σ_b^-	ddb	2250	5112	5112	427,428		5112
Σ_b^0	dub	1250	5212		413,414		5212
Σ_b^+	uub	1150	5222	5222	425,426		5222
Σ_b^{*-}	ddb	2251	5114		455,456		5114
Σ_b^{*0}	dub	1251	5214		441,442		5214
Σ_b^{*+}	uub	1151	5224		453,454		5224
Ξ_b^-	dsb	3250	5132	5132	405,406		5132
Ξ_b^0	usb	3150	5232	5232	403,404		5232
$\Xi_b^{\prime -}$	dsb	2350	5312		417,418		5312
$\Xi_b^{\prime 0}$	usb	1350	5322		415,416		5322
Ξ_b^{*-}	dsb	2351	5314		445,446		5314
Ξ_b^{*0}	usb	1351	5324		443,444		5324
Ω_b^-	ssb	3350	5332	5332	429,430		5332
Ω_b^{*-}	ssb	3351	5334		457,458		5334
Ξ_{bc}^0	dcb	4250	5142		409,410		5142
Ξ_{bc}^+	ucb	4150	5242		407,408		5242
$\Xi_{bc}^{\prime 0}$	dcb	2450	5412		421,422		5412
$\Xi_{bc}^{\prime +}$	ucb	1450	5422		419,420		5422
Ξ_{bc}^{*0}	dcb	2451	5414		449,450		5414
Ξ_{bc}^{*+}	ucb	1451	5424		447,448		5424
Ω_{bc}^0	scb	4350	5342		411,412		5342
$\Omega_{bc}^{\prime 0}$	scb	3450	5432		423,424		5432
Ω_{bc}^{*0}	scb	3451	5434		451,452		5434
Ω_{bcc}^+	ccb	4450	5442		431,432		5442
Ω_{bcc}^{*+}	ccb	4451	5444		459,460		5444

Bottom Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Ξ_{bb}^{-}	dbb	2550	5512		435,436		5512
Ξ_{bb}^{0}	ubb	1550	5522		433,434		5522
Ξ_{bb}^{*-}	dbb	2551	5514		463,464		5514
Ξ_{bb}^{*0}	ubb	1551	5524		461,462		5524
Ω_{bb}^{-}	sbb	3550	5532		437,438		5532
Ω_{bb}^{*-}	sbb	3551	5534		465,466		5534
Ω_{bbc}^{0}	cbb	4550	5542		439,440		5542
Ω_{bbc}^{*0}	cbb	4551	5544		467,468		5544
Ω_{bbb}^{*-}	bbb	5551	5554		469,470		5554

Top Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Λ_t^+	dut	2160	6122	6122			6122
Σ_t^0	ddt	2260	6112	6112			6112
Σ_t^+	dut	1260	6212				6212
Σ_t^{++}	uut	1160	6222	6222			6222
Σ_t^{*0}	ddt	2261	6114				6114
Σ_t^{*+}	dut	1261	6214				6214
Σ_t^{*++}	uut	1161	6224				6224
Ξ_t^0	dst	3260	6132	6132			6132
Ξ_t^+	ust	3160	6232	6232			6232
$\Xi_t^{\prime 0}$	dst	2360	6312				6312
$\Xi_t^{\prime +}$	ust	1360	6322				6322
Ξ_t^{*0}	dst	2361	6314				6314
Ξ_t^{*+}	ust	1361	6324				6324
Ω_t^0	sst	3360	6332	6332			6332
Ω_t^{*0}	sst	3361	6334				6334
Ξ_{tc}^+	dct	4260	6142				6142
Ξ_{tc}^{++}	uct	4160	6242				6242
$\Xi_{tc}^{\prime +}$	dct	2460	6412				6412
$\Xi_{tc}^{\prime ++}$	uct	1460	6422				6422
Ξ_{tc}^{*+}	dct	2461	6414				6414
Ξ_{tc}^{*++}	uct	1461	6424				6424
Ω_{tc}^+	sct	4360	6342				6342
$\Omega_{tc}^{\prime +}$	sct	3460	6432				6432
Ω_{tc}^{*+}	sct	3461	6434				6434
Ω_{tcc}^{++}	cct	4460	6442				6442
$\Omega_{tcc}*^{++}$	cct	4461	6444				6444

Top Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Ξ^0_{tb}	dbt	5260	6152				6152
Ξ^+_{tb}	ubt	5160	6252				6252
Ξ'^0_{tb}	dbt	2560	6512				6512
Ξ'^+_{tb}	ubt	1560	6522				6522
Ξ^{*0}_{tb}	dbt	2561	6514				6514
Ξ^{*+}_{tb}	ubt	1561	6524				6524
Ω^0_{tb}	sbt	5360	6352				6352
Ω'^0_{tb}	sbt	3560	6532				6532
Ω^{*0}_{tb}	sbt	3561	6534				6534
Ω^+_{tbc}	cbt	5460	6452				6452
Ω'^+_{tbc}	cbt	4560	6542				6542
Ω^{*+}_{tbc}	cbt	4561	6544				6544
Ω^0_{tbb}	bbt	5560	6552				6552
Ω^{*0}_{tbb}	bbt	5561	6554				6554
Ξ^+_{tt}	dtt	2660	6612				6612
Ξ^{++}_{tt}	utt	1660	6622				6622
Ξ^{*+}_{tt}	dtt	2661	6614				6614
Ξ^{*++}_{tt}	utt	1661	6624				6624
Ω^+_{tt}	stt	3660	6632				6632
Ω^{*+}_{tt}	stt	3661	6634				6634
Ω^{++}_{ttc}	ctt	4660	6642				6642
Ω^{*++}_{ttc}	ctt	4661	6644				6644
Ω^+_{ttb}	btt	5660	6652				6652
Ω^{*+}_{ttb}	btt	5661	6654				6654
Ω^{*++}_{ttt}	ttt	6661	6664				6664