StdHep 5.06.01
PM0091

Fermilab

StdHep
5.06.01
Monte Carlo Standardization at FNAL
Fortran and C Implementation

Lynn Garren

November 20, 2006

mc-users@fnal.gov
http://cepa.fnal.gov/psm/stdhep/

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), (630) 840-2061, garren@fnal.gov. The StdHep home page is http://cepa.fnal.gov/psm/stdhep/.

Contents

1 Introduction

A basic particle numbering scheme was proposed by the Particle Data Group in 1988[1]. Practical application of this scheme exposed some limitations. After consultation[2], the scheme has been revised to better match the practice of program authors. The revised scheme includes numbering states by orbital and radial quantum numbers to allow systematic inclusion of quark model states which are as yet undiscovered. The new scheme also includes numbering for hypothetical particles such as SUSY particles. A standard output common block [3] has also been agreed to. Because the old P.D.G. convention allowed a fair amount of room for interpretation, the implementation tended to have minor variations among packages. With the new scheme, there is much less variation, but occasional differences remain. StdHep provides routines to convert Isajet, Pythia, 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.

This document is particular to the Fortran and C implentatons of StdHep.

The C++ implementation may be found in CLHEP[4], available at http://savannah.cern.ch/projects/clh

2 Particle Numbering Scheme

The standard particle numbering scheme is explained in full detail in reference [5]. The particle ID numbers between 1 and 80 are for elementary particles: quarks, gluons, leptons, gauge and higgs bosons, etc. Numbers 81-100 are for generator specific use. The list of elementary particles is in Appendix A.

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. Many states appearing in the PDG meson listing do not yet have definite $q\bar{q}$ model assignments. For these states, $n_{q_{2-3}}$ and n_J are assigned according to the state's most likely flavors and spin. Within these groups $n_L = 0, 1, 2, \ldots$ is used to distinguish states of increasing mass. These states are flagged with $n = 9$. 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 list of diquark states is in Appendix A.) A negative number indicates an antiparticle. The states are generally listed in order of increasing mass. K^0_L and K^0_S are exceptions. Their assigned identification numbers are 130 and 310, respectively.

SUSY particles are indicated with $n = 1$ for right-handed particles or $n = 2$ for left-handed particles. Technicolor states have $n = 3$. Excited (composite) quarks and leptons are identified by setting $n = 4$.

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.

Pythia, Herwig, and QQ use the PDG algorithm in addition to internal compressed numbering schemes. Although the latest implementations of Pythia, Herwig, and QQ conform closely to the new numbering scheme, some differences remain. The new numbering scheme attemts to list all states needed by the Monte Carlo generators. Appendix B contains a full list of meson states and their ID numbers, up through the top quark states. Isajet, Herwig, and Pythia contain the possibility of defining fourth family states.

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. The numbering for these states is explicitly stated in the new numbering scheme. Appendix C contains a full list of the baryon states.

An ad-hoc numbering scheme for ions was added as of StdHep 3.01. Ion numbers are FAAAZZZ00L. 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

3.1 HEPEVT

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 needed. 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 now the standard, as proposed for LEP 2 [11].

Nevhep is -1 for initialization records and -2 for final records. Pythia does not use event numbers, so nevhep is initialized the first time you call the Pythia 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 is in 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

StdHep defines 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. Know values of isthep are listed in Appendix D.

3.2 Parton Generators

A generic user process interface has been defined[12] to transfer event configurations from parton level generators to showering and hadronization event generators. This information is in the HEPRUP common block. Parton generator run information is in the HEPEUP common block. These common blocks can be found in $STDHEP_DIR/src/inc/hepeup.inc, hepeup.h, heprup.inc, and heprup.h.

StdHep provides I/O routines for these common blocks. In addition, there are routines to read various parton generator data files.
The user process event (hepeup) common block has the following form:

```
      integer maxnup
      parameter (maxnup=500)
      integer nup,idprup,idup,istup,mothup,icolup
      double precision xwgtup,scalup,aqedup,aqcdup,pup,vtimup,spinup
      common/hepeup/nup,idprup,xwgtup,scalup,aqedup,aqcdup,idup(maxnup),
     &istup(maxnup),mothup(2,maxnup),icolup(2,maxnup),pup(5,maxnup),
     &vtimup(maxnup),spinup(maxnup)
```

nup	-	number of particle entries in this event
idprup	-	ID of the process for this event (ID's are generator-specific)
xwgtup	-	event weight
scalup	-	scale of the event in GeV, as used to calculate PDFs
aqedup	-	QED coupling used for this event
aqcdup	-	QCD coupling used for this event
idup	-	particle ID according to PDG convention
istup	-	status code:
	-	-1 incoming particle
	-	$+1$ outgoing final state particle
	-	-2 intermediate space like propagator
	-	$+2$ intermediate resonance, mass should be preserved
	-	$+3$ intermediate resonance for documentation only
	-	-9 incoming beam particles (generally not needed)
mothup	-	index of first and last mother
icolup	-	tag for color flow lines
pup	-	lab frame 4 momentum and mass in GeV
vtimup	-	invariant lifetime
spinup	-	cos of angle between spin-vector of particle and 3 momentum of decaying particle

Scales 1 to 6 are defined for standard $2 \to 1 \to 2$ or $2 \to 2$ processes with $p1 + p2 \to q1 + q2$ kinematics. Additionally, scale 7 is defined for $2 \to 3$ processes, with $p1 + p2 \to q1 + q2 + q3$. Scales 8 to 10 are user defined.

scalup(1)	-	Q2 hard scale (used in PDF and couplings)
scalup(2)	-	Q2 scale of parton shower
scalup(3)	-	s-hat, invariant (p1+p2)**2
scalup(4)	-	t-hat, invariant (p1-q1)**2
scalup(5)	-	u-hat, invariant (p1-q2)**2
scalup(6)	-	squared transverse momentum of q1 (i.e., pt-hat**2)
scalup(7)	-	squared transverse momentum of q2
scalup(8)	-	user defined, 0 by default
scalup(9)	-	user defined, 0 by default
scalup(10)	-	user defined, 0 by default

The user process initialization (heprup) common block has the following form:

```
      integer maxpup
      parameter (maxpup=100)
      integer idbmup,pdfgup,pdfsup,idwtup,nprup,lprup
      double precision ebmup,xsecup,xerrup,xmaxup
      common/heprup/idbmup(2),ebmup(2),pdfgup(2),pdfsup(2),
     &idwtup,nprup,xsecup(maxpup),xerrup(maxpup),xmaxup(maxpup),
     &lprup(maxpup)
```

maxpup	-	max. number of different processes to be interfaced at one time
idbmup	-	ID of beam particles 1 and 2 according to the PDG convention
ebmup	-	energy in GeV of beam particles 1 and 2
pdfgup	-	author group for beam particles 1 and 2 according to PDFlib specifications
pdfsup	-	PDF set ID for beam particles 1 and 2 according to PDFlib specifications
idwtup	-	master switch dictating how the event weights are interpreted
nprup	-	number of different user subprocesses
xsecup	-	cross section for process in pb
xerrup	-	statistical error associated with XSECUP
xmaxup	-	maximum XWGTUP (in common block HEPEUP) for this process
lprup	-	user process ID's for this run

For e+e- or when the SHG defaults are to be used, set PDFGUP=-1 and PDFSUP=-1.

idwtup	event selection criteria	control of mixing or unweighting	xwgtup input	output
$+1$	xmaxup	SHG	+weighted	$+1$
-1	xmaxup	SHG	±weighted	±1
$+2$	xsecup	SHG	+ weighted	$+1$
-2	xsecup	SHG	±weighted	±1
$+3$		user interface	$+1$	$+1$
-3		user interface	±1	±1
$+4$		user interface	+weighted	+weighted
-4		user interface	±weighted	±weighted

3.3 Additional Information

Because much of the information in HEPEUP is also in HEPEVT, we have collected the extra information in the HEPEV4 common block. HEPEV4 is to be used after full event generation is finished. This common block is not intended for use when interfacing a parton generator to a hadron generator.
The HEPEV4 common block has the following form:

```
      double precision eventweightlh, scalelh
      double precision alphaqedlh, alphaqcdlh, spinlh
      integer          icolorflowlh, idruplh
      common/hepev4/eventweightlh, alphaqedlh, alphaqcdlh, scalelh(10),
     1              spinlh(3,nmxhep), icolorflowlh(2,nmxhep), idruplh
```

idruplh	-	The identity of the current process, as given by the LPRUP codes.
eventweightlh	-	The event weight: Equal to (total cross section)/(total generated) for the output of Pythia, Herwig, etc.
alphaqedlh	-	QED coupling α_{em}.
alphaqcdlh	-	QCD coupling α_s.
scalelh(10)	-	Squared Scale Q of the event
spinlh(3,...)	-	spin information
icolorflowlh(2,..)	-	(Anti-)Colour flow.

StdHep provides a routine to fill HEPEV4 from Pythia, as well as a MCFio interface to allow the user to read or write this information.

3.4 Multiple Interactions

Users sometimes want to combine signal events from one file with background events from another - usually in the context of simulating multiple beam collisions. StdHep provides common blocks and tools to enable the user to combine several events into a single "observed" event and to get information about the original events when requested.
Common blocks HEPEV2, HEPEV3, and HEPEV5 track the multiple interaction information.

```
      integer nmulti,jmulti
      common/hepev2/nmulti,jmulti(nmxhep)

      integer nmxmlt
      parameter (nmxmlt=16)
      integer nevmulti,itrkmulti,mltstr
      common/hepev3/nevmulti(nmxmlt),itrkmulti(nmxmlt),mltstr(nmxmlt)

      double precision eventweightmulti, scalemulti
      double precision alphaqedmulti, alphaqcdmulti
      integer          idrupmulti
      common/hepev5/eventweightmulti(nmxmlt),alphaqedmulti(nmxmlt),
     1              alphaqcdmulti(nmxmlt),scalemulti(10,nmxmlt),
     2              idrupmulti(nmxmlt)
```

nmulti	-	number of interactions in the list
jmulti(...)	-	multiple interaction number
nevmulti(i)	-	event number of original interaction
itrkmulti(i)	-	first particle in the original interaction
mltstr(i)	-	input stream identifier
idrupmulti(i)	-	identity of the original interaction
eventweightmulti(i)	-	event weight of the original interaction
alphaqedmulti(i)	-	QED coupling of the original interaction
alphaqcdmulti(i)	-	QCD coupling of the original interaction
scalemulti(...,i)	-	Scales of the original interaction

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, Pythia, Herwig, or QQ event format to standard output format and mconv = 2 converts standard output format to Isajet, Pythia, Herwig, or QQ event format. Please note that some information is lost if you convert back to generator format.

The routine pythia2ev4 fills the non-track related part of HEPEV4. This routine is called by lunhep(1).

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.

Use readmadgraph(version,lok) to read a MadGraph data file. Because we expect the MadGraph data file format to change, this routine takes a real argument "version", which is the MadGraph version number.

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 Pythia, and qqtran(id,mconv) for QQ. Id is the particle identification number to be translated to or from StdHep.

The routine std3to4(mconv) will translate from the old StdHep 3.x numbering scheme to the StdHep 4.06 numbering scheme. Particle ID translation is done by the function std3to4tran(id,mconv).

Integer function cnv1998to2000(id,mconv) translates particle ID numbers from the 1998 convention (last used for StdHep 4.09) to the 2000 convention[7]. Integer function cnv2000to2004(id,mconv) translates particle ID numbers from the 2000 convention (last used for StdHep 5.01) to the 2004 convention[9]. Integer function cnv2004to2006(id,mconv) translates particle ID numbers from the 2004 convention (last used for StdHep 5.05.05) to the 2006 convention[10]. These are small changes compared to the changes invoked by the new numbering scheme.

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.

The masses assigned to a given particle vary between generators. An attempt has been made to provide a consistent set of masses. However, particles can also be generated to lie quite far off the mass shell. The routine stdfixmass is called by stddecayqq to force particles onto the mass shell before passing them to QQ. Alternatively, stdfixpart(it) can be used to force a single particle to lie on the mass shell.

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 tables of particle masses and widths for known particles. These tables are distributed with the StdHep package.

The old style table (mass_width_2006.mc) is defined as environmental variable PDG_MASS_TBL. Two routines are supplied to read (pdgrdtb) or print (pdgprtb) the contents of this 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 new style table is mass_width_2006.csv, which can be read with pdgrdcsvtb.
The routine pdgrdtb fills the /STDTBL/ common block:

```
      integer nmxln
      parameter (nmxln=2000)
      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 for old style, i=3 for new style). 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.

A machine independent binary (xdr) event I/O option is provided. 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. 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. 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. 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
```

The supported values of ilbl are:

ilbl = 1	-	standard HEPEVT common block
ilbl = 2	-	HEPEVT, HEPEV2, and HEPEV3 common blocks
ilbl = 4	-	HEPEVT and HEPEV4 common blocks
ilbl = 5	-	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl = 11	-	HEPEUP common block
ilbl = 12	-	HEPRUP common block
ilbl = 100	-	StdHep begin run record
ilbl = 200	-	StdHep end run record

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/ is in include file stdcm1.inc (or stdcm1.h).

The routine stdxrdm(ilbl,istream,lok) allows 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 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,lnhpdf,lnhdmp,lnhrdm,ixdrstr
common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy,lnhpdf,lnhdmp,lnhrdm(16)
common/stdstr/ixdrstr(16)
```

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
lnhpdf	-	parton data function file unit number
lnhdmp	-	ascii dump file unit number
lnhrdm	-	array of input unit numbers for reading multiple files
ixdrstr	-	array of input streams for use by MCFio

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 available StdHep routines. More extensive details on the StdHep routines can be found in Section 10 or in $STDHEP_DIR/doc/stdhep.doc.

8 Linking

All routines are in the StdHep library and may be included by linking $STDHEP_DIR/lib/libstdhep.a after setting up stdhep. Pythia provides a similar translation routine, pyhepc, 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. If you use the C interface routines for MCFio, link with libstdhepC.a. MCFio documentation can be found in $MCFIO_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 Pythia event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Pythia 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

pythia2ev4 fill the non-track related part of HEPEV4.

std3to4(mconv) Translate between StdHep 3.04 and 4.06 numbering.

mconv	=	1	convert from StdHep 3.04 to StdHep 4.06
	=	2	convert from StdHep 4.06 to StdHep 3.04

cnv1998to2000(id,mconv) Translate between the new and old StdHep numbering schemes.

cnv1998to2000	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG 1998 (StdHep 4.09) to PDG 2000 (StdHep 4.10)
	=	2	convert from PDG 2000 (StdHep 4.10) to PDG 1998 (StdHep 4.09)

cnv2000to2004(id,mconv) Translate between the 2000 and 2004 PDG/StdHep numbering schemes.

cnv2000to2004	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG 2000 (StdHep 5.01) to PDG 2004 (StdHep 5.03)
	=	2	convert from PDG 2004 (StdHep 5.03) to PDG 2000 (StdHep 5.01)

cnv2004to2006(id,mconv) Translate between the 2004 and 2006 PDG/StdHep numbering schemes.

cnv2004to2006	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG 2004 (StdHep 5.05) to PDG 2006 (StdHep 5.06)
	=	2	convert from PDG 2006 (StdHep 5.06) to PDG 2004 (StdHep 5.05)

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

pdgrdcsvtb Read particle mass and width information from the new style PDG table. Fill the PDGTBL and STDTBL common blocks. IMPORTANT: This routine only works with mass_width_2006.csv. Use parsecsv_2004 for mass_width_2004.csv.

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 Pythia.

hptrlstq Print full translation list for QQ.

std3to4lst Print full translation list for new vs old numbering.

cnv98to00lst Print full translation list for PDG 1998 vs PDG 2000.

cnv2000to2004lst Print full translation list for PDG 2000 vs PDG 2004.

cnv2004to2006lst Print full translation list for PDG 2004 vs PDG 2006.

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 old style PDG table
	=	3	print complete list of particle properties from new style 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 Pythia particle id numbering.

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

pdgtonew(id,mconv) Translate between PDG96 and the new numbering scheme.

pdgtonew	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG96 to StdHep 4.06 numbering scheme
	=	2	convert from StdHep 4.06 to PDG96 numbering scheme

std3to4tran(id,mconv) Translate between the new and old StdHep numbering schemes.

std3to4tran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from StdHep 3.04 to StdHep 4.06 numbering scheme
	=	2	convert from StdHep 4.06 to StdHep 3.04 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.

stdfixmass Force all "stable" particles in the HEPEVT common block to lie on the mass shell. This routine assumes that phep(5) is the correct mass. Stdfixmass is called by stddecayqq.

stdfixpart(it) Force a single particle in the HEPEVT common block to lie on the mass shell. This routine assumes that phep(5) is the correct mass.

it = the index for this particle within HEPEVT

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,kql,kqj,kqr,kqx) 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
kqj	=	2*Jspin + 1
kql	=	n_L
kqr	=	n_r
kqx	=	used to tag various special or imprecisely defined states

stdcquarks(ip,kq1,kq2,kq3,kql,kqj,kqr,kqx) 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
kqj	=	2*Jspin + 1
kql	=	n_L
kqr	=	n_r
kqx	=	used to tag various special or imprecisely defined states

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

ip	=	index of particle in HEPEVT
xjsp	=	J spin

sum_4vec_d(nv,psum,p1,p2,...) Do a 4 vector sum of an arbitrary list of double precision 4 vectors.

nv	=	number of 4 vectors to sum
psum	=	output 4 vector
p1	=	the first input 4-vector
p2	=	the second input 4-vector

sum_4vec_f(nv,psum,p1,p2,...) Do a 4 vector sum of an arbitrary list of single precision 4 vectors.

nv	=	number of 4 vectors to sum
psum	=	output 4 vector
p1	=	the first input 4-vector
p2	=	the second input 4-vector

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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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 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.

zerohepeup Set the contents of the HEPEUP common block to zero.

readmadgraph(version,lok) Read a MadGraph ascii data file. At this time, the file contains information used to fill HEPEUP. The user must fill HEPRUP with information from other sources, and may choose to manipulate the information in HEPEUP before using it.

version	=		MadGraph version
lok	=	0	if no problems were encountered

stdprthd Print the contents of stdhd.inc, which contains the MCFio header information.

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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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, HEPEV2, and HEPEV3 common blocks
ilbl	=	4	HEPEVT and HEPEV4 common blocks
ilbl	=	5	HEPEVT, HEPEV2, HEPEV3, HEPEV4, and HEPEV5 common blocks
ilbl	=	11	HEPEUP common block
ilbl	=	12	HEPRUP common block
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.

StdHepZeroHEPEUP Set the contents of the hepeup_ structure to zero. Hepeup_ is a mirror of the HEPEUP common block.

version	=		MadGraph version
lok	=	0	if no problems were encountered

StdHepPrintHeader Print the contents of stdhd.h, which contains the MCFio header information.

10.6.3 Ascii Dump

dumphepeup(iev) Dump the contentents of the HEPEUP common block to an ascii file specified by lnhdmp in HEPLUN.

iev	=	event number
lnhdmp	=	unit number of dump file (in HEPLUN)

dumpheprup Dump the contentents of the HEPRUP common block to an ascii file specified by lnhdmp in HEPLUN.

11 History

5.06.01
- StdHep 5.06.01 was built with Pythia 6.413, Herwig 6.510a, Isajet 7.72a, and QQ 9.2b.
- Makefiles work with gfortran on MacOSX.
- Use PDG 2006 numbers.
- Get mass_width_2006.csv and mass_width_2006.mc.

5.05.05
- StdHep 5.05.05 was built with Pythia 6.402, Herwig 6.510a, Isajet 7.72a, and QQ 9.2b.
- Bug fix for stdstdsclst.
- Add stdhd.inc, stdhd.h, StdHepPrintHeader, and stdprthd to make title, comment, and date available when reading or writing mcfio files.

5.05.03
- StdHep 5.05.03 was built with Pythia 6.325, Herwig 6.510a, Isajet 7.72a, and QQ 9.2b.
- Translate IDTAUL and IDTAUR for Isajet.
- Read mass_width_2004.csv, which has a new format.
- Build only static libraries by default.

5.05
- StdHep 5.05 was built with Pythia 6.325, Herwig 6.510, Isajet 7.72, and QQ 9.2b.
- Using PDG 2004 numbers.
- Move generator includes into subdirectories of $STDHEP_DIR/src/inc.

5.04.02
- MCFio is part of StdHep again.

5.04
- StdHep 5.04 was built with Pythia 6.319, Herwig 6.507, Isajet 7.51, QQ 9.2b, and MCFio 5.3.
- Using PDG 2004 numbers.

5.03
- StdHep 5.03 was built with Pythia 6.223, Herwig 6.505, Isajet 7.51, and QQ 9.2b.
- MCFio has been moved to a separate product. StdHep 5.03 was built with MCFio 5.3.

- Using PDG 2004 numbers (use cnv2000to2004 when using files written by StdHep 5.01).

5.02
- StdHep 5.02 was built with Pythia 6.216, Herwig 6.5, Isajet 7.51, and QQ 9.2b.
- Add routines to fill and write Les Houches common blocks HEPEUP and HEPRUP.
- Read MadGraph data files.
- Supply stdhep_mcfio.h header with prototypes for C I/O routines.
- Change STDCM1.

5.01
- StdHep 5.01 was built with Pythia 6.203, Herwig 6.4, Isajet 7.51, and QQ 9.2b.
- Pythia has changed and added numbers.
- StdHepC++ is distributed with CLHEP, not with the Fortran stdhep.
- Define common blocks HEPEV4 and HEPEV5.

4.10
- StdHep 4.10 was built with Pythia 6.155, Herwig 6.2, Isajet 7.51, and QQ 9.2b.
- PDG 2000 numbers (use cnv409to410).
- Add more StdHepC++ utilities.

4.09
- StdHep 4.09 was built with Pythia 6.129, Herwig 6.1, Isajet 7.37, and QQ 9.2b.
- Restructure StdHepC++ classes.

4.08
- StdHep 4.08 was built with Pythia 6.129, Herwig 6.1, Isajet 7.37, and QQ 9.2b.
- Modify hep2geant to only add non-degenerate vertices.

4.07
- StdHep 4.07 was built with Pythia 6.125, Herwig 5.9a, Isajet 7.37, and QQ 9.2b.
- Add status 23 for isajet "initial" partons.
- Makefiles support building debug libraries (e.g. gmake DEBUG=-g all).
- Begin to add C++ binding as libStdHepCXX.a.

4.06
- StdHep 4.06 was built with Pythia 6.125, Herwig 5.9a, Isajet 7.37, and QQ 9.2b.
- Test with Isajet 7.42 (no modifications required).
- Fix translation of Isajet neutralinos.
- Fix typo in hepchg (present in versions 4.01 to 4.05).

4.05
- Update table file for UPS.
- Bug fix for mcfio.

4.04
- StdHep 4.04 was built with Pythia 6.115, Herwig 5.9a, Isajet 7.37, and QQ 9.2b.
- Include more information for multiple interaction events.
- Update mcfio.

4.03
- StdHep 4.03 was built with Pythia 6.115, Herwig 5.9a, Isajet 7.37, and QQ 9.2b.

- Bug fixes for qqtran and hepnam.
- New PDG mass table using the new numbering scheme.

4.02
- StdHep 4.02 was built with Pythia 6.115, Herwig 5.9a, Isajet 7.31, and QQ 9.2a.
- Late-breaking modifications to the new numbering scheme were implemented.

4.01
- StdHep 4.01 was built with Pythia 6.115, Herwig 5.9, Isajet 7.31, and QQ 9.2a.
- The new numbering scheme was implemented.

4.00
- StdHep 4.00 was built with Pythia 6.114, Herwig 5.9, Isajet 7.31, and QQ 9.2.
- Native mode binary and Zebra I/O routines were dropped.
- Translation routines for DPMJET 2.4 were added.
- Fortran preprocessing is now handled by cpp.
- Use the same /HEPEVT/ common block for Herwig and StdHep. The translation routine changes idhep in place.

3.05
- StdHep 3.05 was built with Jetset 7.0408, Herwig 5.8a, Isajet 7.16, and QQ 9.2.
- Change qqtran for 4 QQ particle translations.

3.04
- StdHep 3.04 was built with Jetset 7.0408, Herwig 5.8a, Isajet 7.16, and QQ 9.2.
- Fix XDR I/O bug (affected final particle in event list).

3.03
- StdHep 3.03 was built with Jetset 7.0408, Herwig 5.8a, Isajet 7.16, and QQ 9.2.
- Add qqufile to stdluns.inc.
- Add stdfixmass to adjust particle energy if mass is off shell.
- Add 4 vector sum routines (Rob Kutschke).
- Improved logic for stdchgdsclst (Rob Kutschke).
- Fix hep2geant interface to convert double precision to real.

3.02
- StdHep 3.02 was built with Jetset 7.0408, Herwig 5.8, Isajet 7.13, and QQ 9.1.
- Compile OSF1 version with floating point checking on.
- Fix some bugs in display package (Phase and Space).

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 [11].

- 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 [13], 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 [14], 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] Particle Data Group: G.P. Yost *et al.*, Phys. Lett. **B204**, 1 (1988).

[2] I. G. Knowles *et al.*, in "Physics at LEP2", CERN 96-01, vol. 2, p. 103.

[3] T. Sjöstrand *et al.*, in "Z physics at LEP1", CERN 89-08, vol. 3, p.327.

[4] CLHEP - A Class Library for High Energy Physics http://wwwasd.web.cern.ch/wwwasd/lhc++/clhep/ and http://savannah.cern.ch/projects/clhep/

[5] Particle Data Group: W-M Yao *et al.*, Journal of Physics **G33**, 1 (2006). http://www-pdg.lbl.gov/2007/mcdata/mc_particle_id_contents.html

[6] Particle Data Group: C. Caso *et al.*, The European Physical Journal **C3** (1998) 180.

[7] Particle Data Group: D.E. Groom *et al.*, The European Physical Journal **C15** (2000) 205.

[8] Particle Data Group: K. Hagiwara *et al.*, Physical Review **D66**, 1 (2002).

[9] Particle Data Group: S. Eidelman *et al.*, Physics Letters **B592**, 1 (2004). http://www-pdg.lbl.gov/

[10] Particle Data Group: W-M Yao *et al.*, Journal of Physics **G33**, 1 (2006). http://www-pdg.lbl.gov/

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

[12] E. Boos *et al.*, arXiv:hep-ph/0109068 (2001).

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

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

A Elementary Particle Identification Code Listing

Numbers which have changed since StdHep 5.05 and PDG 2004 are in bold text.

Useful conversion functions (see section 10.1)
cnv1998to2000 - convert between PDG 1998 and PDG 2000 numbers
cnv2000to2004 - convert between PDG 2000 and PDG 2004 numbers
use cnv2000to2004 when reading files written by StdHep 5.01
cnv2004to2006 - convert between PDG 2004 and PDG 2006 numbers
use cnv2004to2006 when reading files written by StdHep 5.03.xx to 5.05.xx

Quarks and Leptons						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
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
b'	7 (y)	7	7 (v & h)		7	7
t'	8 (x)	8	8 (a & h')		8	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
τ'^-		17			17	17
$\nu_{\tau'}$		18			18	18

Gauge and Higgs Bosons						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
g	9	21	21	-13	21 (9)	21 (9)
γ	10	22	22	1	22	22
$\gamma_{virtual}$						10022
$Cerenkov$						20022
Z^0	90	23	23	2	23	23
W^+	80	24	24	3,4	24	24
h^0/H_1^0	81	25	25		25	25
Z'/Z_2^0		32	32		32	32
Z''/Z_3^0		33			33	33
W'/W_2^+		34			34	34
H^0/H_2^0	83 (H_H^0)	35	35		35	35
A^0/H_3^0	84 (H_A^0)	36	36		36	36
H^+	86	37	37		37	37
H_1^{++}	88					52
H_2^+	87					53
H_2^{++}	89					54
H_4^0	85					55
H_5^0						
$H_L^{0\,NOTE}$	82		26			51

Special Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
G (graviton)	92	39	39		39	39
R^0		41			41	41
LQ^c		42			42	42
$reggeon$		110			110	110
$pomeron$		990			990	990
$odderon$					9990	9990
internal code		81-99	81-91	82-93	81-100	81-100

NOTE: H_L^0 is redundant with h^0/H_1^0, but is given a different number in Isajet and Herwig.

Supersymmetric Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$\tilde{d}_L$	22	1000001	1000001		1000001	1000001
$\tilde{u}_L$	21	1000002	1000002		1000002	1000002
$\tilde{s}_L$	23	1000003	1000003		1000003	1000003
$\tilde{c}_L$	24	1000004	1000004		1000004	1000004
$\tilde{b}_1/\tilde{b}_L$	25	1000005	1000005		1000005	1000005
$\tilde{t}_1/\tilde{t}_L$	26	1000006	1000006		1000006	1000006
$\tilde{e}_L^-$	32	1000011	1000011		1000011	1000011
$\tilde{\nu}_{eL}$	31	1000012	1000012		1000012	1000012
$\tilde{\mu}_L^-$	34	1000013	1000013		1000013	1000013
$\tilde{\nu}_{\mu L}$	33	1000014	1000014		1000014	1000014
$\tilde{\tau}_1^-/\tilde{\tau}_L^-$	36	1000015	1000015		1000015	1000015
$\tilde{\nu}_{\tau L}$	35	1000016	1000016		1000016	1000016
$\tilde{d}_R$	42	2000001	2000001		2000001	2000001
$\tilde{u}_R$	41	2000002	2000002		2000002	2000002
$\tilde{s}_R$	43	2000003	2000003		2000003	2000003
$\tilde{c}_R$	44	2000004	2000004		2000004	2000004
$\tilde{b}_2/\tilde{b}_R$	45	2000005	2000005		2000005	2000005
$\tilde{t}_2/\tilde{t}_R$	46	2000006	2000006		2000006	2000006
$\tilde{e}_R^-$	52	2000011	2000011		2000011	2000011
$\tilde{\nu}_{eR}$	51	2000012	2000012			2000012
$\tilde{\mu}_R^-$	54	2000013	2000013		2000013	2000013
$\tilde{\nu}_{\mu R}$	53	2000014	2000014			2000014
$\tilde{\tau}_2^-/\tilde{\tau}_R^-$	56	2000015	2000015		2000015	2000015
$\tilde{\nu}_{\tau R}$	55	2000016	2000016			2000016

Supersymmetric Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$\tilde{g}$	29	1000021	1000021		1000021	1000021
$\tilde{\chi}_1^0/\tilde{\gamma}$	30 (Z_1^{ss})	1000022	1000022		1000022	1000022
$\tilde{\chi}_2^0/\tilde{Z}^0$	40 (Z_2^{ss})	1000023	1000023		1000023	1000023
$\tilde{\chi}_1^+/\tilde{W}^+$	39 (W_1^{+ss})	1000024	1000024		1000024	1000024
$\tilde{\chi}_3^0/\tilde{H}_1^0$	50 (Z_3^{ss})	1000025	1000025		1000025	1000025
$\tilde{\chi}_4^0/\tilde{H}_2^0$	60 (Z_4^{ss})	1000035	1000035		1000035	1000035
$\tilde{\chi}_2^+/\tilde{H}^+$	49 (W_2^{+ss})	1000037	1000037		1000037	1000037
$\tilde{G}$	91	1000039	1000039		1000039	1000039

R-Hadrons						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$R^0_{\tilde{g}g}$					**1000993**	**1000993**
$R^0_{\tilde{g}d\bar{d}}$					**1009113**	**1009113**
$R^+_{\tilde{g}u\bar{d}}$					**1009213**	**1009213**
$R^0_{\tilde{g}u\bar{u}}$					**1009223**	**1009223**
$R^0_{\tilde{g}d\bar{s}}$					**1009313**	**1009313**
$R^+_{\tilde{g}u\bar{s}}$					**1009323**	**1009323**
$R^0_{\tilde{g}s\bar{s}}$					**1009333**	**1009333**
$R^-_{\tilde{g}ddd}$					**1091114**	**1091114**
$R^0_{\tilde{g}udd}$					**1092114**	**1092114**
$R^+_{\tilde{g}uud}$					**1092214**	**1092214**
$R^{++}_{\tilde{g}uuu}$					**1092224**	**1092224**
$R^-_{\tilde{g}sdd}$					**1093114**	**1093114**
$R^0_{\tilde{g}sud}$					**1093214**	**1093214**
$R^+_{\tilde{g}suu}$					**1093224**	**1093224**
$R^-_{\tilde{g}ssd}$					**1093314**	**1093314**
$R^0_{\tilde{g}ssu}$					**1093324**	**1093324**
$R^-_{\tilde{g}sss}$					**1093334**	**1093334**
$R^+_{\tilde{t}_1\bar{d}}$					**1000612**	**1000612**
$R^0_{\tilde{t}_1\bar{u}}$					**1000622**	**1000622**
$R^+_{\tilde{t}_1\bar{s}}$					**1000632**	**1000632**
$R^0_{\tilde{t}_1\bar{c}}$					**1000642**	**1000642**
$R^+_{\tilde{t}_1\bar{b}}$					**1000652**	**1000652**
$R^0_{\tilde{t}_1dd_1}$					**1006113**	**1006113**
$R^+_{\tilde{t}_1ud_0}$					**1006211**	**1006211**
$R^+_{\tilde{t}_1ud_1}$					**1006213**	**1006213**
$R^{++}_{\tilde{t}_1uu_1}$					**1006223**	**1006223**
$R^0_{\tilde{t}_1sd_0}$					**1006311**	**1006311**
$R^0_{\tilde{t}_1sd_1}$					**1006313**	**1006313**
$R^+_{\tilde{t}_1su_0}$					**1006321**	**1006321**
$R^+_{\tilde{t}_1su_1}$					**1006323**	**1006323**
$R^0_{\tilde{t}_1ss_1}$					**1006333**	**1006333**

Diquarks						
Diquark	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$(ud)_0$	1200	2101	2101		2101	2101
$(sd)_0$	2300	3101	3101		3101	3101
$(su)_0$	1300	3201	3201		3201	3201
$(cd)_0$	-2400	4101			4101	4101
$(cu)_0$	-1400	4201			4201	4201
$(cs)_0$	-3400	4301			4301	4301
$(bd)_0$	2500	5101			5101	5101
$(bu)_0$	1500	5201			5201	5201
$(bs)_0$	3500	5301			5301	5301
$(bc)_0$	4500	5401			5401	5401
$(dd)_1$	2200	1103	1103		1103	1103
$(ud)_1$		2103			2103	2103
$(uu)_1$	1100	2203	2203		2203	2203
$(sd)_1$		3103			3103	3103
$(su)_1$		3203			3203	3203
$(ss)_1$	3300	3303	3303		3303	3303
$(cd)_1$		4103			4103	4103
$(cu)_1$		4203			4203	4203
$(cs)_1$		4303			4303	4303
$(cc)_1$	4400	4403			4403	4403
$(bd)_1$		5103			5103	5103
$(bu)_1$		5203			5203	5203
$(bs)_1$		5303			5303	5303
$(bc)_1$		5403			5403	5403
$(bb)_1$	5500	5503			5503	5503

Technicolor Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
π^0_{tech}		3000111			3000111	3000111
π^+_{tech}		3000211			3000211	3000211
$\pi'^0_{tech}{}^{NOTE}$		3000221			3000221	3000221
$\eta^0_{tech}{}^{NOTE}$		3000331			3100221	3100221
ρ^0_{tech}		3000113			3000113	3000113
ρ^+_{tech}		3000213			3000213	3000213
ω^0_{tech}		3000223			3000223	3000223
V_8		3100021			3100021	3100021
$\pi^1_{tech_{22}}$		3100111			3060111	3060111
$\pi^8_{tech_{22}}$		3200111			3160111	3160111
$\rho_{tech_{11}}$		3100113			3130113	3130113
$\rho_{tech_{12}}$		3200113			3140113	3140113
$\rho_{tech_{21}}$		3300113			3150113	3150113
$\rho_{tech_{22}}$		3400113			3160113	3160113
NOTE: Newer technicolor models use π'^0_{tech} instead of η^0_{tech}.						

Excited Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
d^*		4000001			4000001	4000001
u^*		4000002			4000002	4000002
e^*		4000011			4000011	4000011
ν^*_e		4000012			4000012	4000012
G^*		5000039				4000039
Other Exotics						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
ν_{Re}		9900012				9900012
$\nu_{R\mu}$		9900014				9900014
$\nu_{R\tau}$		9900016				9900016
Z^0_R		9900023				9900023
W^+_R		9900024				9900024
H^{++}_L		9900041				9900061
H^{++}_R		9900042				9900062

Pentaquarks						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Θ^+					9221132	9221132
Φ^{--}					9331122	9331122
Miscellaneous Particles						
Particle	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
ρ^0_{diffr}		9900110				9910113
π^+_{diffr}		9900210				9910211
ω^0_{diffr}		9900220				9910223
ϕ^0_{diffr}		9900330				9910333
J/ψ^0_{diffr}		9900440				9910443
n^0_{diffr}		9902110				9912112
p^+_{diffr}		9902210				9912212
$ccbar[1S0(8)]$		9900441				9900441
$ccbar[3S1(8)]$		9900443				9900443
$ccbar[3P0(8)]$		9910441				9910441
$bbbar[1S0(8)]$		9900551				9900551
$bbbar[3S1(8)]$		9900553				9900553
$bbbar[3P0(8)]$		9910551				9910551
remnant photon			98			9920022
remnant nucleon			99			9922212
Hydrogen						1001001002
Deuterium						1002001002
Tritium						1003001002
He^3						1003002001
α						1004002001
geantino						

B Complete Meson Particle Identification Code Listing

Numbers which have changed since StdHep 5.05.xx and PDG 2004 are in bold text.

Light Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
π^0	110	111	111	51	111	111
π^+	120	211	211	21,22	211	211
$a_0^0(980)$		10111	9000111	251	9000111	9000111
$a_0^+(980)$		10211	9000211	252,253	9000211	9000211
$\pi^0(1300)$					100111	100111
$\pi^+(1300)$					100211	100211
$a_0^0(1450)$			10111		10111	10111
$a_0^+(1450)$			10211		10211	10211
$\pi^0(1800)$					9010111	9010111
$\pi^+(1800)$					9010211	9010211
$\rho^0(770)$	111	113	113	91	113	113
$\rho^+(770)$	121	213	213	61,62	213	213
$b_1^0(1235)$		10113	10113	265	10113	10113
$b_1^+(1235)$		10213	10213	266,267	10213	10213
$a_1^0(1260)$	10111	20113	20113	108	20113	20113
$a_1^+(1260)$	10121	20213	20213	107,109	20213	20213
$\pi_1^0(1400)^{NOTE}$					9000113	9000113
$\pi_1^+(1400)^{NOTE}$					9000213	9000213
$\rho^0(1450)$					100113	100113
$\rho^+(1450)$					100213	100213
$\pi_1^0(1600)$					9010113	9010113
$\pi_1^+(1600)$					9010213	9010213
$a_1^0(1640)$					**9020113**	**9020113**
$a_1^+(1640)$					**9020213**	**9020213**
$\rho^0(1700)/\rho^0(D)$			30113	106	30113	30113
$\rho^+(1700)/\rho^+(D)$			30213	205,206	30213	30213
$\rho^0(1900)$					**9030113**	**9030113**
$\rho^+(1900)$					**9030213**	**9030213**
$\rho^0(2150)$					**9040113**	**9040113**
$\rho^+(2150)$					**9040213**	**9040213**
NOTE: $\pi_1^0(1400)$ was $\hat{\rho}(1405)$ in PDG 1998.						

Light Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$a_2^0(1320)$		115	115	254	115	115
$a_2^+(1320)$		215	215	255,256	215	215
$\pi_2^0(1670)$			10115		10115	10115
$\pi_2^+(1670)$			10215		10215	10215
$a_2^0(1700)$					**9000115**	**9000115**
$a_2^+(1700)$					**9000215**	**9000215**
$\pi_2^0(2100)$					**9010115**	**9010115**
$\pi_2^+(2100)$					**9010215**	**9010215**
$\rho_3^0(1690)$			117		117	117
$\rho_3^+(1690)$			217		217	217
$\rho_3^0(1990)$					9000117	9000117
$\rho_3^+(1990)$					9000217	9000217
$\rho_3^0(2250)$					9010117	9010117
$\rho_3^+(2250)$					9010217	9010217
$a_4^0(2040)$					119	119
$a_4^+(2040)$					219	219

$u\bar{u}$, $d\bar{d}$, and $s\bar{s}$ Meson admixtures						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
η	220	221	221	52	221	221
$\eta'(958)$	330	331	331	53	331	331
$f_0(600)^{NOTE}$					9000221	9000221
$f_0(980)$	10110	10221	9010221 f'_0	257	9010221	9010221
$\eta(1295)$					100221	100221
$f_0(1370)/f'_0$		10331	10221 $f^0_0(H)$	258	10221	10221
$\eta(1405)$					9020221	9020221
$\eta(1475)^{NOTE}$					100331	100331
$f_0(1500)$					9030221	9030221
$f_0(1710)^{NOTE}$					10331	10331
$\eta(1760)$					**9040221**	**9040221**
$f_0(2020)$					**9050221**	**9050221**
$f_0(2100)^{NOTE}$					**9060221**	**9060221**
$f_0(2200)$					**9070221**	**9070221**
$\eta(2225)$					**9080221**	**9080221**
$\omega(782)$	221	223	223	92	223	223
$\phi(1020)$	331	333	333	93	333	333
$h_1(1170)$		10223	10223	263	10223	10223
$f_1(1285)$		20223	20223	259	20223	20223
$h_1(1380)/h'_1$		10333	10333	264	10333	10333
$f_1(1420)/f'_1$		20333	20333 $f_1(H)$	260	20333	20333
$\omega(1420)$					100223	100223
$f_1(1510)$					9000223	9000223
$h_1(1595)$					**9010223**	**9010223**
$\omega(1650)$			30223		30223	30223
$\phi(1680)$					100333	100333

NOTE: $f_0(600)$ was $f_0(400-1200)$ in PDG 2000.

NOTE: $\eta(1475)$ was $\eta(1440)$ in PDG 2002.

NOTE: $f_0(1710)$ was $f_J(1710)$ in PDG 1998.

NOTE: $f_0(2100)$ was $f_0(2060)$ in PDG 2000.

Also, Herwig's $f_1(H)$ had a mass of 1512 in Herwig 5.9 and a mass of 1426 as of Herwig 6.1.

$u\bar{u}$, $d\bar{d}$, and $s\bar{s}$ Meson admixtures						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$f_2(1270)$	112	225	225	261	225	225
$f_2(1430)$					9000225	9000225
$f_2'(1525)$		335	335	262	335	335
$f_2(1565)$					9010225	9010225
$f_2(1640)$					9020225	9020225
$\eta_2(1645)$			10225		10225	10225
$f_2(1810)$					9030225	9030225
$\eta_2(1870)$			10335		10335	10335
$f_2(1910)$					9040225	9040225
$f_2(1950)$					9050225	9050225
$f_2(2010)$					9060225	9060225
$f_2(2150)$					9070225	9070225
$f_2(2300)$					9080225	9080225
$f_2(2340)$					9090225	9090225
$\omega_3(1670)$			227		227	227
$\phi_3(1850)$			337		337	337
$f_4(2050)$					229	229
$f_J(2220)$					9000229	9000229
$f_4(2300)$					9010229	9010229

Strange Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
K_S^0	20	310	310	57	310	310
K_L^0	-20	130	130	58	130	130
K^0	230	311	311	25,26	311	311
K^+	130	321	321	23,24	321	321
$K_0^{*0}(800)$					9000311	9000311
$K_0^{*+}(800)$					9000321	9000321
$K_0^{*0}(1430)$		10311	10311	268,269	10311	10311
$K_0^{*+}(1430)$		10321	10321	270,271	10321	10321
$K^0(1460)$					100311	100311
$K^+(1460)$					100321	100321
$K^0(1830)$					**9010311**	**9010311**
$K^+(1830)$					**9010321**	**9010321**
$K_0^{*0}(1950)$					**9020311**	**9020311**
$K_0^{*+}(1950)$					**9020321**	**9020321**
$K^{*0}(892)$	231	313	313	65,66	313	313
$K^{*+}(892)$	131	323	323	63,64	323	323
$K_1^0(1270)$	10231	10313	10313 ($K_1^0(L)$)	59,60	10313	10313
$K_1^+(1270)$	10131	10323	10323 ($K_1^+(L)$)	201,202	10323	10323
$K_1^0(1400)$		20313 (K_1^{*0})	20313 ($K_1^0(H)$)	19,20	20313	20313
$K_1^+(1400)$		20323 (K_1^{*+})	20323 ($K_1^+(H)$)	203,204	20323	20323
$K^{*0}(1410)$	30231				100313	100313
$K^{*+}(1410)$	30131				100323	100323
$K_1^0(1650)$					**9000313**	**9000313**
$K_1^+(1650)$					**9000323**	**9000323**
$K^{*0}(1680)$			30313		30313	30313
$K^{*+}(1680)$			30323		30323	30323
In Herwig 6.4, $K_1^0(L)$ has mass 1570 and $K_1^0(H)$ has mass 1850.						

Strange Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$K_2^{*0}(1430)$	232	315	315	272,273	315	315
$K_2^{*+}(1430)$	132	325	325	274,275	325	325
$K_2^0(1580)$					9000315	9000315
$K_2^+(1580)$					9000325	9000325
$K_2^0(1770)$			10315		10315	10315
$K_2^+(1770)$			10325		10325	10325
$K_2^0(1820)$			20315		20315	20315
$K_2^+(1820)$			20325		20325	20325
$K_2^{*0}(1980)$					**9010315**	**9010315**
$K_2^{*+}(1980)$					**9010325**	**9010325**
$K_2^0(2250)$					**9020315**	**9020315**
$K_2^+(2250)$					**9020325**	**9020325**
$K_3^{*0}(1780)$			317	278,279	317	317
$K_3^{*+}(1780)$			327	280,281	327	327
$K_3^0(2320)$					9010317	9010317
$K_3^+(2320)$					9010327	9010327
$K_4^{*0}(2045)$					319	319
$K_4^{*+}(2045)$					329	329
$K_4^0(2500)$					9000319	9000319
$K_4^+(2500)$					9000329	9000329

Charmed Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
D^+	-240	411	411	29,30	411	411
D^0	-140	421	421	27,28	421	421
$D_0^{*+}(2400)$		10411	10411	221,225	10411	10411
$D_0^{*0}(2400)$		10421	10421	229,233	10421	10421
$D(2S)^+$						100411
$D(2S)^0$						100421
$D^{*+}(2010)$	-241	413	413	69,70	413	413
$D^{*0}(2007)$	-141	423	423	67,68	423	423
$D_1^+(2420)/D_1^+(L)$		10413	10413	223,227	10413	10413
$D_1^0(2420)/D_1^0(L)$		10423	10423	231,235	10423	10423
$D_1^+(H)/D_1^{*+}$		20413	20413	222,226	20413	20413
$D_1^0(2430)/D_1^{*0}$		20423	20423	230,234	20423	20423
$D(2S)^{*+}$						100413
$D(2S)^{*0}$						100423
$D_2^{*+}(2460)$		415	415	224,228	415	415
$D_2^{*0}(2460)$		425	425	232,236	425	425
D_s^+	-340 (F^+)	431	431	31,32	431	431
$D_{s0}^{*+}(2317)$		10431	10431	237,241	10431	10431
D_s^{*+}	-341 (F^{*+})	433	433	71,72	433	433
$D_{s1}^+(2536)/D_{s1}^+(L)$		10433	10433	239,243	10433	10433
$D_{s1}^+(2460)/D_{s1}^{*+}$		20433	20433	238,242	20433	20433
$D_{s2}^{*+}(2573)$		435	435	240,244	435	435

$c\bar{c}$ Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$\eta_c(1S)$	440	441	441	54	441	441
$\chi_{c0}(1P)$	20440	10441	10441 (χ_{c1})	110	10441	10441
$\eta_c(2S)$					100441	100441
$J/\psi(1S)$	441	443	443	94	443	443
$h_c(1P)$		10443	10443	215	10443	10443
$\chi_{c1}(1P)$	20441	20443	20443 (χ_{c0})	111	20443	20443
$\psi(2S)/\psi'$	10441	100443	100443	219	100443	100443
$\psi(3770)$			30443	211	30443	30443
$\psi(4040)$				212	9000443	9000443
$\psi(4160)$				213	9010443	9010443
$\psi(4415)$				214	9020443	9020443
$\chi_{c2}(1P)$	20442	445	445	112	445	445
$\chi_{c2}(2P)$					**100445**	**100445**
$\psi(3836)$					9000445	9000445

Bottom Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
B^0	250	511	511	36,35	511	511
B^+	150	521	521	34,33	521	521
B_0^{*0}		10511	10511	479,483	10511	10511
B_0^{*+}		10521	10521	471,475	10521	10521
B^{*0}	251	513	513	76,75	513	513
B^{*+}	151	523	523	74,73	523	523
$B_1^0(L)$		10513	10513	481,485	10513	10513
$B_1^+(L)$		10523	10523	473,477	10523	10523
$B_1^0(H)/B_1^{*0}$		20513	20513	480,484	20513	20513
$B_1^+(H)/B_1^{*+}$		20523	20523	472,476	20523	20523
B_2^{*0}		515	515	482,486	515	515
B_2^{*+}		525	525	474,478	525	525
B_s^0	350	531	531	38,37	531	531
B_{s0}^{*0}		10531	10531	487,491	10531	10531
B_s^{*0}	351	533	533	78,77	533	533
$B_{s1}^0(L)$		10533	10533	489,493	10533	10533
$B_{s1}^0(H)/B_{s1}^{*0}$		20533	20533	488,492	20533	20533
B_{s2}^{*0}		535	535	490,494	535	535
B_c^+	450	541	541	40,39	541	541
B_{c0}^{*+}		10541	10541		10541	10541
B_c^{*+}	451	543	543	80,79	543	543
$B_{c1}^+(L)$		10543	10543		10543	10543
$B_{c1}^+(H)/B_{c1}^{*+}$		20543	20543		20543	20543
B_{c2}^{*+}		545	545		545	545

$b\bar{b}$ Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
$\eta_b(1S)$	550	551	551	55	551	551
$\chi_{b0}(1P)$		10551	10551	100	10551	10551
$\eta_b(2S)$					100551	100551
$\chi_{b0}(2P)$			110551	103	110551	110551
$\eta_b(3S)$					200551	200551
$\chi_{b0}(3P)$					210551	210551
$\Upsilon(1S)$	551	553	553	95	553	553
$h_b(1P)$		10553	10553	218	10553	10553
$\chi_{b1}(1P)$		20553	20553	101	20553	20553
$\Upsilon_1(1D)$					30553	30553
$\Upsilon(2S)/\Upsilon'$		100553	100553	97	100553	100553
$h_b(2P)$					110553	110553
$\chi_{b1}(2P)$			120553	104	120553	120553
$\Upsilon_1(2D)$					130553	130553
$\Upsilon(3S)$			200553	98	200553	200553
$h_b(3P)$					210553	210553
$\chi_{b1}(3P)$					220553	220553
$\Upsilon(4S)$			300553	99	300553	300553
$\Upsilon(10860)$				216	9000553	9000553
$\Upsilon(11020)$				217	9010553	9010553
$\Upsilon(7S)$				220		9020553
$\chi_{b2}(1P)$		555	555	102	555	555
$\eta_{b2}(1D)$					10555	10555
$\Upsilon_2(1D)$					20555	20555
$\chi_{b2}(2P)$			100555	105	100555	100555
$\eta_{b2}(2D)$					110555	110555
$\Upsilon_2(2D)$					120555	120555
$\chi_{b2}(3P)$					200555	200555
$\Upsilon_3(1D)$					557	557
$\Upsilon_3(2D)$					100557	100557

Top Mesons						
Meson	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
T^{+}	-260		611	43,44		611
T^{0}	-160		621	41,42		621
T^{*+}	-261			83,84		613
T^{*0}	-161			81,82		623
T_s^{+}	-360		631	45,46		631
T_s^{*+}	-361			85,86		633
T_c^{0}	460		641	47,48		641
T_c^{*0}	461			87,88		643
T_b^{+}	-560		651	49,50		651
T_b^{*+}	-561			89,90		653
η_t	660			56		661
θ	661		663	96 (ψ_t)		663

C List of baryon states

Light Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
p	1120	2212	2212	137,138	2212	2212
n	1220	2112	2112	143,144	2112	2112
Δ^{++}	1111	2224	2224	193,194	2224	2224
Δ^{+}	1121	2214	2214	169,170	2214	2214
Δ^{0}	1221	2114	2114	175,176	2114	2114
Δ^{-}	2221	1114	1114	195,196	1114	1114

Strange Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Λ	2130	3122	3122	121,122	3122	3122
Σ^{+}	1130	3222	3222	139,140	3222	3222
Σ^{0}	1230	3212	3212	129,130	3212	3212
Σ^{-}	2230	3112	3112	145,146	3112	3112
$\Sigma^{*+}/\Sigma(1385)^{+}$	1131	3224	3224	171,172	3224	3224
$\Sigma^{*0}/\Sigma(1385)^{0}$	1231	3214	3214	161,162	3214	3214
$\Sigma^{*-}/\Sigma(1385)^{-}$	2231	3114	3114	177,178	3114	3114
Ξ^{0}	1330	3322	3322	149,150	3322	3322
Ξ^{-}	2330	3312	3312	151,152	3312	3312
$\Xi^{*0}/\Xi(1530)^{0}$	1331	3324	3324	181,182	3324	3324
$\Xi^{*-}/\Xi(1530)^{-}$	2331	3314	3314	183,184	3314	3314
Ω^{-}	3331	3334	3334	197,198	3334	3334

Charmed Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Λ_c^+	2140	4122	4122	123,124	4122	4122
Σ_c^{++}	1140	4222	4222	141,142	4222	4222
Σ_c^+	1240	4212	4212	131,132	4212	4212
Σ_c^0	2240	4112	4112	147,148	4112	4112
Σ_c^{*++}	1141	4224	4224	173,174	4224	4224
Σ_c^{*+}	1241	4214	4214	163,164	4214	4214
Σ_c^{*0}	2241	4114	4114	179,180	4114	4114
Ξ_c^+	3140	4232	4232	125,126	4232	4232
Ξ_c^0	3240	4132	4132	127,128	4132	4132
$\Xi_c'^+$	1340	4322	4322	133,134	4322	4322
$\Xi_c'^0$	2340	4312	4312	135,136	4312	4312
Ξ_c^{*+}	1341	4324	4324	165,166	4324	4324
Ξ_c^{*0}	2341	4314	4314	167,168	4314	4314
Ω_c^0	3340	4332	4332	153,154	4332	4332
Ω_c^{*0}	3341	4334	4334	185,186	4334	4334
Ξ_{cc}^+	2440	4412		157,158	4412	4412
Ξ_{cc}^{++}	1440	4422		155,156	4422	4422
Ξ_{cc}^{*+}	2441	4414		189,190	4414	4414
Ξ_{cc}^{*++}	1441	4424		187,188	4424	4424
Ω_{cc}^+	3440	4432		159,160	4432	4432
Ω_{cc}^{*+}	3441	4434		191,192	4434	4434
Ω_{ccc}^{*++}	4441	4444		199,200	4444	4444

Bottom Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Λ_b^0	2150	5122	5122	401,402	5122	5122
Σ_b^-	2250	5112	5112	427,428	5112	5112
Σ_b^0	1250	5212	5212	413,414	5212	5212
Σ_b^+	1150	5222	5222	425,426	5222	5222
Σ_b^{*-}	2251	5114	5114	455,456	5114	5114
Σ_b^{*0}	1251	5214	5214	441,442	5214	5214
Σ_b^{*+}	1151	5224	5224	453,454	5224	5224
Ξ_b^-	3250	5132	5132	405,406	5132	5132
Ξ_b^0	3150	5232	5232	403,404	5232	5232
$\Xi_b^{\prime -}$	2350	5312	5312	417,418	5312	5312
$\Xi_b^{\prime 0}$	1350	5322	5322	415,416	5322	5322
Ξ_b^{*-}	2351	5314	5314	445,446	5314	5314
Ξ_b^{*0}	1351	5324	5324	443,444	5324	5324
Ω_b^-	3350	5332	5332	429,430	5332	5332
Ω_b^{*-}	3351	5334	5334	457,458	5334	5334
Ξ_{bc}^0	4250	5142		409,410	5142	5142
Ξ_{bc}^+	4150	5242		407,408	5242	5242
$\Xi_{bc}^{\prime 0}$	2450	5412		421,422	5412	5412
$\Xi_{bc}^{\prime +}$	1450	5422		419,420	5422	5422
Ξ_{bc}^{*0}	2451	5414		449,450	5414	5414
Ξ_{bc}^{*+}	1451	5424		447,448	5424	5424
Ω_{bc}^0	4350	5342		411,412	5342	5342
$\Omega_{bc}^{\prime 0}$	3450	5432		423,424	5432	5432
Ω_{bc}^{*0}	3451	5434		451,452	5434	5434
Ω_{bcc}^+	4450	5442		431,432	5442	5442
Ω_{bcc}^{*+}	4451	5444		459,460	5444	5444

Bottom Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Ξ_{bb}^{-}	2550	5512		435,436	5512	5512
Ξ_{bb}^{0}	1550	5522		433,434	5522	5522
Ξ_{bb}^{*-}	2551	5514		463,464	5514	5514
Ξ_{bb}^{*0}	1551	5524		461,462	5524	5524
Ω_{bb}^{-}	3550	5532		437,438	5532	5532
Ω_{bb}^{*-}	3551	5534		465,466	5534	5534
Ω_{bbc}^{0}	4550	5542		439,440	5542	5542
Ω_{bbc}^{*0}	4551	5544		467,468	5544	5544
Ω_{bbb}^{*-}	5551	5554		469,470	5554	5554

Top Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Λ_t^+	2160		6122			6122
Σ_t^0	2260		6112			6112
Σ_t^+	1260					6212
Σ_t^{++}	1160		6222			6222
Σ_t^{*0}	2261					6114
Σ_t^{*+}	1261					6214
Σ_t^{*++}	1161					6224
Ξ_t^0	3260		6132			6132
Ξ_t^+	3160		6232			6232
$\Xi_t'^0$	2360					6312
$\Xi_t'^+$	1360					6322
Ξ_t^{*0}	2361					6314
Ξ_t^{*+}	1361					6324
Ω_t^0	3360		6332			6332
Ω_t^{*0}	3361					6334
Ξ_{tc}^+	4260					6142
Ξ_{tc}^{++}	4160					6242
$\Xi_{tc}'^+$	2460					6412
$\Xi_{tc}'^{++}$	1460					6422
Ξ_{tc}^{*+}	2461					6414
Ξ_{tc}^{*++}	1461					6424
Ω_{tc}^+	4360					6342
$\Omega_{tc}'^+$	3460					6432
Ω_{tc}^{*+}	3461					6434
Ω_{tcc}^{++}	4460					6442
$\Omega_{tcc*}{}^{++}$	4461					6444

Top Baryons						
Baryon	Isajet 7.72	Pythia 6.413	Herwig 6.510	QQ 9.2b	PDG 2006	StdHep 5.06
Ξ_{tb}^{0}	5260					6152
Ξ_{tb}^{+}	5160					6252
$\Xi_{tb}^{\prime 0}$	2560					6512
$\Xi_{tb}^{\prime +}$	1560					6522
Ξ_{tb}^{*0}	2561					6514
Ξ_{tb}^{*+}	1561					6524
Ω_{tb}^{0}	5360					6352
$\Omega_{tb}^{\prime 0}$	3560					6532
Ω_{tb}^{*0}	3561					6534
Ω_{tbc}^{+}	5460					6452
$\Omega_{tbc}^{\prime +}$	4560					6542
Ω_{tbc}^{*+}	4561					6544
Ω_{tbb}^{0}	5560					6552
Ω_{tbb}^{*0}	5561					6554
Ξ_{tt}^{+}	2660					6612
Ξ_{tt}^{++}	1660					6622
Ξ_{tt}^{*+}	2661					6614
Ξ_{tt}^{*++}	1661					6624
Ω_{tt}^{+}	3660					6632
Ω_{tt}^{*+}	3661					6634
Ω_{ttc}^{++}	4660					6642
Ω_{ttc}^{*++}	4661					6644
Ω_{ttb}^{+}	5660					6652
Ω_{ttb}^{*+}	5661					6654
Ω_{ttt}^{*++}	6661					6664

D Known Values of ISTHEP

ISTHEP			
Isajet 7.51	Pythia 6.319	Herwig 6.505	QQ 9.2b
	0		
1	1	1	1
2	2	2	2
	3	3	
11-13			
21-23			
	31-100		
		100-103	
		110-115	
		120-125	
		130-175	
		181-186	
		190-193	
		195-200	