

# Magnetic Field

---

# Geometry definition class

1. Inherit from the G4VUserDetectorConstruction class to create your own geometry definition class.
2. Describe your geometry within the "construct" function.
3. Create an object of the geometry definition class and register it with G4RunManager.

## Header file

```
//+++++  
// Geometry.hh  
//+++++  
#ifndef Geometry_h  
#define Geometry_h 1  
#include "G4VUserDetectorConstruction.hh"  
class G4VPhysicalVolume;  
//-----  
class Geometry : public G4VUserDetectorConstruction  
//-----  
{  
public:  
    Geometry();  
    ~Geometry();  
    G4VPhysicalVolume* Construct();  
private:  
    G4VPhysicalVolume* ConstructDetector();  
};  
#endif
```

Your own geometry definition class

The base class provided by Geant4 for geometry definition

Write your geometry description in this function

# Magnetic Field

---

Use G4MagneticField to implement the magnetic field in Geant4

Uniform Field :

Use G4UniformMagField class

```
G4MagneticField* magField =  
    new G4UniformMagField{ G4ThreeVector(1.*Tesla,0.,0.) };
```

Non Uniform Field :

Prepare a new class inherit from G4MagneticField with  
GetFieldValue method

```
class MyField : G4MagneticField {.....};  
    .....  
void MyField::GetFieldValue(  
    const G4double Point[4], G4double *field) const {  
    .....  
}
```

- Point[0..2] is space (x,y,z:global coordinate system)、Point[3] is time
- field[0..2] is magnetic field value at these space (x,y,z)

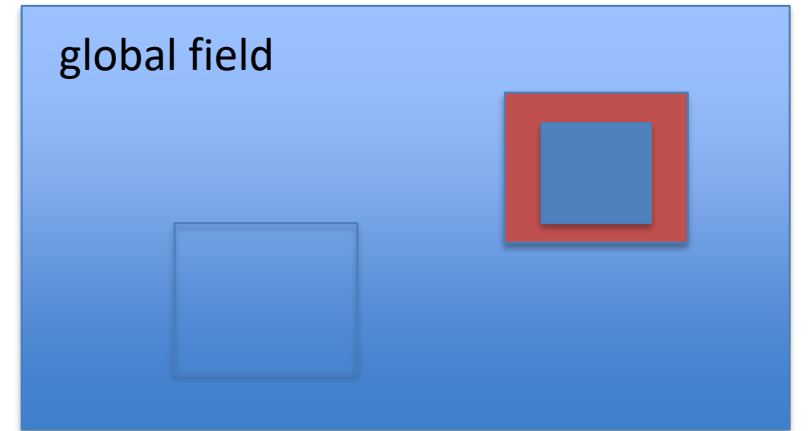
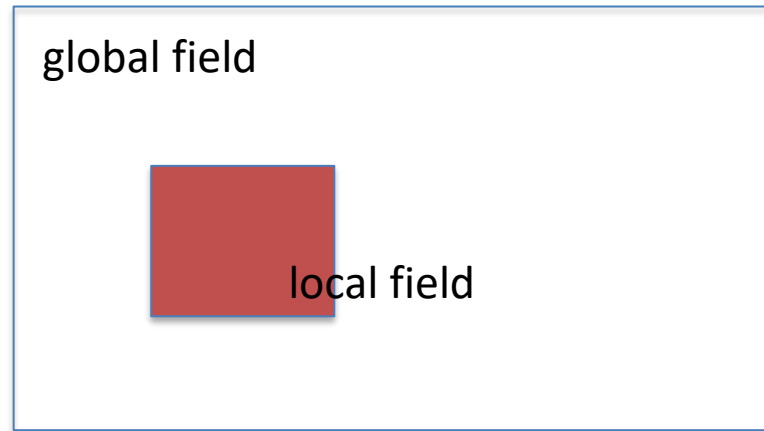
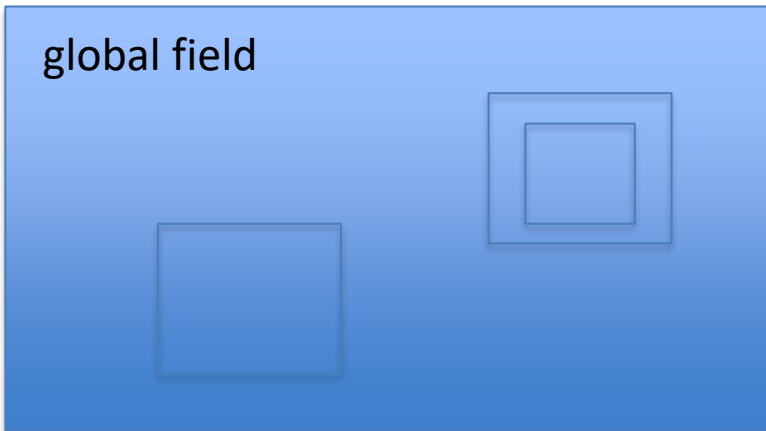
# Logical Volume definition for magnetic field

---

Implementation the G4MagneticField to user geometry

The magnetic field is applied through G4FieldManager

- to World Volume - Global field
- inside the geometry components - Local fields  
local field and global field can be coexistence.  
local field has prior in such case



# Global Field

---

## ■ Global Field

- Set G4MagneticField to G4FieldManager through G4TransportationManager

```
G4FieldManager* globalFieldMgr =  
    G4TransportationManager::GetTransportationManager()->GetFieldManager();  
globalFieldMgr->SetDetectorField(magField);  
globalFieldMgr->CreateChordFinder(magField);
```

## ■ G4GlobalMagFieldMessenger

- creates the global uniform magnetic field

```
void MyDetectorConstruction::CreateSDandField {  
    G4ThreeVector fieldValue = G4ThreeVector();  
    G4GlobalMagFieldMessenger* magFieldMessenger = new G4GlobalMagFieldMessenger(fieldValue);  
    G4AutoDelete::Register(myFieldMessenger);  
}
```

# Local Field

---

## ■ Local Field

create new G4FieldManager and set G4MagneticField object of local field in constructor  
set the G4FieldManager through setFieldManager method of logical volume

```
G4FieldManager* localFieldMgr = new G4FieldManager{ magField };  
logicalVolume->setFieldManager(localFieldMgr, true);
```

- magnetic field can be apply to arbitrary region of geometry components using local field
- when local field is applied, global field is ignored
- the second argument of setFieldManager is
  - true: magnetic field is apply to all daughter volumes
  - false: magnetic field is apply to the daughter volumes without G4FieldManager

```

void DetectorConstruction::ConstructSDandField(){
    ...

    //
    // Magnetic Field for MRI
    //
    G4String fieldnamey_MRI = "MRILV";
    auto* magFieldmri = new GaletMagneticField(fieldnamey_MRI);
    magFieldmri->SetFieldY(1.0*tesla);
    magFieldmri->SetFieldX(0.0*tesla);
    //
    SetMagneticField(fieldnamey_MRI, magFieldmri);

}

```

```

void DetectorConstruction::SetMagneticField(const G4String& name_lv,
    G4MagneticField* field,
    G4bool forceToAllDaughters){

```

```

    G4LogicalVolume* fieldLV = G4LogicalVolumeStore::GetInstance()->GetVolume(name_lv);
    if ( fieldLV ){
        G4FieldManager* fieldManager = fieldLV->GetFieldManager();
        if ( fieldManager ) fieldManager->SetDetectorField(field);
        else fieldManager = new G4FieldManager(field);

        fieldLV->SetFieldManager(fieldManager,forceToAllDaughters);

```

...