Dream machine & Geant4 Instruction

Hanwook Bae
Kyungpook National University

SPDAK 2019

Contents

DREAM cluster

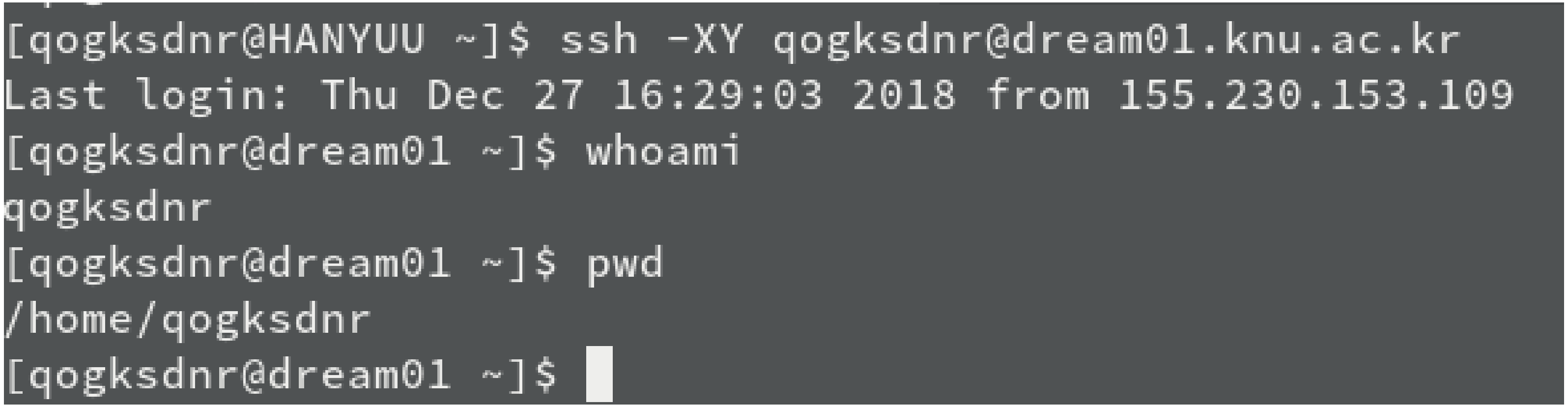
- There are two servers of DREAM cluster
  dream01.knu.ac.kr / dream02.knu.ac.kr
- Each server has 40 cores of CPU
- You should connect to one of the servers using SSH client and Xming (for Windows users) or just SSH client (for Linux users)

How to connect to DREAM cluster (for Linux users)

- command:

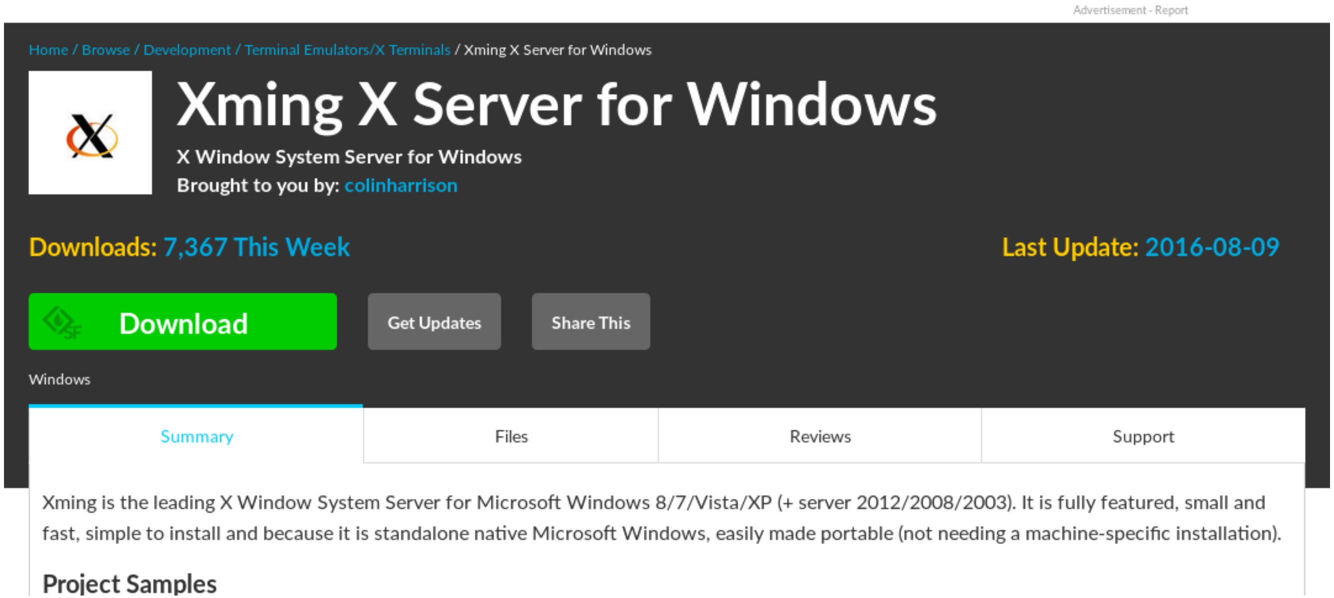
  ssh -XY [userID]@dream01.knu.ac.kr

  OR ssh -XY [userID]@dream02.knu.ac.kr



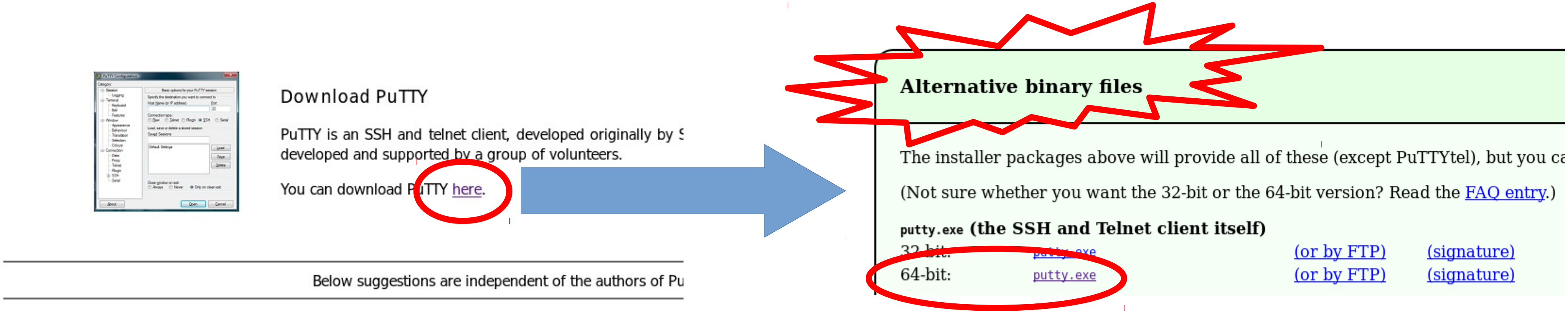
How to connect to DREAM cluster (for Windows users)

- Download Xming server for Windows (from: https://sourceforge.net/projects/xming/) or just google a keyword of “xming”



How to connect to DREAM cluster (for Windows users)

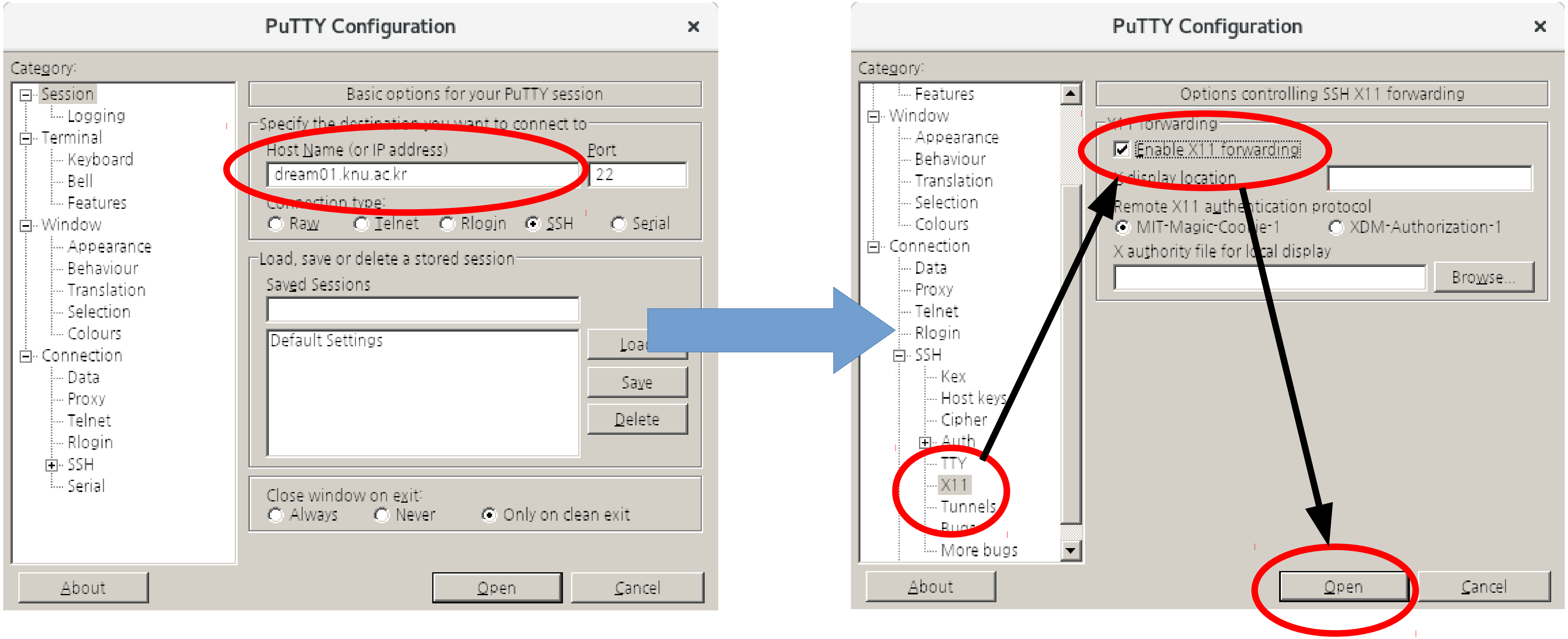
- Download Putty:
  (from: https://www.putty.org/)
  of just google “putty”
- In the above link...



How to connect to DREAM cluster (for Windows users)

- Install Xming → Execute the Xming server
- Set X11 forwarding setting for putty
- Connect to the DREAM cluster

Example for connection with putty



Configuring to use Geant4 & ROOT

- The path of common directory for this school: /gatbawi/palgongsan/spdak2019
- Copy the bash profile to your home directory
  : cp /gatbawi/palgongsan/spdak2019/bash_profile_2019 ~/.bash_profile
- Logout and re-login to DREAM cluster

Configuring to use Geant4 & ROOT

- Example for configuring:
  Entering the common directory and copy the bash profile to the home directory.

```
[qogksdnr@dream01 ~]$ pwd
/home/qogksdnr
[qogksdnr@dream01 ~]$ cd /gatbawi/palgongsan/spdak2019/
[qogksdnr@dream01 spdak2019]$ ls
BGOS  bash_profile_2019
[qogksdnr@dream01 spdak2019]$ cp bash_profile_2019 ~/.bash_profile
[qogksdnr@dream01 spdak2019]$
```

Configuring to use Geant4 & ROOT

- Logout and re-login to the cluster then check settings for Geant4 & ROOT

```
[qogksdnr@dream01 ~]$ exit
logout
Connection to dream01.knu.ac.kr closed.
[qogksdnr@HANYUU ~]$ ssh -XY qogksdnr@dream01.knu.ac.kr
Last login: Thu Dec 27 16:39:51 2018 from 155.230.153.109
[qogksdnr@dream01 ~]$ echo $ROOTSYS
/gatbawi/palgongsan/sw/root/6.06.02/x86_64/sl6-gcc4.9.3
[qogksdnr@dream01 ~]$ echo $G4LEDATA
/gatbawi/palgongsan/sw/Geant4/10.04.p02/x86_64/sl6-gcc4.9.3/share/Geant4-10.4.2/data/G4EMLOW7.3
```

BGOS Simulation Framework

- BGOS: BGO Simulation
- A simple Geant4 simulation source code with BGO scintillation crystal
- Skeleton source code is located at /gatbawi/palgongsan/spdak2019/BGOS/source_XXXX on the cluster
- Copy the entire source code directory to some directory

```
cd /gatbawi/palgongsan/spdak2019/BGOS
cp -r source /to/some/directory
```

Example macro of BGOS code

- Content of run_Example.mac

Dimension of BGO crystyal (X, Y, Z) →

```
#Following commands MUST be placed before a command of "/run/initialize"
#Dimension commands
/BGOS/det/cellSizeX 1 cm # X dimension of single BGO+GLASS+PMT Cell
/BGOS/det/cellSizeY 1 cm # Y dimension of single BGO+GLASS+PMT Cell
/BGOS/det/cellSizeZ 1 cm # Z dimension of single BGO+GLASS+PMT Cell
/BGOS/det/cellArrayNum 1 # Number of cell in single row of cell array.
```

Number of threads →

```
#Thread number commands
/run/numberOfThreads 4 # Number of threads
```

```
#Initialization of the Geant4 simulation engine
/run/initialize
```

Partigle gun commands →

```
#Particle gun commands
/gun/particle e- #Particle name
/gun/position 0 0 -2 cm #Starting position of particle
/gun/direction 0 0 1 #Momentum direction of particle
/gun/energy 500 keV #Kinetic energy of particle
```

Simulate 1000 events →

```
#Run the simulation
/run/beamOn 1000
```

Structure of BGOS code

Main source code (Entry point of simulation program): It defines physics list and initialize the simulation engine

Inner source codes

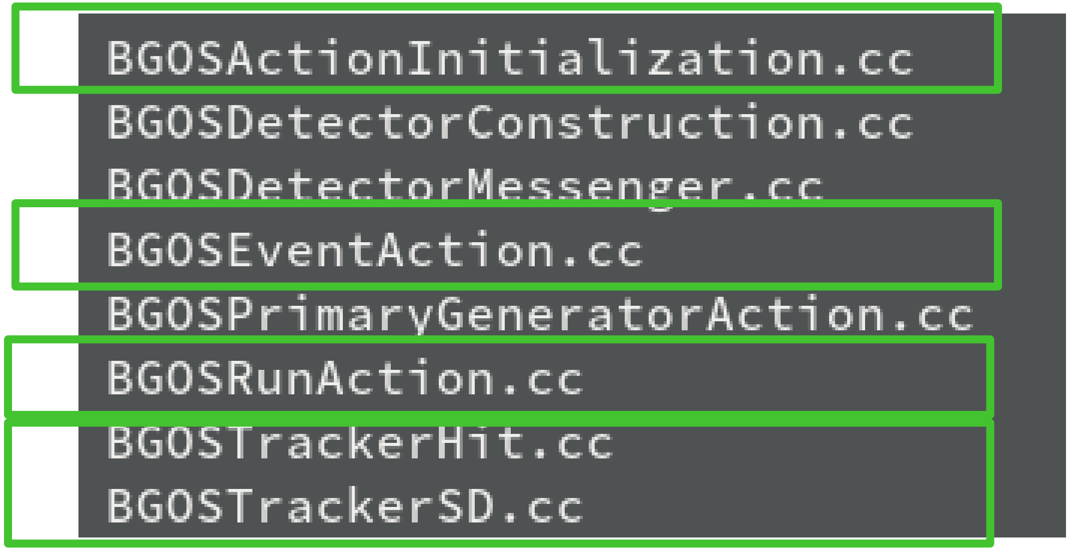
Example macro file

```
15:06 BGOScint.cc
15:06 BGOScint.in
15:06 BGOScint.out
15:06 CMakeLists.txt
15:06 gui.mac
15:06 include
15:06 init_vis.mac
15:06 run1.mac
15:06 run2.mac
16:20 run_Example.mac
16:28 src
15:06 startup.mac
15:06 vis.mac
```

Inner structure of BGOS code

Action Initialization: It contains initialization statements for various user action classes(EventAction, RunAction and so on...)

Event/Run Action: It defines an action that user wants to perform at the begin (or end) of an event or a run



Sensitive Detector and Hit: It manages the behaviour of our PMT

Inner structure of BGOS code

DetectorConstruction: It contains description of geometries for the simulation →

PrimaryGeneratorAction: It describes an event generator of the simulation →

```
BGOSActionInitialization.cc
BGOSDetectorConstruction.cc
BGOSDetectorMessenger.cc
BGOSEventAction.cc
BGOSPrimaryGeneratorAction.cc
BGOSRunAction.cc
BGOSTrackerHit.cc
BGOSTrackerSD.cc
```

The tuple

- Tuple is a data structure for storing finite sequence of an element

  Example: (1, 2, 5, 7, 3, 2, 1) OR
  (Z, A, B, E, A, C, A, B) …

- Tuple can have multiple elements and if the tuple has set of N elements, the tuple is called "N-tuple"
- Example of N-tuple: Personal information book

  Elements of i-th entry: Name, Sex, Age, City…
  (["Momo", "Female", 24, "Seoul"], ["Nayeon", "Female", 25, "Seoul"], ["Hanwook", "Male", 28, "Daegu"])

Output file of BGOS

- BGOS code creates a .ROOT file as its output
- The file contains an N-tuple named "NT"
- NT tuple consists of two elements: PECount, BGOEdep
- If you run the simulation in multithreading mode, the output file will be written separaterly by each thread

Description of each element

- PECount: Number of counts of photoelectrons emitted from BGO crystal in a single event
- BGOEdep: Total amount of energy deposit on BGO crystal in a single event

How to define a thing in Geant4

- In the real world, various properties needs to tell about an object: shape, color, conductivity, transparency, and so on…
- In Geant4, only two minimal properties are required to perform the detector simulation: Shape and Material
- This information is managed by the G4LogicalVolume class.
- We can add a thing in Geant4 world by creating a logical volume with proper parameters: Shape and Material

How to describe the shape

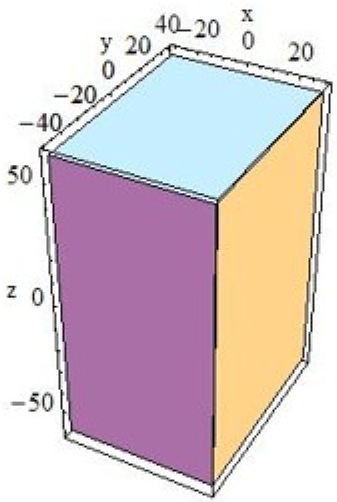
- First of all, Geant4 requires the shape of an object for the simulation (this is called Solid)
- This information can be dealt with Geometry classes: G4Tubs, G4Box, G4Sphere etc…
- Every geometry classes requires the name of solid
- Further parameter will be needed to set the dimension of a solid and this is different

Example of solids - Box

Box:

To create a **box** one can use the constructor:

```
G4Box(const G4String& pName,
            G4double  pX,
            G4double  pY,
            G4double  pZ)
```



In the picture:

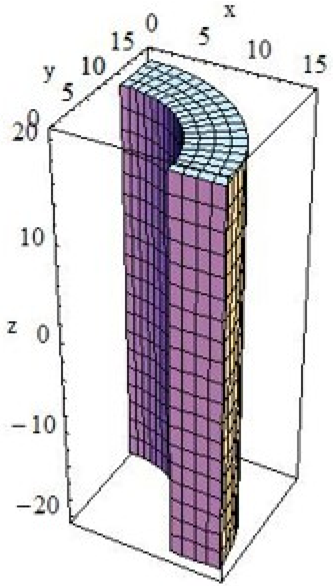
`pX = 30, pY = 40, pZ = 60`

Example of solids - Tube

Cylindrical Section or Tube:

Similarly to create a **cylindrical section** or **tube**, one would use the constructor:

```
G4Tubs(const G4String& pName,
             G4double  pRMin,
             G4double  pRMax,
             G4double  pDz,
             G4double  pSPhi,
             G4double  pDPhi)
```



In the picture:

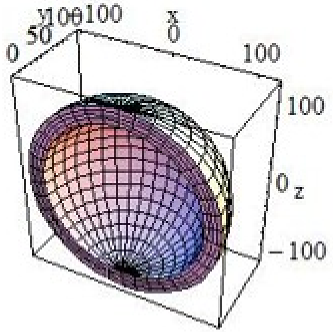
pRMin = 10, pRMax = 15, pDz = 20

Example of solids - Sphere

Sphere or Spherical Shell Section:

To build a **sphere**, or a **spherical shell section**, use:

```
G4Sphere(const G4String& pName,
               G4double   pRmin,
               G4double   pRmax,
               G4double   pSPhi,
               G4double   pDPhi,
               G4double   pSTheta,
               G4double   pDTheta )
```



In the picture:

`pRmin = 100, pRmax = 120, pSPhi = 0*Degree, pDPhi = 180*Degree, pSTheta = 0 Degree, pDTheta = 180*Degree`

to obtain a solid with name `pName` and parameters:

How to define a material

- The information of material is also important for the detector simulation
- G4Material class manages this task
- This class contains density and abundance of elements consist of the material
- The material can be constructed from element information, another material or NIST data

Three ways to define the material

- From elements

```
// Build elements of Oxygen and Nitrogen
_elementN  = new G4Element(name = "Nitrogen", symbol = "N", z = 7, a = 14.01 * g / mole);
_elementO  = new G4Element(name = "Oxygen", symbol = "O", z = 8, a = 16.00 * g / mole);

name      = "Air";
density   = 1.29e-3 * g / cm3;
nelements = 2;
// Define material of Air
_air = new G4Material(name, density, nelements);
_air->AddElement(_elementN, 70 * perCent);
_air->AddElement(_elementO, 30 * perCent);
```

Three ways to define the material

- From another material

```
// Defines another material consists of new material
_SiRubber = new G4Material(name, density, nelements);
_B4C = new G4Material(name, density, nelements);

// Create new material and add other materials
_B4CRubber24perCent = new G4Material(name, density, ncomponents = 2);
_B4CRubber24perCent->AddMaterial(_B4C, 24. * perCent);
_B4CRubber24perCent->AddMaterial(_SiRubber, (100. - 24.) * perCent);
```

requires another material defined with elements

Three ways to define the material

- Using NIST database

```
// Glass and Aluminium definition with NIST database
G4Material *Air       = nistManager->FindOrBuildMaterial("G4_AIR");
G4Material *Aluminium = nistManager->FindOrBuildMaterial("G4_Al");
G4Material *Glass     = nistManager->FindOrBuildMaterial("G4_GLASS_PLATE");
```

Gean4 provides pre-defined material database from NIST databook (also you can construct an element from the databook)

The logical volume (LV)

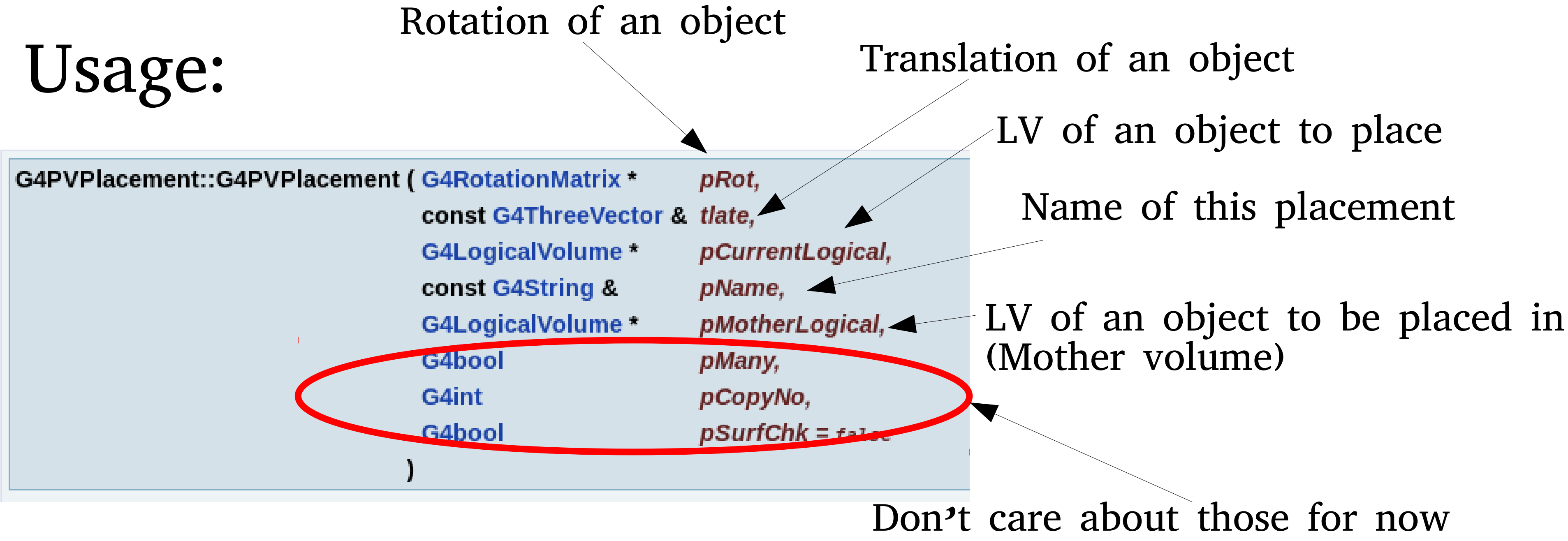
- The logical volume is an image of a thing in Geant4 simulation
- This class equivalent to an object in the real world such as a single crystal or detector
- The logical volume can contain another logical volume, thus they have hierarchical structure

How to place the Logical Volume

- In the real world, an object can be placed inside of another object with translation and rotation
- In Geant4, LV is placed at some position and it can be rotated in another LV
- G4PVPlacement has the information for arrangement of single G4LogicalVolume in another one

G4PVPlacement

- This class manages the 3D transformation of a LV
- Usage:



Example – Let there be world!

- The world MUST not have any rotation, translation, and mother volume

```
//  Must place the World Physical volume unrotated at (0,0,0).
//
G4VPhysicalVolume *worldPV = new G4PVPlacement(0,                 // no rotation
                                               G4ThreeVector(), // at (0,0,0)
                                               worldLV,         // its logical volume
                                               "World",         // its name
                                               0,               // its mother  volume
                                               false,           // no boolean operations
                                               0);              // copy number
```

Example – Put a box in the world

```
// Single Box (G4Box/G4Material) -> G4LogicalVolume -> G4PVPlacement
G4Box *BBox               = new G4Box("BBox", fCellArraySize * fCellSizeX / 2,
                            fCellArraySize * fCellSizeY / 2, totalArraySizeZ / 2.);
G4Material *air           = G4NistManager::Instance()->FindOrBuildMaterial("G4_AIR");
G4LogicalVolume *BBoxLV = new G4LogicalVolume(BBox, air, "BBox");
// Place the box
new G4PVPlacement(0,                        // no rotation
                  G4ThreeVector(0, 0, 0), // at (0,0,0)
                  BBoxLV,                   // its logical volume
                  "BBoxPV",                 // its name
                  worldLV,                  // its mother  volume
                  false,                    // no boolean operations
                  0);                       // copy number
```

Adding CsI(Tl) Crystal

- CsI(Tl) crystal is a common-used scintillation crystal for the particle detection
- Let's make a CsI(Tl) crystal in the world of Geant4!

Properties of CsI(Tl) crystal

Properties	Value
Density [g/cm^3]	4.51
Melting point [K]	894
Thermal expansion coefficient [C^{-1}]	54 x 10^{-6}
Cleavage plane	None
Hardness (Mohs)	2
Hygroscopic	Slightly
Wavelength of emission max. [nm]	550
Lower wavelength cutoff [nm]	320
Refractive index @ emission max.	1.79
Primary decay time [ns]	1000
Light yield [photons/keVγ]	54
Photoelectron yield [% of NaI(Tl)] (for γ-rays)	45

The most important properties!!

Properties of CsI(Tl) crystal

But, there is one more thing...

Properties	**Value**
Density [g/cm^3]	4.51
Melting point [K]	894
Thermal expansion coefficient [C^{-1}]	54×10^{-6}
Cleavage plane	None
Hardness (Mohs)	2
Hygroscopic	Slightly
Wavelength of emission max. [nm]	550
Lower wavelength cutoff [nm]	320
Refractive index @ emission max	1.79
Primary decay time [ns]	1000
Light yield [photons/keVγ]	54
Photoelectron yield [% of NaI(Tl)] (for γ-rays)	45

Properties of CsI(Tl) crystal

That's it!

Properties	Value
Density [g/cm^3]	4.51
Melting point [K]	894
Thermal expansion coefficient [C^{-1}]	54 x 10^{-6}
Cleavage plane	None
Hardness (Mohs)	2
Hygroscopic	Slightly
Wavelength of emission max. [nm]	550
Lower wavelength cutoff [nm]	320
Refractive index @ emission max	1.79
Primary decay time [ns]	1000
Light yield [photons/keVγ]	54
Photoelectron yield [% of NaI(Tl)] (for γ-rays)	45

Definition of CsI(Tl) crystal

- Step1: construct elements (Cs, I)
- Step2: define CsI material with G4Material
- Step3: add the elements into CsI material

```
// Construct the Cesium and Iodine
G4Element *elementCs = nistManager->FindOrBuildElement("Cs");
G4Element *elementI  = nistManager->FindOrBuildElement("I");

// Make the CsI material
G4Material *materialCsI = new G4Material("CsI", density = 4.51 * g / cm3, nelements = 2);
materialCsI->AddElement(elementCs, 1);
materialCsI->AddElement(elementI, 1);
```

Dopping the thallium (Optional)

- Step1: build Tl material w/ single element(Tl)
- Step2: define CsI(Tl) material w/ G4Material
- Step3: add the elements into CsI material

```
// Build the dopant material (Thallium)
G4Material *dopantTl = nistManager->FindOrBuildMaterial("G4_Tl");

// Dopping the thallium into CsI material
G4Material *materialDoppedCsI =
    new G4Material("CsI:Tl", density = 4.51 * g / cm3, nmaterials = 2);
materialDoppedCsI->AddMaterial(dopantTl, concentrationOfTl = 0.001 * perCent);
materialDoppedCsI->AddMaterial(materialCsI, 100 * perCent - concentrationOfTl);
```

Add the optical properties

- It might need to add additional properties to material for additional physical processes
- In the simulation with scintillator, user have to add the optical properties to simulate behavior of optical photon (NOT for gammas)

  Example: Refractive index, Light yield, …
- Those properties are just optional things for detector simulation, so Geant4 provides additional material information class: G4MaterialPropertiesTable

Optical properties for CsI(Tl)

- For the optical photon simulation, refractive index must be included in material property
- Also, various factor is required for scintillation process!
- In our simulation, three values are needed for CsI(Tl)

Refractive index @ emission max	1.79
Primary decay time [ns]	1000
Light yield [photons/keVγ]	54

Tell optical properties to Geant4

```
// Declear the M.P.T.
G4MaterialPropertiesTable *CsI_mt = new G4MaterialPropertiesTable();

// Define the energy region for optical properties
G4double CsI_Energy[] = {1.0 * eV, 2.5 * eV, 5.0 * eV};
// Relative intensity of scintillation light over specified energy range
G4double CsI_SCINT[] = {1., 1., 1.};
// Refractive index of CsI(Tl) crystal
G4double CsI_RIND[] = {1.79, 1.79, 1.79};
// Absorption length of CsI(Tl) crystal for optical photons
G4double CsI_ABSL[] = {10. * m, 10. * m, 10. * m};
```

Tell optical properties to Geant4

```
CsI_mt->AddProperty("FASTCOMPONENT", CsI_Energy, CsI_SCINT, 3);
CsI_mt->AddProperty("RINDEX", CsI_Energy, CsI_RIND, 3);
CsI_mt->AddProperty("ABSLENGTH", CsI_Energy, CsI_ABSL, 3);

CsI_mt->AddConstProperty("SCINTILLATIONYIELD", 54. / keV);
CsI_mt->AddConstProperty("RESOLUTIONSCALE", 1.0);
CsI_mt->AddConstProperty("FASTTIMECONSTANT", 1000. * ns);
CsI_mt->AddConstProperty("YIELDRATIO", 1.0);

// Attach material properties table to each materials (CsI(Tl))
materialCsI->SetMaterialPropertiesTable(CsI_mt);
materialDoppedCsI->SetMaterialPropertiesTable(CsI_mt);
```

Shape CsI(Tl) crystal

- The box shape

```
// Define Box / Tube for CsI(Tl) crystal
G4Box *CsIBox        = new G4Box("CsI_Box",
                           fCellSizeX / 2.,  // X-dimension of the Box
                           fCellSizeY / 2.,  // Y-dimension of the Box
                           fCellSizeZ / 2.); // Z-dimension of the Box
```

- The tube shape

```
// Define Tube for CsI(Tl) crystal
G4Tubs *CsITube      = new G4Tubs("CsI_Tube",
                              0,           // Minimum radius of the tube
                              fCellSizeX, // Maximum radius of the tube
                              fCellSizeZ, // Total half length of the tube
                              0,           // Starting angle of the tube
                              2 * M_PI);   // Ending angle of the tube
```

Shape CsI(Tl) crystal

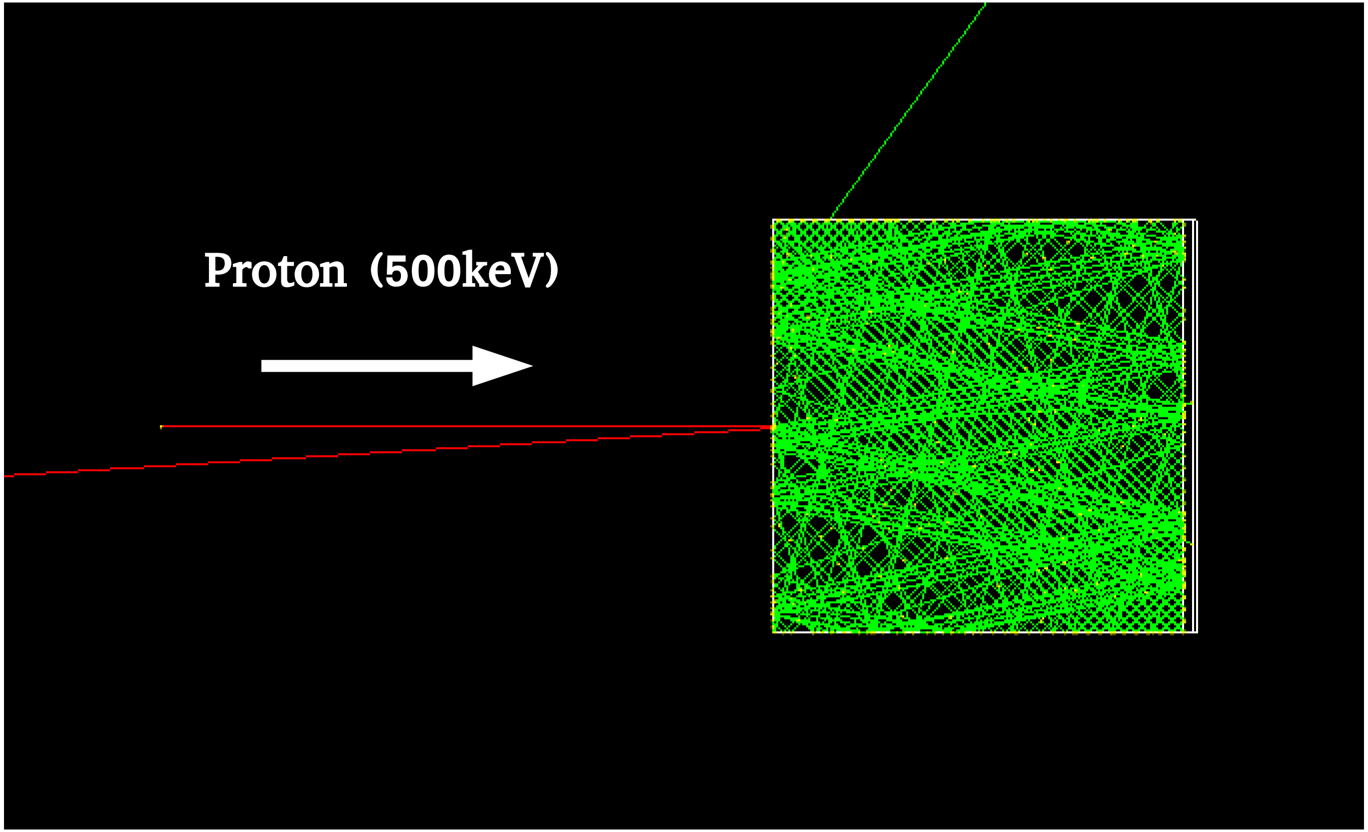
- The orb (Complete sphere) shape

```
// Define Sphere for CsI(Tl) crystal
G4Sphere *CsISphere = new G4Sphere("CsI_Sphere",
                                   0,            // Minimum radius of the sphere
                                   fCellSizeX, // Maximum radius of the sphere
                                   0,            // Starting phi (XY plane) angle of the sphere
                                   2 * M_PI,     // Ending phi (XY plane) angle of the sphere
                                   0,            // Starting theta angle of the sphere
                                   M_PI);        // Ending theta angle of the sphere
```

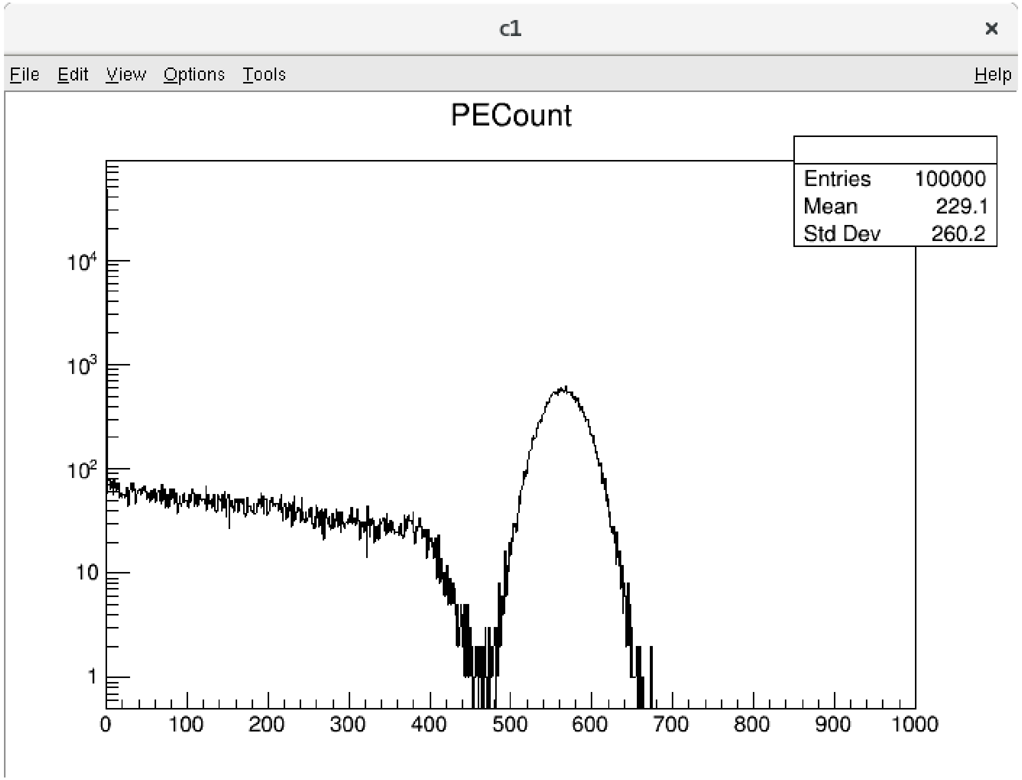
Definition & Placing CsI(Tl)

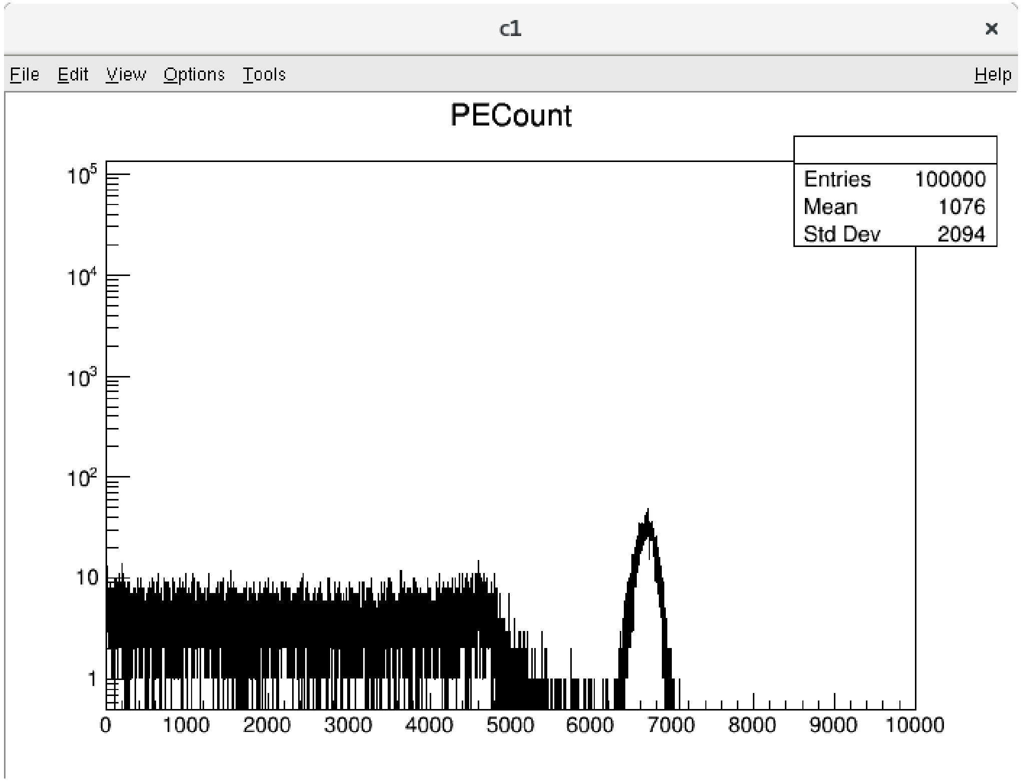
```
// Add logical volume and place it in the world
G4LogicalVolume *CsILV = new G4LogicalVolume(CsIBox, CsI, "CsI_LV");
new G4PVPlacement(nullptr, G4ThreeVector(), CsILV, "CsI_PV", worldLV, false, 0);
```

Example screenshot



Example photoelectron count





- Using BGO crystal
- Using CsI(Tl) crystal

A QUEST FOR STUDENTS

- source_original: original source code using BGO crystal
- source_modified: modified source code for adding CsI(Tl) material, but it is imcomplete

- QUEST: Complete the missing part of source_modified and change the material of crystal to CsI(Tl)!

Homeworks

- Make the dimension of scintillation crystal to 30 cm x 30 cm x 30 cm
- Shoot 1GeV, 10GeV, 20GeV proton
- Calibrate your detector
- Obtain energy distributions measured with your detector
- Try to plot energy resolution and linearity plot
- Find light yield in # of p.e. / GeV

Appendix

How to compile BGOS

- Make a directory for building BGOS
  Example) mkdir building
- Perform the configuring for building
  Example) cmake /path/to/source
- Execute the compile process
  Example) make -j 2

How to merge output ROOT files

- using hadd utility
  Example) hadd [output file name] *root
  (in the directory of BGOS output)

- Warning: [output file name] should be absent in working directory!