Primary particle

Geant4 tutorial
Paris, 4-8 June 2007

Giovanni Santin
ESA / ESTEC

Rhea System SA

With material from previous tutorials by Makoto Asai

Contents

- G4VUserPrimaryGeneratorAction
- Primary vertex and primary particle
- Built-in primary particle generators
  - Particle gun
  - Interfaces to HEPEVT and HEPMC
  - General particle source
- (Exotic primary particle, pre-assigned decay)

Primary particle generation

User classes

- Initialization classes
  - Use G4RunManager::SetUserInitialization() to define.
  - Invoked at the initialization
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
- Action classes
  - Use G4RunManager::SetUserAction() to define.
  - Invoked during an event loop
    - G4VUserPrimaryGeneratorAction 
    - G4UserRunAction
    - G4UserEventAction
    - G4UserStackingAction
    - G4UserTrackingAction
    - G4UserSteppingAction
- main()
  - Geant4 does not provide *main().*

Note : classes written in orange are mandatory.

G4VUserPrimaryGeneratorAction

- This class is one of the mandatory user classes and

  controls the generation of primaries
  - What kind of particle (how many) what energy, position, direction, polarisation, etc

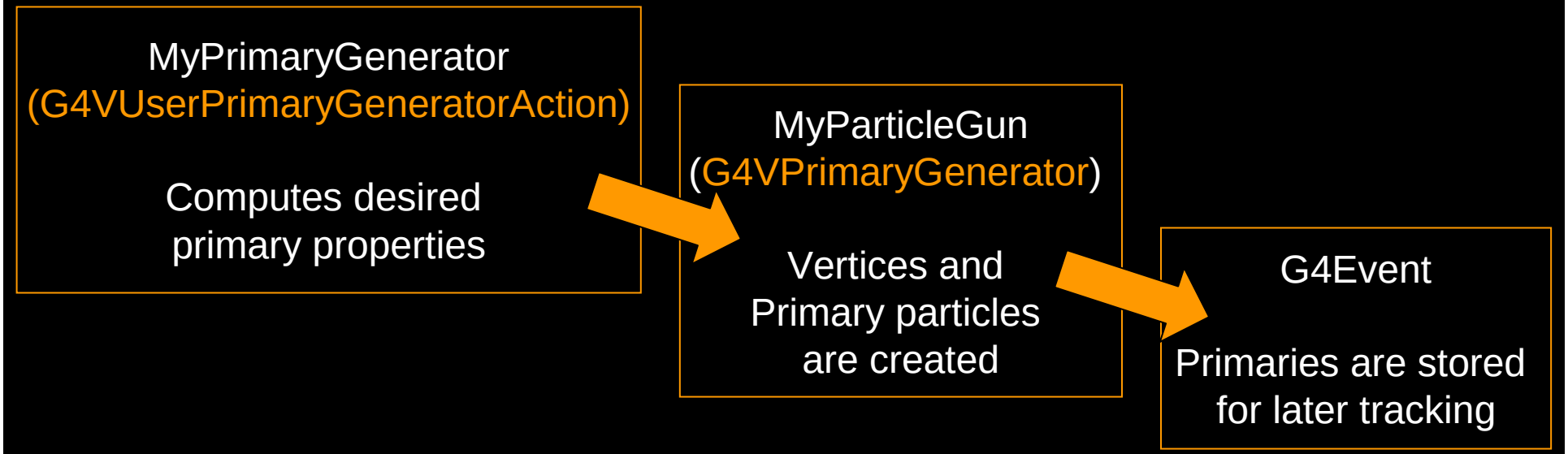
- This class itself should NOT generate primaries but invoke **`GeneratePrimaryVertex()`** method of primary generator(s) to make primaries (G4VPrimaryGenerator)

- Constructor
  - Instantiate primary generator(s)
  - Set default values to it (them)

- **`GeneratePrimaries()`** method
  - Randomize particle-by-particle value(s)
  - Set these values to primary generator(s)
    - Never use hard-coded UI commands
  - Invoke **`GeneratePrimaryVertex()`** method of primary generator(s)

Primary vertices and primary particles

- Primary vertices and primary particles are stored in G4Event in advance to processing an event.
  - G4PrimaryVertex and G4PrimaryParticle classes
  - These classes don't have any dependency to G4ParticleDefinition nor G4Track.
  - They will become "primary tracks" only at Begin of Event phase and put into a "stack"



Pre-assigned decay for primaries

- Capability of bookkeeping decay chains
- Primary particles may not necessarily be particles which can be tracked by Geant4
  - Pre-assigned decay channels attached to particles
  - Also, “exotic” particles can be imported from Particle Generators, followed by either decay or user defined physics processes

    (e.g. Higgs, W/Z boson, SUSY particle, …)
- See talk by Makoto Asai

Built-in primary particle generators

- Geant4 provides some concrete implementations of G4VPrimaryGenerator.
  - G4ParticleGun
  - G4HEPEvtInterface, G4HEPMCInterface
  - G4GeneralParticleSource

G4ParticleGun

- Concrete implementations of G4VPrimaryGenerator
  - A good example for experiment-specific primary generator implementation
- It shoots one primary particle of a certain energy from a certain point at a certain time to a certain direction.
  - Various set methods are available
  - Intercoms commands are also available for setting initial values
- G4ParticleGun is basic, but it can be used from inside UserPrimaryGeneratorAction to model complex source types / distributions:
  - Generate the desired distributions (by shooting random numbers)
  - Use set methods of G4ParticleGun
  - Use G4ParticleGun as many times as you want
  - Use any other primary generators as many times as you want to make overlapping events

G4VUserPrimaryGeneratorAction

- Example of usage of G4ParticleGun

```
void T01PrimaryGeneratorAction::
         GeneratePrimaries(G4Event* anEvent)
{ G4ParticleDefinition* particle;
  G4int i = (int)(5.*G4UniformRand());
  switch(i)
  { case 0: particle = positron; break; ... }
  particleGun->SetParticleDefinition(particle);
  G4double pp =
    momentum+(G4UniformRand()-0.5)*sigmaMomentum;
  G4double mass = particle->GetPDGMass();
  G4double Ekin = sqrt(pp*pp+mass*mass)-mass;
  particleGun->SetParticleEnergy(Ekin);
  G4double angle = (G4UniformRand()-0.5)*sigmaAngle;
  particleGun->SetParticleMomentumDirection
           (G4ThreeVector(sin(angle),0.,cos(angle)));
  particleGun->GeneratePrimaryVertex(anEvent);
}
```

- You can repeat this for generating more than one primary particles.

Interfaces to HEPEvt and HepMC

- Concrete implementations of G4VPrimaryGenerator
  - Good examples for experiment-specific primary generator implementation
  - Interface to external physics generators

- G4HEPEvtInterface
  - Event record structure based on /HEPEVT/ common block
  - Used by (FORTRAN) HEP physics generators
  - Developed and agreed on within the framework of the 1989 LEP physics study
  - ASCII file input

- G4HepMCInterface
  - HepMC Event record for MC generators. Object Oriented, C++
  - Used by new (C++) HEP physics generators.
  - ASCII file input or direct linking to a generator through HepMC.

G4GeneralParticleSource (GPS)

An advanced concrete implementation of G4VPrimaryGenerator

Offers as pre-defined many common (and not so common) options for particle generation:

- Primary vertex can be randomly positioned with options
  - *Point*, *Beam, Plane (Circle, Annulus, Ellipsoid, Square or Rectangle)*, *Surface* or *Volume (Sphere, Ellipsoid, Cylinder or Para)*
- Angular emission can be
  - isotropic (iso), cosine-law (cos), planar wave (planar), 1-d accelerator beam (beam1d), 2-d accelerator beam (beam2d), focusing to a point (focused) or user-defined (user)
- Kinetic energy of the primary particle can also be randomized.
  - mono-energetic (Mono), linear (Lin), power-law (Pow), exponential (Exp), Gaussian (Gauss), bremsstrahlung (Brem), black-body (Bbody), cosmic diffuse gamma ray (Cdg), user-defined histogram (User), arbitrary point-wise spectrum (Arb) and user-defined energy per nucleon histogram (Epn)
- Multiple sources
  - With user defined relative intensity
- Capability of event biasing (variance reduction).
  - By enhancing particle type, distribution of vertex point, energy and/or direction
- All features can be used via C++ or command line (or macro) UI

GPS
Typical UserPrimaryGeneratorAction class

- Can be extremely simple:

```
GRASPrimaryGeneratorAction::GRASPrimaryGeneratorAction() {
     m_particleGun = new G4GeneralParticleSource();
}

GRASPrimaryGeneratorAction::~GRASPrimaryGeneratorAction() {
     delete m_particleGun;
}

void GRASPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent) {
     m_particleGun->GeneratePrimaryVertex(anEvent);
}
```

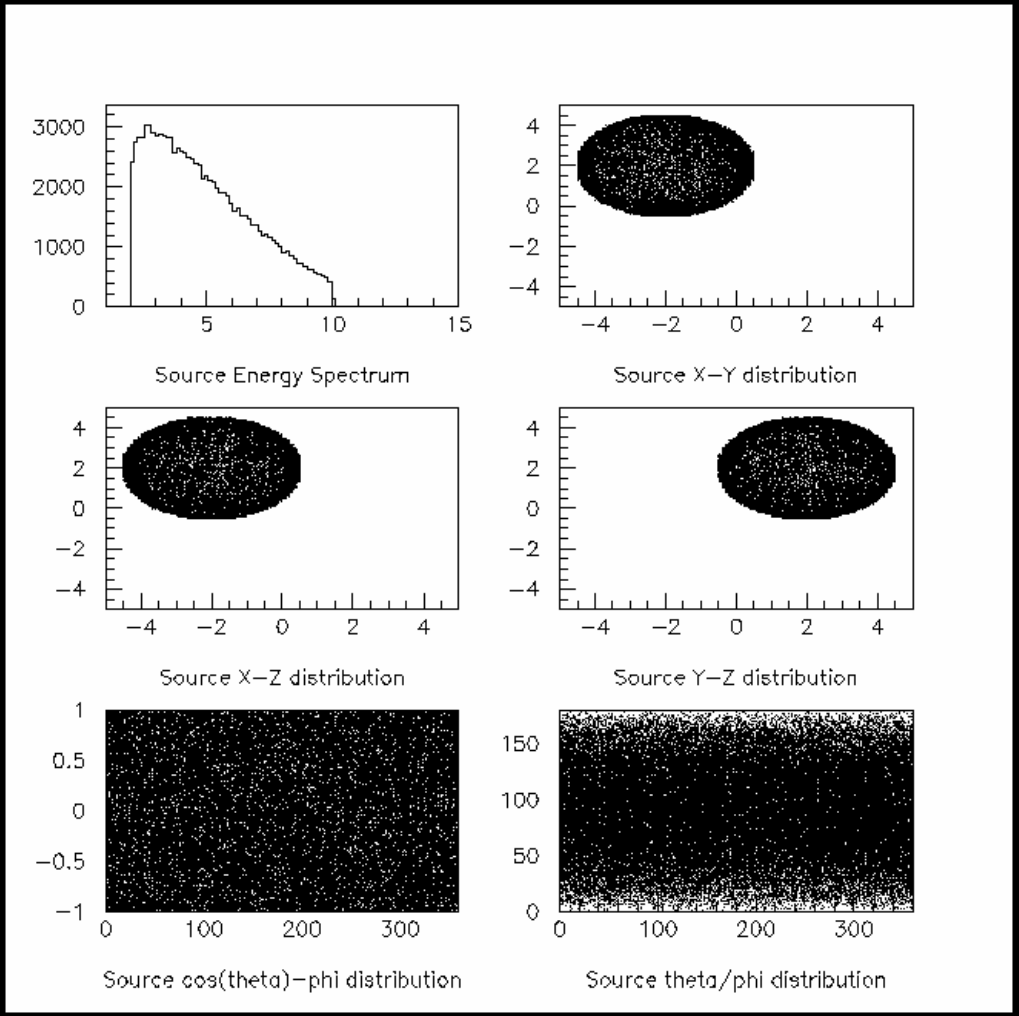
- All user instructions given via macro UI commands
- Extensive documentation at http://reat.space.qinetiq.com/gps

GPS
Example 6

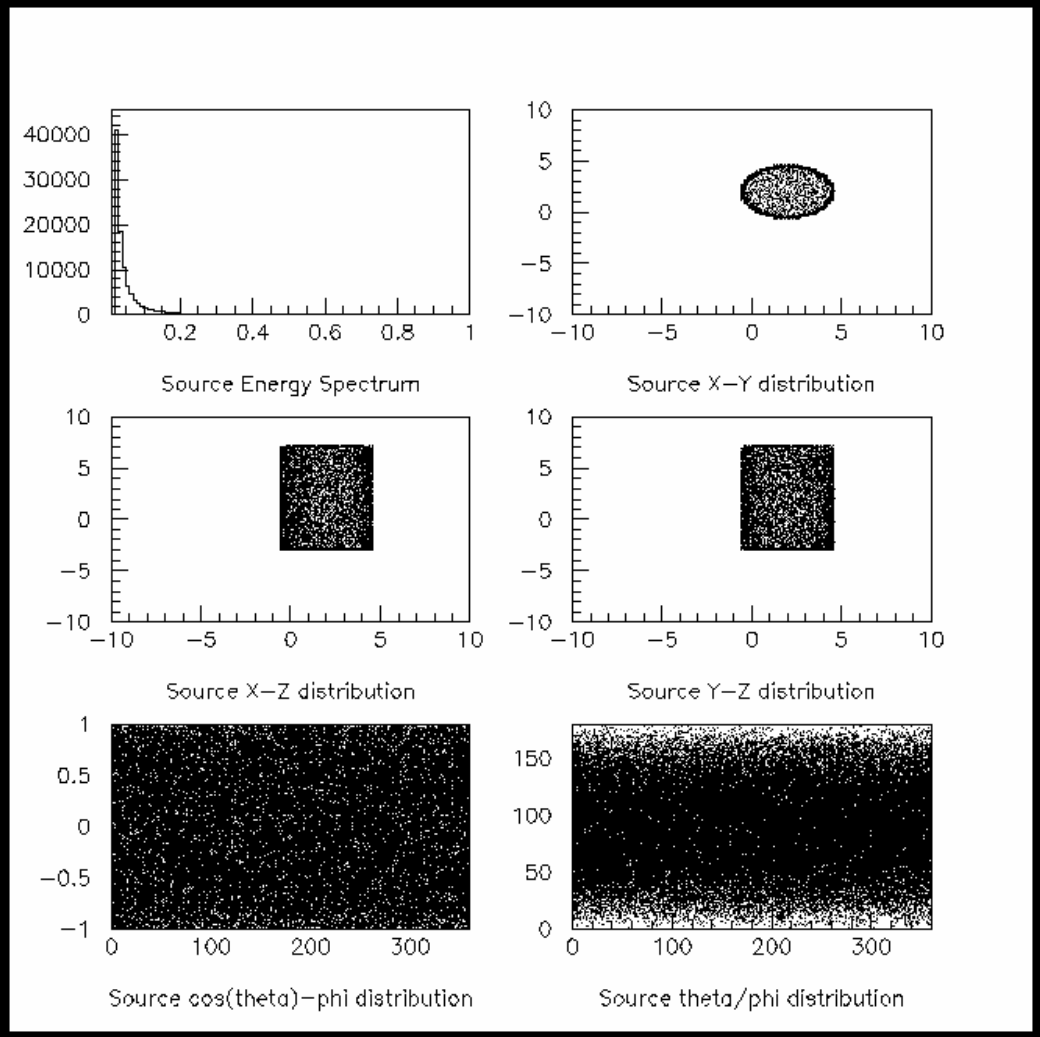
- Vertex on sphere surface
- Isotropic emission
- Pre-defined spectrum (black-body)

- Macro

```
/gps/particle geantino
/gps/pos/type Surface
/gps/pos/shape Sphere
/gps/pos/centre -2. 2. 2. cm
/gps/pos/radius 2.5 cm
/gps/ang/type iso
/gps/ene/type Bbody
/gps/ene/min 2. MeV
/gps/ene/max 10. MeV
/gps/ene/temp 2e10
/gps/ene/calculate
```



GPS
Example 7



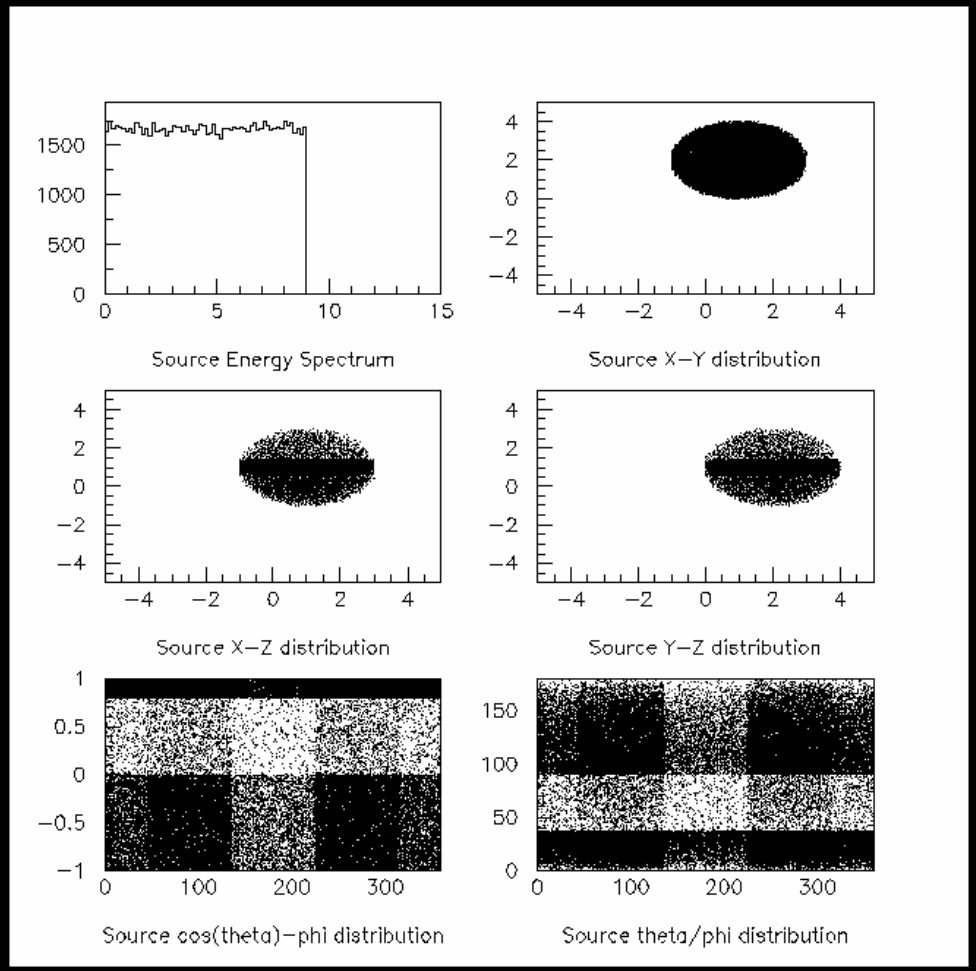
- Vertex on cylinder surface
- Cosine-law emission
  (to mimic isotropic source in space)
- Pre-defined spectrum
  (Cosmic Diffuse Gamma)

- Macro

```
/gps/particle gamma
/gps/pos/type Surface
/gps/pos/shape Cylinder
/gps/pos/centre 2. 2. 2. cm
/gps/pos/radius 2.5 cm
/gps/pos/halfz 5. cm
/gps/ang/type cos
/gps/ene/type Cdg
/gps/ene/min 20. keV
/gps/ene/max 1. MeV
/gps/ene/calculate
```

GPS
Example 24

- Vertex in sphere volume with z biasing
- Isotropic radiation with theta and phi biasing
- Integral arbitrary point-wise energy distribution with linear interpolation.

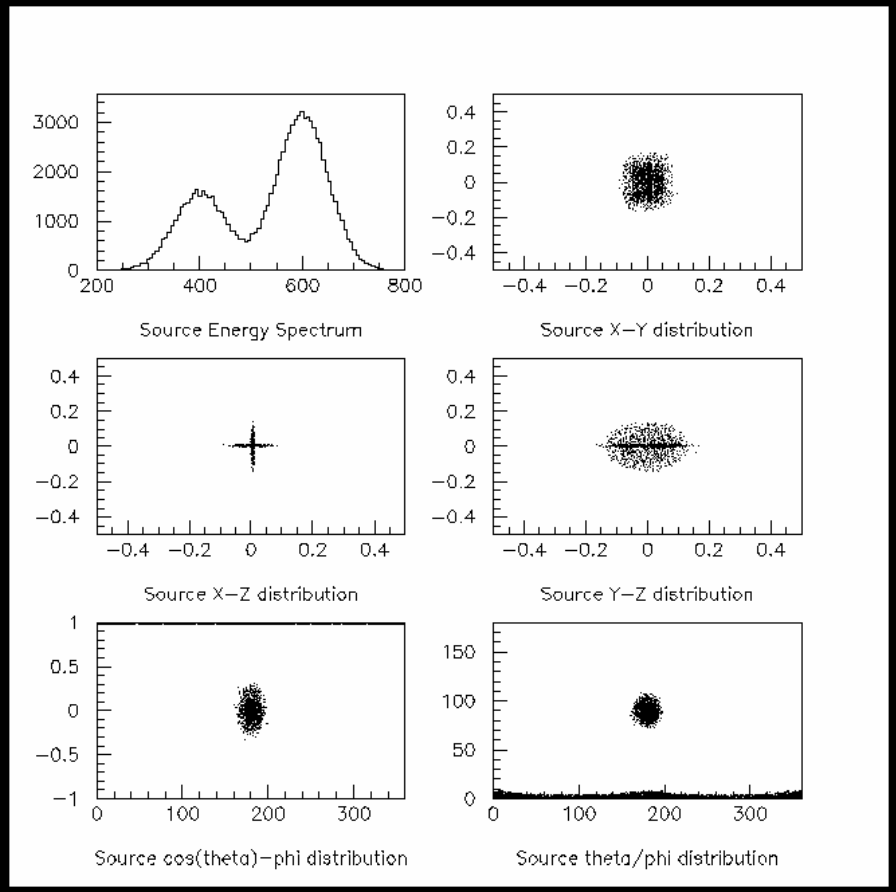


- Macro

```
/gps/particle geantino
/gps/pos/type Volume
/gps/pos/shape Sphere
/gps/pos/centre 1. 2. 1. cm
/gps/pos/radius 2. Cm

/gps/ang/type iso

/gps/ene/type Arb
/gps/ene/diffspec 0
/gps/hist/type arb
/gps/hist/point 0.0 11.
/gps/hist/point 1.0 10.
/gps/hist/point 2.0 9.
/gps/hist/point 3.0 8.
/gps/hist/point 4.0 7.
/gps/hist/point 7.0 4.
/gps/hist/point 8.0 3.
/gps/hist/point 9.0 2.
/gps/hist/point 10.0 1.
/gps/hist/point 11.0 0.
/gps/hist/inter Lin

/gps/hist/type biasz
/gps/hist/point 0. 0.
/gps/hist/point 0.4 0.5
/gps/hist/point 0.6 1.
/gps/hist/point 1. 0.2

/gps/hist/type biast
/gps/hist/point 0. 0.
/gps/hist/point 0.1 1.
/gps/hist/point 0.5 0.1
/gps/hist/point 1. 1.

/gps/hist/type biasp
/gps/hist/point 0. 0.
/gps/hist/point 0.125 1.
/gps/hist/point 0.375 4.
/gps/hist/point 0.625 1.
/gps/hist/point 0.875 4.
/gps/hist/point 1. 1.
```

GPS
Example 31

- Two-beam source definition (multiple sources)
- Gaussian profile
- Can be focused / defocused



- Macro

```
# beam #1
    # default intensity is 1 now change to 5.
    /gps/source/intensity 5.

    /gps/particle proton
    /gps/pos/type Beam

    # the incident surface is in the y-z plane
    /gps/pos/rot1 0 1 0
    /gps/pos/rot2 0 0 1

    # the beam spot is centered at the origin and is
    # of 1d gaussian shape with a 1 mm central plateau
    /gps/pos/shape Circle
    /gps/pos/centre 0. 0. 0. mm
    /gps/pos/radius 1. mm
    /gps/pos/sigma_r .2 mm
    #
    # the beam is travelling along the X_axis with 5
    degrees dispersion
    /gps/ang/rot1 0 0 1
    /gps/ang/rot2 0 1 0
    /gps/ang/type beam1d
    /gps/ang/sigma_r 5. deg
    #
    # the beam energy is in gaussian profile centered
    at 400 MeV
    /gps/ene/type Gauss
    /gps/ene/mono 400 MeV
    /gps/ene/sigma 50. MeV


# beam #2
    # 2x the instensity of beam #1
    /gps/source/add 10.
    #
    # this is a electron beam
    ...
```