Table of Contents

LoKi's Generator (HepMC) Particle Functions

G3Q, the instance of LoKi::GenParticles::ThreeCharge()

The simple function which returns the triple charge of the particle:

```
1000const HepMC::GenParticle* p = ... ;
1010const double q3 = G3Q ( p ) ;
```

GABSID, the instance of LoKi::GenParticles::AbsIdentifier()

The simple function which returns the absolute value of PDG-identifier of the particle:

```
1000const HepMC::GenParticle* p = ... ;
1010const double absid = GABSID( p ) ;
```

The special equality/non-equality operators against `std::string` and `LHCb::ParticleID` objects are defined:

```
1000GRange kaons = gselect ( "kaons" , "K+" == GABSID ) ;
1010GRange nonmuons = gselect ( "!mu" , LHCb::ParticleID( 13 ) != GABSID ) ;
```

See also GID.

GALL, the instance of LoKi::BooleanConstant<const HepMC::GenParticle*>(true)

The most trivial *"select-all"* predicate which always returns **true**

GANCESTOR, C++ type

`LoKi::GenParticles::IsAnAncestor`

The predicate which evaluates for `true` for all 'ancestors' of given particle:

```
1000const HepMC::GenParticle bquark = ... ;
1010
1020// get all particles from b-quark
1030typedef std::vector<const HepMC::GenParticle*> GPs ;
1040GPs gps ;
1050const LHCb::HepMCEvents* events = ... ;
1060LoKi::Extract::genParticles
1070                      (  events                       ,   // source
1080                         std::back_inserter ( gps ) ,   // target
1090                         GANCESTOR ( bquark )       ) ; // predicate
1100
```

GBAR, the instance of LoKi::GenParticles:BarCode()

Simple function which returns the *bar-code* of the particle:

```
1000GRange one = gselect ( "One" , 1 == BAR ) ;
```

GBARCODE

It is an alias for `GBAR`, see GBAR

GBARYON, the instance of LoKi::GenParticles::IsBaryon()

Simple predicate which returns `true` for baryons

```
1000// select all  beuaty baryons
1010GRange good = gselect ( "good" ,  GBARYON && GBEAUTY )  ;
```

GBEAUTY, the instance of LoKi::GenParticles::HasQuark(LHCb::ParticleID::bottom)

Simple predicate which checks the presence of beauty quark:

```
1000// select energetic beauty particles
1010GRange good = gselect ( "good" ,  GBEAUTY && GPT > 1 * GeV )  ;
```

GCHARGED, the instance of LoKi::GenParticles::IsCharged()

Simple predicate which checks the charge of the particle:

```
1000// select all charged msons
1010GRange good = gselect ( "mesons" ,  GMESON && GCHARGED )  ;
```

GCHARM, the instance of LoKi::GenParticles::HasQuark(LHCb::ParticleID::charm)

Simple predicate which checks the presence of charm quark:

```
1000// select energetic charm particles
1010GRange good = gselect ( "good" ,  GCHARM && GPT > 1 * GeV )  ;
```

GE, the instance of LoKi::GenParticles::Energy()

The function which returns the transverse momenum of the particle:

```
1000// select all fast particles only:
1010GRange fast  = gselect ( "fast" ,  GE > 50 * GeV  )  ;
```

GETA, the instance of LoKi::GenParticles::PseudoRapidity()

Simple function which evaluates pseudorapidity, η, of the particle:

```
1000GRange inAcc = gselect ( "inAcc" , 5 > GETA &&  GETA > 2  ) ;
```

GDELTAR2

It is an alias for `GDR2`, see GDR2

GDETA, C++ type LoKi::GenParticles::DeltaEta

Simple function which evaluates the difference in the pseudorapidity η with respect to some reference value η_0:

```
1000const HepMC::GenParticle* p = ... ;
1010const Gaudi::LorentzVector& v = ... ;
1020Fun deta = GDETA( v ) ;
1030const double result = deta ( p ) ;
```

GDPHI, C++ type LoKi::GenParticles::DeltaPhi

Simple function which evaluates the difference in the azimuthal angle ϕ with respect to some reference value ϕ_0:

```
1000const HepMC::GenParticle* p = ... ;
1010const Gaudi::LorentzVector& v = ... ;
1020Fun dphi = GDPHI( v ) ;
1030const double result = dphi ( p ) ;
```

GDR2, C++ type LoKi::GenParticles::DeltaR2

Simple function which evaluates $\sqrt{\Delta\phi^2 + \Delta\eta^2}$

```
1000const HepMC::GenParticle* p = ... ;
1010const Gaudi::LorentzVector& v = ... ;
1020Fun dr2 = GDR2( v.Phi() ) ;
1030const double result = dr2 ( p ) ;
```

GFALSE

It is an alias for GNONE, see GNONE

GFAPVX, C++ type C++ type LoKi::GenParticles::AdapterToProductionVertex

The helper adapter which delegates the evaluation of the *"vertex"* function to *production vertex* of the particle:

```
1000/// Extract all particles, which are produces at |z|<10
1010const LHCb::HepMCEvents* events
1020  get<LHCb::HepMCEvents>( LHCb::HepMCEventLocation::Default ) ;
1030
1040typedef std::vector<HepMC::GenParticle*> PARTICLES ;
1050PARTICLES parts ;
1060
1070// create the predicate:
1080GCut cut = GFAPVX( abs( GVZ  ) , 0 )  < 10 ;
1090LoKi::Extract::genParticles ( events , std::back_inserter ( parts )  , cut ) ;
```

GFAEVX, C++ type C++ type LoKi::GenParticles::AdapterToEndVertex

The helper adapter which delegates the evaluation of the *"vertex"* function to *end-vertex* of the particle:

```
1000/// Extract all particles, which are decayed after  |z|>1*meter
1010const LHCb::HepMCEvents* events
1020  get<LHCb::HepMCEvents>( LHCb::HepMCEventLocation::Default ) ;
1030
1040typedef std::vector<HepMC::GenParticle*> PARTICLES ;
```

```
1050PARTICLES parts ;
1060
1070// create the predicate:
1080GCut cut = GFAEVX( abs( GVZ  ) , 0 )  >  1 * meter ;
1090LoKi::Extract::genParticles ( events , std::back_inserter ( parts )  , cut ) ;
```

GFROMTREE, C++ type LoKi::GenParticles::FromHepMCTree

The predicate which evaluates for `true` for all particles which belongs to "descendants" of the given particle/vertex

```
1000const HepMC::GenParticle bquark = ... ;
1010
1020// get all particles from b-quark
1030typedef std::vector<const HepMC::GenParticle*> GPs ;
1040GPs gps ;
1050const LHCb::HepMCEvents* events = ... ;
1060LoKi::Extract::genParticles
1070                      (  events                                  ,   // source
1080                         std::back_inserter ( gps )    ,   // target
1090                         GFROMTREE ( bquark )       ) ; // predicate
1100
```

GHADRON, the instance of LoKi::GenParticles::IsHadron()

Simple predicate which returns `true` for hdrons

```
1000// select all charged hadrons
1010GRange good = gselect ( "good" ,  GHADRON && GCHARGED )  ;
```

GID, the instance of LoKi::GenParticles::Identifier()

The simple function which returns the value of PDG-identifier of the particle:

```
1000const HepMC::GenParticle* p = ... ;
1010const double absid = GID( p ) ;
```

The special equality/non-equality operators against `std::string` and `LHCb::ParticleID` objects are defined:

```
1000GRange kaons = gselect ( "kaons" , "K+" == GID ) ;
1010GRange nonmuons = gselect ( "!mu" , LHCb::ParticleID( 13 ) != GID ) ;
```

See also GABSID.

GINTREE, C++ type LoKi::GenParticles::InTree()

Simple predicate which checks the presence of the particle, which satisfies the criteris in the decay tree

```
1000/// select semileptonic decays of beauty baryons:
1010GRange good = gselect ( "good" , GBEAUTY && GBARYON && GINTREE( GLEPTON ) ) ;
```

GLEPTON, the instance of LoKi::GenParticles::IsLepton()

Simple predicate which returns `true` for leptons

```
1000// select all neutral leptons
1010GRange neutrinos = gselect ( "neutrinos" ,  GLEPTON && GNEUTRAL )  ;
```

GM, the instance of LoKi::GenParticles::Mass()

The function which returns the mass of the particle:

```
1000const HepMC::GenParticle* p = ... ;
1010const double mass = GM (  p ) ;
```

GMAX, C++ type LoKi::Max<const HepMC::GenParticle*>

The function which evaluated the maximum from the several other functions:

```
1000GRange inCaloAcc = gselect ( "inCaloAcc" , GMAX( abs(PX/PZ) , abs(PY,PZ) ) < 300 * mrad ) ;
1010/// the same:
1020GRange inCaloAcc = gselect ( "inCaloAcc" , max( abs(PX/PZ) , abs(PY,PZ) ) < 300 * mrad ) ;
```

GMESON, the instance of LoKi::GenParticles::IsMeson()

Simple predicate which returns `true` for mesons

```
1000// select all charmed mesons:
1010GRange good = gselect ( "good" ,  GMESON && GCHARM )  ;
```

GMIN, C++ type LoKi::Min<const HepMC::GenParticle*>

The function which evaluated the minimum from the several other functions:

```
1000GRange inCaloAcc = gselect ( "inCaloAcc" , GMIN( abs(PX/PZ) , abs(PY,PZ) ) > 30 * mrad ) ;
1010/// the same:
1020GRange inCaloAcc = gselect ( "inCaloAcc" , min( abs(PX/PZ) , abs(PY,PZ) ) > 30 * mrad ) ;
```

GMOMDIST, C++ type LoKi::GenParticles::MomentumDistance

The function which evaluates the euclidian distance between particle's 4-momentum and some reference 4-momentum. It is very useful for kinematical matching.

GNEUTRAL, the instance of LoKi::GenParticles::IsNeutral()

Simple predicate which checks the charge of the particle:

```
1000// select all neutral leptons
1010GRange neutrinos = gselect ( "neutrinos" ,  GLEPTON && GNEUTRAL )  ;
```

GNINTREE, C++ type LoKi::GenParticles::NInTree

The function which counts number of particles in th decay tree of the particle, which satisfy certain criteria:

```
1000// get Generator information
1010const LHCb::HepMCEvents* events = get<LHCb::HepMCEvents>( LHCb::HepMCEventLocation::Default
1020typedef std::vector<const HepMC::GenParticle*> GenParticles ;
1030/// select b(and antib) quarks from decay of higgs,
1040/// == "count of Higgs occurances within parents "
1050GenParticles bquarks ;
1060LoKi::Extract::genParticles
1070   ( events                          ,
1080     std::back_inserter( bquarks )  ,
1090     ( "b" == GABSID ) &&
1100        1 == GNINTREE( "H_10" == GABSID , HepMC::parents ) ) ;
```

GNLT, the instance of LoKi::GenParticles::NominalLifeTime()

The function which return the nominal life time of the particle,

```
1000// select long-lived particles
1010GRange inAcc = gselect ( "inAcc" ,  GNLT > 1 * mm )  ;
```

GNONE, the instance of LoKi::BooleanConstant<const HepMC::GenParticle*>(false)

The most trivial *"select-nothing"* predicate which always returns **false**

GNUCLEUS, the instance of LoKi::GenParticles::IsNucleus()

Simple predicate which returns `true` for nuclea

```
1000// select all nuclea
1010GRange nuclea = gselect ( "nuclea" ,  GNUCLEUS )  ;
```

GONE, the instance of LoKi::Constant<const HepMC::GenParticle*>(1)

The most trivial *"select-nothing"* predicate which always returns **1**

GQUARK, C++ type LoKi::GenParticles::HasQuark

Simple predicate which checks the quark content of the particles:

```
1000// select all particle which simultaneously contain charm and strange quark:
1010GRange ds = gselect ( "ds" , GQUARK(LHCb::ParticleID::charm) && GQUARK(LHCb::ParticleID::str
```

GP, the instance of LoKi::GenParticles::Momentum()

The function which return the momenum of the particle:

```
1000// select all fast particles only:
1010GRange fast  = gselect ( "fast" ,  GP > 10 * GeV  )  ;
```

GPHI, the instance of LoKi::GenParticles::Phi()

Simple function which evaluates the azimuthal angle, ϕ, of the particle.

GPT, the instance of LoKi::GenParticles::TransverseMomentum()

The function which returns the transverse momenum of the particle:

```
1000// select all fast particles only:
1010GRange fast  = gselect ( "fast" ,  GPT> 1 * GeV  )  ;
```

GPTDIR, C++ type LoKi::GenParticles::TransverseMomentumRel

The function which evaluates the transverse momentum with respect some direction, e.g. the direction of the jet:

```
1000const Gaudi::LorentzVector& jet = ... ;
1010// select only leptons which large pt with respect to the jet:
1020GRange withLargePt = gselect ( "LargePT" , LEPTON && GPTDIR ( jet )  > 1 * GeV ) ;
```

GPX, the instance of LoKi::GenParticles::MomentumX()

The function which return x-component of particle's momentum:

```
1000// select all fast particles only:
1010GRange fast  = gselect ( "fast" ,  sqrt(GPX*GPX+GPY*GPY) >  2 * GeV  )  ;
```

GPY, the instance of LoKi::GenParticles::MomentumY()

The function which return y-component of particle's momentum:

```
1000// select all fast particles only:
1010GRange fast  = gselect ( "fast" ,  sqrt(GPX*GPX+GPY*GPY) >  2 * GeV  )  ;
```

GPZ, the instance of LoKi::GenParticles::MomentumZ()

The function which return z-component of particle's momentum:

```
1000// select all fast particles only:
1010GRange inAcc = gselect ( "inAcc" ,  sqrt(GPX*GPX+GPY*GPY)/GPZ <  300 * mrad  )  ;
```

GSSWITCH, C++ type LoKi::SimpleSwitch<const HepMC::GenParticle*>

The function which acts according to the rule: _ *result* = *condition ? constant1 : constant2*_. Essentially it could be considered as some kind of *converter* from *predicate* to *function*

GSTATUS, the instance of LoKi::GenParticles::Status()

The function which returns *status* of the particle

GSTRANGE, the instance of LoKi::GenParticles::HasQuark(LHCb::ParticleID::strange)

Simple predicate which checks the presence of strange quark:

```
1000// select beauty and strange baryons
1010GRange bs = gselect ( "bs" ,  GBARYON && GBEAUTY && GSTRANGE )  ;
```

GSWITCH, C++ type LoKi::Switch<const HepMC::GenParticle*>

The function which acts according to the rule: _ *result* = *condition ? result1 : result2*_:

```
1000GFun fun = GSWITCH ( 0 < G3Q , 1/GP , GPT ) ;
```

GTHETA, the instance of LoKi::GenParticles::Theta()

Simple function which evaluates the polar angle, θ, of the particle.

GTIME, the instance of LoKi::GenParticles::ProperLifeTime()

The function which return the proper life time of the particle, $c * \cdot \tau$:

```
1000// select long-lived particles
1010GRange inAcc = gselect ( "inAcc" ,  CTIME > 1 * mm )  ;
```

GTOP, the instance of LoKi::GenParticles::HasQuark(LHCb::ParticleID::top)

Simple predicate which checks the presence of top quark:

```
1000// select top particles (top-quark only?)
1010GRange good = gselect ( "good" ,  GTOP )  ;
```

GTRUE

It is an alias for GALL, see GALL

GVEV, the instance of LoKi::GenParticles::ValidEndVertex()

Simple predicate which evaluates the validity of the *end-vertex* of the particle:

```
1000GRange decayedB = gselect ( "Bdec" , "B0" == GID && GVEV ) ;
```

GZERO, the instance of LoKi::Constant<const HepMC::GenParticle*>(0)

The most trivial *"select-nothing"* predicate which always returns 0

-- Vanya BELYAEV - 20 Jul 2007

This topic: LHCb > LoKiGenParticleFunctions
Topic revision: r5 - 2013-12-15 - AlexN