StdHepC++

A common Monte Carlo Interface

Lynn Garren

AIHENP'99

http://www–pat.fnal.gov/stdhep/c++/aihenp99.ps

Fermilab

In the beginning:

- Monte Carlo generators all have their own common blocks
- users need different analysis code

PDG Monte Carlo numbering scheme I

HEPEVT common block

StdHep

- numbering scheme had too much room for interpretation
- only some generators used HEPEVT
- useful utilities

PDG Monte Carlo numbering scheme II

C++ standard particle and event classes

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

StdHep interfaces

- **Pythia**
- **Isajet**
- **Herwig**
- **QQ**
- **Geant3**
- **PDG**

Authors

Lynn Garren – StdHep interfaces and utilities

Paul Lebrun – I/O package

PURPOSE

A standard event class library will be created to allow users a common interface to Monte Carlo event generators.

Work is ongoing by several people to create C++ Monte Carlo generators.   This class is intended as a common interface for all commonly used generators. It is consistent with the Pythia 7 strategies outlined at

http://www.thep.lu.se/tf2/staff/leif/Pythia7/Welcome.html

As a first step, an interface to the existing fortran HEPEVT common block will be created and maintained. As generators evolve, interfaces to specific generators may no longer be necessary.

The StdHep C++ class library must implement all functions currently in the fortran stdhep implementation.

XDR, LHC++, and root I/O will be supported.

METHODS

Methods will be available to implement all functions and subroutines currently in the fortran implementation of stdhep.

Some of the current functionality is:

- I/O via XDR with MCFio.
- Charge, name, and quark content of a particle.
- Lists of descendants and parents.
- Translation from Monte Carlo generator output to and from stdhep.
- Translation of particle ID codes to and from the standard PDG set.

A method will be available for moving to and from the HEPEVT common block.

MULTIPLE COLLISION INFO

```
C ----------------------------------------------
C
      common/hepev2/nmulti,jmulti(NMXHEP)
      integer nmulti,jmulti

C... nmulti - number of interactions in the list
C... jmulti(..)   - multiple interaction number
C
C ---------------------------------------------
      integer NMXMLT
      parameter (NMXMLT=16)
      common/hepev3/nevmulti(NMXMLT),
     *     itrkmulti(NMXMLT), mltstr(NMXMLT)
      integer nevmulti,itrkmulti,mltstr

C... nevmulti(i) - event number of original
C                  interaction
C... itrkmulti(i) - first particle in the
C                   original interaction
C... mltstr(i)    - stream this event is from
C
C ---------------------------------------------
```

HEPEVT COMMON BLOCK

```
C ------------------------------------
C
      integer NMXHEP
      parameter (NMXHEP=4000)
      common/HEPEVT/NEVHEP, NHEP,
     *       ISTHEP(NMXHEP), IDHEP(NMXHEP),
     *       JMOHEP(2,NMXHEP), JDAHEP(2,NMXHEP),
     *       PHEP(5,NMXHEP), VHEP(4,NMXHEP)
      integer NEVHEP, NHEP, ISTHEP, IDHEP,
     *          JMOHEP, JDAHEP
      double precision PHEP, VHEP

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

StdHep::Event

list of events

list of particles

event number

number of particles in event

originating event stream

→ intrinsic multiple collisions

```
//
// StdEvent.hh
//
#ifndef STDEVENT_HH
#define STDEVENT_HH

#include <vector>

#include "StdParticle.hh"

namespace StdHep  {

class Event {

public:

  // ---  birth/death:
  //
  Event( int eventNumber, int nEntries );
  Event( Event const & orig );
  ~Event();

  // ---  accessors:
  //
  int                        const    eventNumber()   const;
  int                        const    nEntries()      const;

  StdHep::Particle HEParticle( int i ) const;

  // ---  mutators:
  //
  void  seteventNumber ( int nmbr );
  void  setnEntries    ( int ntries );
  void  appendParticle ( StdHep::Particle* p );

private:

  int itseventNumber;
  int itsnEntries;
  vector<StdHep::Particle*> itsHEParticles;

};       // Event

}
#endif // STDEVENT_HH
```

```
//
// StdEvent.cc
//

#include "StdEvent.hh"

namespace StdHep  {

Event::Event( int eventNumber, int nEntries )
: itseventNumber ( eventNumber ),
  itsnEntries    ( nEntries )
{ ; }

Event::Event( Event const & orig )
: itseventNumber ( orig.itseventNumber ),
  itsnEntries    ( orig.itsnEntries )
{ ; }

Event::~Event()  { ; }

int const  Event::eventNumber() const  {  return
itseventNumber; }

int const  Event::nEntries() const  {  return itsnEntries; }

StdHep::Particle Event::HEParticle( int i ) const {
    return *(itsHEParticles[i]);
}

void  Event::seteventNumber( int nmbr )  {
   itseventNumber = nmbr;
}

void  Event::setnEntries( int ntries )  {
   itsnEntries = ntries;
}

void  Event::appendParticle ( StdHep::Particle* p ) {
    itsHEParticles.push_back( p );
}

}  // StdHep
```

StdHep::Particle

PDG number

ParticleData

status

list of mothers

list of daughters

4–momentum

generated mass

creation vertex (space & time)

decay vertex (space & time)

```
//
// StdParticle.hh
//
#ifndef STDPARTICLE_HH
#define STDPARTICLE_HH

#include "CLHEP/Vector/ThreeVector.h"
#include "CLHEP/Vector/LorentzVector.h"

namespace StdHep {

class Particle {

public:

  // ---  birth/death:
  //
   Particle( int ParticleID, int StatusCode, int mother,
            int secondMother, int daughtersBegin,
            int daughtersEnd, HepLorentzVector pHep,
            double genmass, HepLorentzVector CreationVertex,
            HepLorentzVector DecayVertex );
  Particle( Particle const & orig );
  ~Particle();

  // ---  accessors:
  //
  int const              ParticleID()     const;
  int const              StatusCode()     const;
  int const              mother()         const;
  int const              secondMother()   const;
  int const              daughtersBegin() const;
  int const              daughtersEnd()   const;
  HepLorentzVector const pHep()           const;
  double const           genmass()        const;
  HepLorentzVector const CreationVertex() const;
  HepLorentzVector const DecayVertex()    const;


/// continued on next page
```

```
//
// StdParticle.hh - continued
//

  // ---  mutators:
  //
  void  setParticleID      ( int pid );
  void  setStatusCode      ( int status );
  void  setmother          ( int mth );
  void  setsecondMother    ( int mth2 );
  void  setdaughtersBegin  ( int dtr );
  void  setdaughtersEnd    ( int dtr2 );
  void  setpHep            ( HepDouble px, HepDouble py,
                             HepDouble pz, HepDouble E );
  void  setgenmass         ( double mass );
  void  setCreationVertex  ( HepDouble cx, HepDouble cy,
                             HepDouble cz, HepDouble ct );
  void  setDecayVertex     ( HepDouble dx, HepDouble dy,
                             HepDouble dz, HepDouble dt );

private:

  int itsParticleID;

  int itsStatusCode;

  int itsmother;
  int itssecondMother;
  int itsdaughtersBegin;
  int itsdaughtersEnd;

  HepLorentzVector itspHep;
  double           itsgenmass;    // (in GeV/c**2)

  HepLorentzVector itsCreationVertex;
  HepLorentzVector itsDecayVertex;

};      // Particle

}
#endif // STDPARTICLE_HH
```

```
//
// StdParticle.cc
//

#include "StdParticle.hh"

namespace StdHep  {

Particle::Particle( int ParticleID, int StatusCode,
            int mother, int secondMother,
            int daughtersBegin, int daughtersEnd,
            HepLorentzVector pHep,
            double genmass, HepLorentzVector CreationVertex,
            HepLorentzVector DecayVertex )
: itsParticleID      ( ParticleID ),
  itsStatusCode      ( StatusCode ),
  itsmother          ( mother ),
  itssecondMother    ( secondMother ),
  itsdaughtersBegin  ( daughtersBegin ),
  itsdaughtersEnd    ( daughtersEnd ),
  itspHep            ( pHep ),
  itsgenmass         ( genmass ),
  itsCreationVertex  ( CreationVertex ),
  itsDecayVertex     ( DecayVertex )
{ ; }

Particle::Particle( Particle const & orig )
: itsParticleID      ( orig.itsParticleID ),
  itsStatusCode      ( orig.itsStatusCode ),
  itsmother          ( orig.itsmother ),
  itssecondMother    ( orig.itssecondMother ),
  itsdaughtersBegin  ( orig.itsdaughtersBegin ),
  itsdaughtersEnd    ( orig.itsdaughtersEnd ),
  itspHep            ( orig.itspHep ),
  itsgenmass         ( orig.itsgenmass ),
  itsCreationVertex  ( orig.itsCreationVertex ),
  itsDecayVertex     ( orig.itsDecayVertex )
{ ; }

Particle::~Particle()  { ; }

/// continued on next page
```

```
//
// StdParticle.cc  - continuation 1
//

// ---  accessors:
//
int     const Particle::ParticleID()     const {
   return itsParticleID;
}
int     const Particle::StatusCode()     const {
   return itsStatusCode;
}
int     const Particle::mother()         const {
   return itsmother;
}
int     const Particle::secondMother()   const {
   return itssecondMother;
}
int     const Particle::daughtersBegin() const {
   return itsdaughtersBegin;
}
int     const Particle::daughtersEnd()   const {
   return itsdaughtersEnd;
}
HepLorentzVector const Particle::pHep()  const {
   return itspHep;
}
double   const Particle::genmass()       const {
   return itsgenmass;
}
HepLorentzVector const Particle::CreationVertex() const {
   return itsCreationVertex;
}
HepLorentzVector const Particle::DecayVertex() const {
   return itsDecayVertex;
}

/// continued on next page
```

```
//
// StdParticle.cc - continuation 2
//


// ---  mutators:
//
void Particle::setParticleID( int pid ) {
   itsParticleID = pid; }
void Particle::setStatusCode( int status ) {
   itsStatusCode = status; }
void Particle::setmother( int mth ) { itsmother = mth; }
void Particle::setsecondMother( int mth2 ) {
   itssecondMother = mth2; }
void Particle::setdaughtersBegin( int dtr ) {
   itsdaughtersBegin = dtr; }
void Particle::setdaughtersEnd( int dtr2 ) {
   itsdaughtersEnd = dtr2; }
void Particle::setpHep( HepDouble px, HepDouble py,
                        HepDouble pz, HepDouble E ) {
  itspHep.setPx( px );
  itspHep.setPy( py );
  itspHep.setPz( pz );
  itspHep.setE( E );
}
void  Particle::setgenmass( double mass )  {
   itsgenmass = mass; }
void  Particle::setCreationVertex( HepDouble cx,
             HepDouble cy, HepDouble cz, HepDouble ct ) {
  itsCreationVertex.setX( cx );
  itsCreationVertex.setY( cy );
  itsCreationVertex.setZ( cz );
  itsCreationVertex.setT( ct );
}
void  Particle::setDecayVertex( HepDouble dx, HepDouble dy,
                              HepDouble dz, HepDouble dt ) {
  itsDecayVertex.setX( dx );
  itsDecayVertex.setY( dy );
  itsDecayVertex.setZ( dz );
  itsDecayVertex.setT( dt );
}

}  // StdHep
```

StdHep::ParticleData

PDG particle number

name as a character string

PDG mass

fixed information about this type of particle

might have decay information

```
//
// StdParticleData.hh
//


#include <string>

namespace StdHep  {

class ParticleData  {

public:

  // ---  birth/death:
  //
  ParticleData( int id, std::string const & name,
                double PDGmass );
  ParticleData( ParticleData const & orig );
  ~ParticleData();

  // ---  accessors:
  //
  int const              id()      const;
  std::string const &    name()    const;
  double const           PDGmass() const;

  // ---  mutators:
  //
  void  setPDGmass( double mass );

private:

  // ---  class-specific data:
  //
  int           itsId;
  std::string   itsName;
  double        itsPDGmass;

  // ---  forbidden operations:
  //
  ParticleData &  operator=( ParticleData const & );

};  // ParticleData

}  // StdHep
```

```
//
// StdParticleData.cc
//


#include "StdParticleData.hh"

namespace StdHep  {

ParticleData::ParticleData( int id,
                std::string const & name, double PDGmass )
: itsId     ( id ),
  itsName   ( name ),
  itsPDGmass( PDGmass )
{ ; }

ParticleData::ParticleData( ParticleData const & orig )
: itsId     ( orig.itsId ),
  itsName   ( orig.itsName ),
  itsPDGmass( orig.itsPDGmass )
{ ; }

ParticleData::~ParticleData()  { ; }

int const  ParticleData::id() const  {  return itsId; }

std::string const &  ParticleData::name() const  {
   return itsName;
}

double const  ParticleData::PDGmass() const  {
   return itsPDGmass;
}

void  ParticleData::setPDGmass( double mass )  {
   itsPDGmass = mass;
}

}  // StdHep
```

CONCLUSION

intended for inclusion in CLHEP

discussions with

- CLHEP
- CDF
- D0
- BaBar
- (Herwig)
- (Isajet)
- Leif Lonnblad (Pythia 7)
- QQ
- PDG

code writing in progress

http://www–pat.fnal.gov/stdhep/c++/

stdhep–cpp@fnal.gov

garren@fnal.gov

31. MONTE CARLO PARTICLE NUMBERING SCHEME

Revised April 1998 by L. Garren (Fermilab), I.G. Knowles (Edinburgh U.), T. Sjöstrand (Lund U.), and T. Trippe (LBNL).

The PDG particle numbering scheme [1] is designed to facilitate interfacing between the event generator and analysis packages used in particle physics. It is used in several generators, *e.g.* HERWIG and PYTHIA/JETSET, and in the /HEPEVT/ [2] standard interface. After consultation [3], the scheme has been revised to better match the practice of program authors. The revised scheme includes numbering of states by orbital and radial quantum numbers to allow systematic inclusion of quark model states which are as yet undiscovered, and also includes numbering for hypothetical particles such as SUSY particles. The general form is a 7–digit number:

$$\pm n\, n_r\, n_L\, n_{q_1}\, n_{q_2}\, n_{q_3}\, n_J\ .$$

This encodes information about the particle's spin, flavor content, and internal quantum numbers. The details are as follows:

1. Particles are given positive numbers, antiparticles negative numbers. The PDG convention for mesons is used, so that K^+ and B^+ are particles.
2. Quarks and leptons are numbered consecutively starting from 1 and 11 respectively; to do this they are first ordered by family and within families by weak isospin.
3. In composite quark systems (diquarks, mesons, and baryons) $n_{q_{1-3}}$ are quark numbers used to specify the quark content, while the rightmost digit $n_J = 2J + 1$ gives the system's spin (except for the K^0_S and K^0_L). The scheme does not cover particles of spin $J > 4$.
4. Diquarks have 4-digit numbers with $n_{q_1} \geq n_{q_2}$ and $n_{q_3} = 0$.
5. The numbering of mesons is guided by the nonrelativistic (L–S decoupled) quark model, as listed in Table 13.2. This leads to several differences with the earlier numbering [4] for excited mesons.
   a. The numbers specifying the meson's quark content conform to the convention $n_{q_1} = 0$ and $n_{q_2} \geq n_{q_3}$. The special case K^0_L is the sole exception to this rule.
   b. The quark numbers of flavorless, light (u, d, s) mesons are: 11 for the member of the isotriplet ($\pi^0, \rho^0, \ldots$), 22 for the lighter isosinglet ($\eta, \omega, \ldots$), and 33 for the heavier isosinglet ($\eta', \phi, \ldots$). Since isosinglet mesons are often large mixtures of $u\overline{u} + d\overline{d}$ and $s\overline{s}$ states, 22 and 33 are assigned by mass and do not necessarily specify the dominant quark composition.
   c. The special numbers 310 and 130 are given to the K^0_S and K^0_L respectively.
   d. 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. For $J > 0$ the numbers are: $(L, S) = (J-1, 1)$ $n_L = 0$, $(J, 0)$ $n_L = 1$, $(J, 1)$ $n_L = 2$ and $(J+1, 1)$ $n_L = 3$. For the exceptional case $J = 0$ the numbers are $(0, 0)$ $n_L = 0$ and $(1, 1)$ $n_L = 1$ (*i.e.* $n_L = L$). See Table 31.1.

Table 31.1: Meson numbering logic. Here qq stands for $n_{q_2} n_{q_3}$.

	$L = J-1, S = 1$			$L = J, S = 0$			$L = J, S = 1$			$L = J+1, S = 1$		
J	code	J^{PC}	L	code	J^{PC}	L	code	J^{PC}	L	code	J^{PC}	L
0	—	—	—	00qq1	0^{-+}	0	—	—	—	10qq1	0^{++}	1
1	00qq3	1^{--}	0	10qq3	1^{+-}	1	20qq3	1^{++}	1	30qq3	1^{--}	2
2	00qq5	2^{++}	1	10qq5	2^{-+}	2	20qq5	2^{--}	2	30qq5	2^{++}	3
3	00qq7	3^{--}	2	10qq7	3^{+-}	3	20qq7	3^{++}	3	30qq7	3^{--}	4
4	00qq9	4^{++}	3	10qq9	4^{-+}	4	20qq9	4^{--}	4	30qq9	4^{++}	5

   e. If a set of physical mesons correspond to a (non-negligible) mixture of basis states, differing in their internal quantum numbers, then the lightest physical state gets the smallest basis state number. For example the $K_1(1270)$ is numbered 10313 (1^1P_1 K_{1B}) and the $K_1(1400)$ is numbered 20313 (1^3P_1 K_{1A}).
   f. The sixth digit n_r is used to label mesons radially excited above the ground state.
   g. Numbers have been assigned for complete $n_r = 0$ S- and P-wave multiplets, even where states remain to be identified.
   h. In some instances assignments within the $q\overline{q}$ meson model are only tentative; here best guess assignments are made.
   i. Many states appearing in the Meson Listings are not yet assigned within the $q\overline{q}$ model. Here $n_{q_{2-3}}$ and n_J are assigned according to the state's likely flavors and spin; all such unassigned light isoscalar states are given the flavor code 22. Within these groups $n_L = 0, 1, 2, \ldots$ is used to distinguish states of increasing mass. These states are flagged using $n = 9$. It is to be expected that these numbers will evolve as the nature of the states are elucidated.
6. The numbering of baryons is again guided by the nonrelativistic quark model, see Table 13.4.
   a. The numbers specifying a baryon's quark content are such that in general $n_{q_1} \geq n_{q_2} \geq n_{q_3}$.
   b. Two states exist for $J = 1/2$ baryons containing 3 different types of quarks. In the lighter baryon ($\Lambda, \Xi, \Omega, \ldots$) the light quarks are in an antisymmetric ($J = 0$) state while for the heavier baryon ($\Sigma^0, \Xi', \Omega', \ldots$) they are in a symmetric ($J = 1$) state. In this situation n_{q_2} and n_{q_3} are reversed for the lighter state, so that the smaller number corresponds to the lighter baryon.
   c. At present most Monte Carlos do not include excited baryons and no systematic scheme has been developed to denote them, though one is foreseen. In the meantime, use of the PDG 96 [4] numbers for excited baryons is recommended.
7. The gluon, when considered as a gauge boson, has official number 21. In codes for glueballs, however, 9 is used to allow a notation in close analogy with that of hadrons.
8. The pomeron and odderon trajectories and a generic reggeon trajectory of states in QCD are assigned codes 990, 9990, and 110 respectively, where the final 0 indicates the indeterminate nature of the spin, and the other digits reflect the expected "valence" flavor content. We do not attempt a complete classification of all reggeon trajectories, since there is currently no need to distinguish a specific such trajectory from its lowest-lying member.
9. Two-digit numbers in the range 21–30 are provided for the Standard Model gauge bosons and Higgs.
10. Codes 81–100 are reserved for generator-specific pseudoparticles and concepts.
11. The search for physics beyond the Standard Model is an active area, so these codes are also standardized as far as possible.
    a. A standard fourth generation of fermions is included by analogy with the first three.
    b. The graviton and the boson content of a two-Higgs-doublet scenario and of additional SU(2)×U(1) groups are found in the range 31–40.
    c. "One-of-a-kind" exotic particles are assigned numbers in the range 41–80.
    d. Fundamental supersymmetric particles are identified by adding a nonzero n to the particle number. The superpartner of a boson or a left-handed fermion has $n = 1$ while the superpartner of a right-handed fermion has $n = 2$. When mixing occurs, such as between the winos and charged Higgsinos to give charginos, or between left and right sfermions, the lighter physical state is given the smaller basis state number.
    e. Technicolor states have $n = 3$. In the absence of a unique theory we only number generic states whose digits reflect the techniquark content.
    f. Excited (composite) quarks and leptons are identified by setting $n = 4$.

12. Occasionally program authors add their own states. To avoid confusion, these should be flagged by setting $nn_r = 99$.
13. Concerning the non-99 numbers, it may be noted that only quarks, excited quarks, squarks, and diquarks have $n_{q3} = 0$; only diquarks, baryons, and the odderon have $n_{q1} \neq 0$; and only mesons, the reggeon, and the pomeron have $n_{q1} = 0$ and $n_{q2} \neq 0$. Concerning mesons (not antimesons), if n_{q1} is odd then it labels a quark and an antiquark if even.

This text and lists of particle numbers can be found on the WWW [5]. The StdHep Monte Carlo standardization project [6] maintains the list of PDG particle numbers, as well as numbering schemes from most event generators and software to convert between the different schemes.

References:

1. 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 (November 1987). Presented at the Workshop on Detector Simulation for the SSC (August 1987);
   G.P. Yost *et al.*, Particle Data Group, Phys. Lett. **B204**, 1 (1988).
2. T. Sjöstrand *et al.*, in *"Z physics at LEP1"*, CERN 89-08, vol. 3, p. 327.
3. I. G. Knowles *et al.*, in *"Physics at LEP2"*, CERN 96-01, vol. 2, p. 103.
4. R.M. Barnett *et al.*, Particle Data Group, Phys. Rev. **D54**, 1 (1996).
5. `http://pdg.lbl.gov/mc_particle_id_contents.html`.
6. L. Garren, StdHep 3.01, *Monte Carlo Standardization at FNAL*, Fermilab PM0091 (Nov. 17, 1995) and StdHep WWW site: `http://www-pat.fnal.gov/stdhep.html`.

QUARKS

d	1
u	2
s	3
c	4
b	5
t	6
b'	7
t'	8

LEPTONS

e^-	11
ν_e	12
μ^-	13
ν_μ	14
τ^-	15
ν_τ	16
τ'^-	17
$\nu_{\tau'}$	18

EXCITED PARTICLES

d^*	4000001
u^*	4000002
e^*	4000011
ν_e^*	4000012

GAUGE AND HIGGS BOSONS

g	(9) 21
γ	22
Z^0	23
W^+	24
h^0/H_1^0	25
Z'/Z_2^0	32
Z''/Z_3^0	33
W'/W_2^+	34
H^0/H_2^0	35
A^0/H_3^0	36
H^+	37

DIQUARKS

$(dd)_1$	1103
$(ud)_0$	2101
$(ud)_1$	2103
$(uu)_1$	2203
$(sd)_0$	3101
$(sd)_1$	3103
$(su)_0$	3201
$(su)_1$	3203
$(ss)_1$	3303
$(cd)_0$	4101
$(cd)_1$	4103
$(cu)_0$	4201
$(cu)_1$	4203
$(cs)_0$	4301
$(cs)_1$	4303
$(cc)_1$	4403
$(bd)_0$	5101
$(bd)_1$	5103
$(bu)_0$	5201
$(bu)_1$	5203
$(bs)_0$	5301
$(bs)_1$	5303
$(bc)_0$	5401
$(bc)_1$	5403
$(bb)_1$	5503

TECHNICOLOR PARTICLES

$\pi^0_{\rm tech}$	3000111
$\pi^+_{\rm tech}$	3000211
$\eta^0_{\rm tech}$	3000221
$\rho^0_{\rm tech}$	3000113
$\rho^+_{\rm tech}$	3000213
$\omega^0_{\rm tech}$	3000223

SUSY PARTICLES

$\tilde{d}_L$	1000001
$\tilde{u}_L$	1000002
$\tilde{s}_L$	1000003
$\tilde{c}_L$	1000004
$\tilde{b}_1$	1000005^a
$\tilde{t}_1$	1000006^a
$\tilde{e}_L^-$	1000011
$\tilde{\nu}_{eL}$	1000012
$\tilde{\mu}_L^-$	1000013
$\tilde{\nu}_{\mu L}$	1000014
$\tilde{\tau}_1^-$	1000015^a
$\tilde{\nu}_{\tau L}$	1000016
$\tilde{d}_R$	2000001
$\tilde{u}_R$	2000002
$\tilde{s}_R$	2000003
$\tilde{c}_R$	2000004
$\tilde{b}_2$	2000005^a
$\tilde{t}_2$	2000006^a
$\tilde{e}_R^-$	2000011
$\tilde{\mu}_R^-$	2000013
$\tilde{\tau}_2^-$	2000015^a
$\tilde{g}$	1000021
$\tilde{\chi}_1^0$	1000022^b
$\tilde{\chi}_2^0$	1000023^b
$\tilde{\chi}_1^+$	1000024^b
$\tilde{\chi}_3^0$	1000025^b
$\tilde{\chi}_4^0$	1000035^b
$\tilde{\chi}_2^+$	1000037^b
$\tilde{G}$	1000039

SPECIAL PARTICLES

G (graviton)	39
R^0	41
LQ^c	42
reggeon	110
pomeron	990
odderon	9990

for MC internal use 81–100

LIGHT $I = 1$ MESONS

π^0	111
π^+	211
$a_0(980)^0$	**9000111***
$a_0(980)^+$	**9000211***
$\pi(1300)^0$	**100111***
$\pi(1300)^+$	**100211***
$a_0(1450)^0$	**10111***
$a_0(1450)^+$	**10211***
$\pi(1800)^0$	200111
$\pi(1800)^+$	200211
$\rho(770)^0$	113
$\rho(770)^+$	213
$b_1(1235)^0$	10113
$b_1(1235)^+$	10213
$a_1(1260)^0$	20113
$a_1(1260)^+$	20213
$\hat{\rho}(1405)^0$	9000113
$\hat{\rho}(1405)^+$	9000213
$\rho(1450)^0$	**100113***
$\rho(1450)^+$	**100213***
$\rho(1700)^0$	30113
$\rho(1700)^+$	30213
$\rho(2150)^0$	9010113
$\rho(2150)^+$	9010213
$a_2(1320)^0$	115
$a_2(1320)^+$	215
$\pi_2(1670)^0$	10115
$\pi_2(1670)^+$	10215
$\pi_2(2100)^0$	9000115
$\pi_2(2100)^+$	9000215
$\rho_3(1690)^0$	117
$\rho_3(1690)^+$	217
$\rho_3(2250)^0$	9000117
$\rho_3(2250)^+$	9000217
$a_4(2040)^0$	119
$a_4(2040)^+$	219

LIGHT $I = 0$ MESONS
($u\overline{u}$, $d\overline{d}$, and $s\overline{s}$ Admixtures)

η	221
$\eta'(958)$	331
$f_0(400–1200)$	**9000221***
$f_0(980)$	**9010221***
$\eta(1295)$	**100221***
$f_0(1370)$	**10221***
$\eta(1440)$	**100331***
$f_0(1500)$	**9020221***
$f_J(1710)$	**9030221***
$\eta(1760)$	200221
$f_0(2020)$	9040221
$f_0(2060)$	9050221
$f_0(2200)$	**9060221***
$\eta(2225)$	**9070221***
$\omega(782)$	223
$\phi(1020)$	333
$h_1(1170)$	10223
$f_1(1285)$	20223
$h_1(1380)$	10333
$f_1(1420)$	**20333***
$\omega(1420)$	**100223***
$f_1(1510)$	**9000223***
$\omega(1600)$	**30223***
$\phi(1680)$	**100333***
$f_2(1270)$	225
$f_2(1430)$	9000225
$f_2'(1525)$	335
$f_2(1565)$	9010225
$f_2(1640)$	9020225
$\eta_2(1645)$	10225
$f_2(1810)$	100225
$\eta_2(1870)$	**10335***
$f_2(1950)$	**9030225***
$f_2(2010)$	**100335***
$f_2(2150)$	**9040225***
$f_2(2300)$	**9050225***
$f_2(2340)$	**9060225***
$\omega_3(1670)$	227
$\phi_3(1850)$	337
$f_4(2050)$	229
$f_J(2220)$	9000339
$f_4(2300)$	9000229

STRANGE MESONS

K_L^0	130
K_S^0	310
K^0	311
K^+	321
$K_0^*(1430)^0$	10311
$K_0^*(1430)^+$	10321
$K(1460)^0$	100311
$K(1460)^+$	100321
$K(1830)^0$	200311
$K(1830)^+$	200321
$K_0^*(1950)^0$	9000311
$K_0^*(1950)^+$	9000321
$K^*(892)^0$	313
$K^*(892)^+$	323
$K_1(1270)^0$	10313
$K_1(1270)^+$	10323
$K_1(1400)^0$	20313
$K_1(1400)^+$	20323
$K^*(1410)^0$	**100313***
$K^*(1410)^+$	**100323***
$K_1(1650)^0$	9000313
$K_1(1650)^+$	9000323
$K^*(1680)^0$	**30313***
$K^*(1680)^+$	**30323***
$K_2^*(1430)^0$	315
$K_2^*(1430)^+$	325
$K_2(1580)^0$	9000315
$K_2(1580)^+$	9000325
$K_2(1770)^0$	10315
$K_2(1770)^+$	10325
$K_2(1820)^0$	20315
$K_2(1820)^+$	20325
$K_2^*(1980)^0$	100315
$K_2^*(1980)^+$	100325
$K_2(2250)^0$	9010315
$K_2(2250)^+$	9010325
$K_3^*(1780)^0$	317
$K_3^*(1780)^+$	327
$K_3(2320)^0$	9010317
$K_3(2320)^+$	9010327
$K_4^*(2045)^0$	319
$K_4^*(2045)^+$	329
$K_4(2500)^0$	9000319
$K_4(2500)^+$	9000329

CHARMED MESONS

D^+	411
D^0	421
D_0^{*+}	10411
D_0^{*0}	10421
$D^*(2010)^+$	413
$D^*(2007)^0$	423
$D_1(2420)^+$	10413
$D_1(2420)^0$	10423
$D_1(H)^+$	20413
$D_1(H)^0$	20423
$D_2^*(2460)^+$	415
$D_2^*(2460)^0$	425
D_s^+	431
D_{s0}^{*+}	10431
D_s^{*+}	433
$D_{s1}(2536)^+$	10433
$D_{s1}(H)^+$	20433
D_{s2}^{*+}	435

BOTTOM MESONS

B^0	511
B^+	521
B_0^{*0}	10511
B_0^{*+}	10521
B^{*0}	513
B^{*+}	523
$B_1(L)^0$	10513
$B_1(L)^+$	10523
$B_1(H)^0$	20513
$B_1(H)^+$	20523
B_2^{*0}	515
B_2^{*+}	525
B_s^0	531
B_{s0}^{*0}	10531
B_s^{*0}	533
$B_{s1}(L)^0$	10533
$B_{s1}(H)^0$	20533
B_{s2}^{*0}	535
B_c^+	541
B_{c0}^{*+}	10541
B_c^{*+}	543
$B_{c1}(L)^+$	10543
$B_{c1}(H)^+$	20543
B_{c2}^{*+}	545

$c\overline{c}$ MESONS

$\eta_c(1S)$	441
$\chi_{c0}(1P)$	10441
$\eta_c(2S)$	100441
$J/\psi(1S)$	443
$h_c(1P)$	10443
$\chi_{c1}(1P)$	**20443***
$\psi(2S)$	**100443***
$\psi(3770)$	30443
$\psi(4040)$	**9000443***
$\psi(4160)$	**9010443***
$\psi(4415)$	**9020443***
$\chi_{c2}(1P)$	445

$b\overline{b}$ MESONS

$\eta_b(1S)$	551
$\chi_{b0}(1P)$	**10551***
$\eta_b(2S)$	100551
$\chi_{b0}(2P)$	**110551***
$\eta_b(3S)$	200551
$\chi_{b0}(3P)$	210551
$\Upsilon(1S)$	553
$h_b(1P)$	10553
$\chi_{b1}(1P)$	**20553***
$\Upsilon_1(1D)$	30553
$\Upsilon(2S)$	**100553***
$h_b(2P)$	110553
$\chi_{b1}(2P)$	**120553***
$\Upsilon_1(2D)$	130553
$\Upsilon(3S)$	**200553***
$h_b(3P)$	210553
$\chi_{b1}(3P)$	220553
$\Upsilon(4S)$	**300553***
$\Upsilon(10860)$	**9000553***
$\Upsilon(11020)$	**9010553***
$\chi_{b2}(1P)$	555
$\eta_{b2}(1D)$	10555
$\Upsilon_2(1D)$	20555
$\chi_{b2}(2P)$	**100555***
$\eta_{b2}(2D)$	110555
$\Upsilon_2(2D)$	120555
$\chi_{b2}(3P)$	200555
$\Upsilon_3(1D)$	557
$\Upsilon_3(2D)$	100557

LIGHT BARYONS

p	2212
n	2112
Δ^{++}	2224
Δ^+	2214
Δ^0	2114
Δ^-	1114

STRANGE BARYONS

Λ	3122
Σ^+	3222
Σ^0	3212
Σ^-	3112
Σ^{*+}	3224[d]
Σ^{*0}	3214[d]
Σ^{*-}	3114[d]
Ξ^0	3322
Ξ^-	3312
Ξ^{*0}	3324[d]
Ξ^{*-}	3314
Ω^-	3334[d]

CHARMED BARYONS

Λ_c^+	4122
Σ_c^{++}	4222
Σ_c^+	4212
Σ_c^0	4112
Σ_c^{*++}	4224
Σ_c^{*+}	4214
Σ_c^{*0}	4114
Ξ_c^+	**4232***
Ξ_c^0	**4132***
$\Xi_c'^+$	4322
$\Xi_c'^0$	4312
Ξ_c^{*+}	4324
Ξ_c^{*0}	4314
Ω_c^0	4332
Ω_c^{*0}	4334
Ξ_{cc}^+	4412
Ξ_{cc}^{++}	4422
Ξ_{cc}^{*+}	4414
Ξ_{cc}^{*++}	4424
Ω_{cc}^+	4432
Ω_{cc}^{*+}	4434
Ω_{ccc}^{++}	4444

BOTTOM BARYONS

Λ_b^0	5122
Σ_b^-	5112
Σ_b^0	5212
Σ_b^+	5222
Σ_b^{*-}	5114
Σ_b^{*0}	5214
Σ_b^{*+}	5224
Ξ_b^-	5132
Ξ_b^0	5232
$\Xi_b'^-$	5312
$\Xi_b'^0$	5322
Ξ_b^{*-}	5314
Ξ_b^{*0}	5324
Ω_b^-	5332
Ω_b^{*-}	5334
Ξ_{bc}^0	5142
Ξ_{bc}^+	5242
$\Xi_{bc}'^0$	5412
$\Xi_{bc}'^+$	5422
Ξ_{bc}^{*0}	5414
Ξ_{bc}^{*+}	5424
Ω_{bc}^0	5342
$\Omega_{bc}'^0$	5432
Ω_{bc}^{*0}	5434
Ω_{bcc}^+	5442
Ω_{bcc}^{*+}	5444
Ξ_{bb}^-	5512
Ξ_{bb}^0	5522
Ξ_{bb}^{*-}	5514
Ξ_{bb}^{*0}	5524
Ω_{bb}^-	5532
Ω_{bb}^{*-}	5534
Ω_{bbc}^0	5542
Ω_{bbc}^{*0}	5544
Ω_{bbb}^-	5554

Footnotes to the Tables:

$*$) Numbers which have changed since the 1996 *Review* [4] are in bold face.
Numbers which were not assigned in the 1996 *Review* [4] are in regular type.

a) Particulary in the third generation, the left and right sfermion states may mix, as shown. The lighter mixed state is given the smaller number.

b) The physical $\widetilde{\chi}$ states are admixtures of the pure $\widetilde{\gamma}$, $\widetilde{Z}^0$, $\widetilde{W}^+$, $\widetilde{H}_1^0$, $\widetilde{H}_2^0$, and $\widetilde{H}^+$ states.

c) In this draft we have only provided one generic leptoquark code. More general classifications according to spin, weak isospin and flavor content would lead to a host of states, that could be added as the need arises.

d) Σ^* and Ξ^* are alternate names for $\Sigma(1385)$ and $\Xi(1530)$.