

# DIGITS

---

Cern Geant4 Tutorial

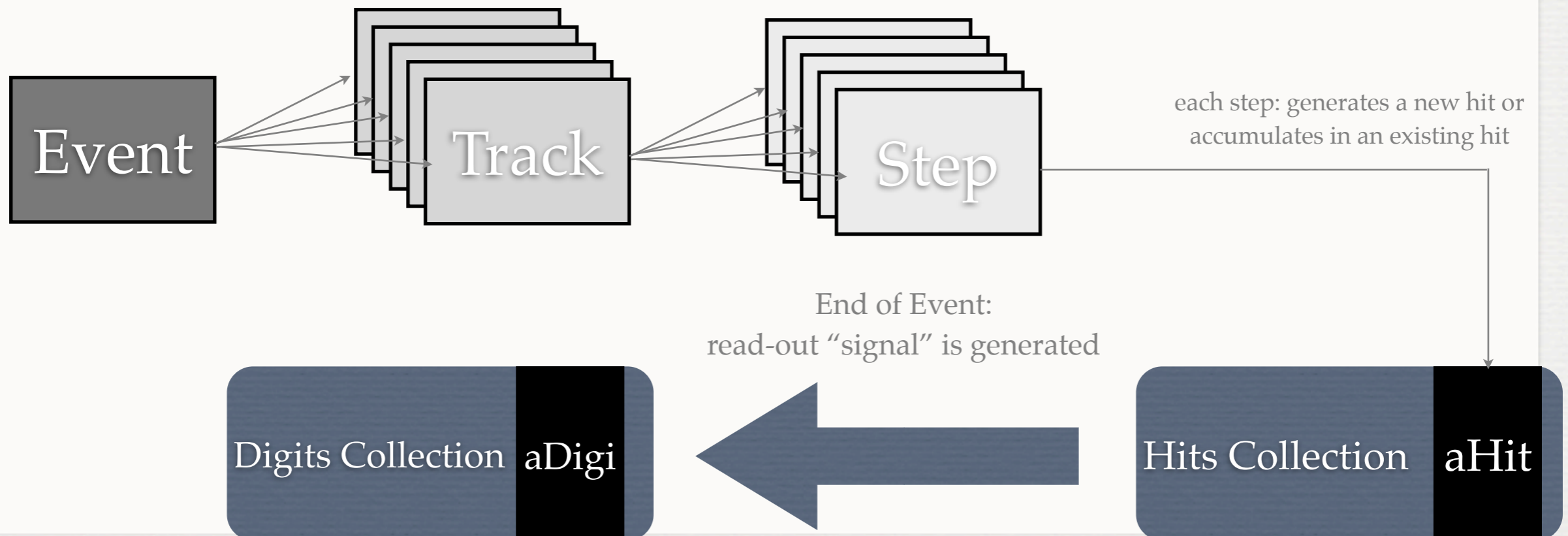
February 19, 2010



**Geant 4**

# Hits Vs Digits

- Hits are a “snapshot” of the physical interaction of a track (step) or an accumulation of interactions of tracks in the sensitive region of the detector, thus hits represent the “true” energy deposited in the detector
- Digits are instead intended to be used to simulate the process of reading-out of the signal: for example “true” energy is transformed into collected charge, electronic noise can be applied together with all instrumental effects



# Digitizer

- Digitizer is identified by name and has to be registered to the DigiManager singleton
- The SiDigitizer class inherits from G4VDigitizerModule base class and implements the Digitize() method
  - Warning: this method has to be called explicitly at the end of the event

```
EventAction::EventAction()  
00026 {  
00027     //We build the digitization module  
00028     SiDigitizer* digitizer = new SiDigitizer("SiDigitizer");  
00029     G4DigiManager * digiManager = G4DigiManager::GetDMpointer();  
00030     digiManager->AddNewModule( digitizer );  
00031 }  
  
00048 void EventAction::EndOfEventAction(const G4Event* anEvent)  
00049 {  
00050     //Digitize!!  
00051     G4DigiManager * digiManager = G4DigiManager::GetDMpointer();  
00052     SiDigitizer* digiModule = static_cast<SiDigitizer*>( digiManager->  
>FindDigitizerModule("SiDigitizer") );  
00055     digiModule->Digitize();
```

# Digitizer

- Digitizer is identified by name and has to be registered to the DigiManager singleton
- The SiDigitizer class inherits from G4VDigitizerModule base class and implements the Digitize() method
  - Warning: this method has to be called explicitly at the end of the event

```
EventAction::EventAction()  
00026 {  
00027     //We build the digitization module  
00028     SiDigitizer* digitizer = new SiDigitizer("SiDigitizer");  
00029     G4DigiManager * digiManager = G4DigiManager::GetDMpointer();  
00030     digiManager->AddNewModule( digitizer );  
00031 }  
  
00048 void EventAction::EndOfEventAction(const G4Event* anEvent)  
00049 {  
00050     //Digitize!!  
00051     G4DigiManager * digiManager = G4DigiManager::GetDMpointer();  
00052     SiDigitizer* digiModule = static_cast<SiDigitizer*>( digiManager->  
-FindDigitizerModule("SiDigitizer") );  
00055     digiModule->Digitize();
```

# Digitizer

- Digitizer creates the digit collection, similarly to hits, these are identified by a collection name

```
00021 SiDigitizer::SiDigitizer(G4String aName) :  
00022   G4VDigitizerModule(aName)  
00043 {  
00044     collectionName.push_back( digiCollectionName );  
00047 }
```

- Digits are created and added to the collection:

```
00049 void SiDigitizer::Digitize()  
00050 {  
00052   SiDigiCollection * digiCollection = new SiDigiCollection("SiDigitizer",digiCollectionName);  
00053   //Create a empty collection with one digits for each strip  
00055   const G4int numPlanes = 3; //Number of Si detectors  
00056   const G4int numStrips = 48; //Number of strip per plane  
00067   for ( G4int plane = 0 ; plane < numPlanes ; ++plane ) {  
00068     for ( G4int strip = 0 ; strip < numStrips ; ++strip )  
00069       {  
00070         SiDigi* newDigi = new SiDigi(plane,strip);  
00072         digiCollection->insert(newDigi);  
00073       }  
00074   }  
00129   StoreDigiCollection(digiCollection);
```

# Digitizer

- Digitizer creates the digit collection, similarly to hits, these are identified by a collection name

```
00021 SiDigitizer::SiDigitizer(G4String aName) :  
00022     G4VDigitizerModule(aName)  
00043 {  
00044     collectionName.push_back( digiCollectionName );  
00047 }
```

- Digits are created and added to the collection:

```
00049 void SiDigitizer::Digitize()  
00050 {  
00052     SiDigiCollection * digiCollection = new SiDigiCollection("SiDigitizer",digiCollectionName):  
00053     //Create a empty collection with one digits for each strip  
00055     const G4int numPlanes = 3; //Number of Si detectors  
00056     const G4int numStrips = 48; //Number of strip per plane  
00067     for ( G4int plane = 0 ; plane < numPlanes ; ++plane ) {  
00068         for ( G4int strip = 0 ; strip < numStrips ; ++strip )  
00069             {  
00070                 SiDigi* newDigi = new SiDigi(plane,strip);  
00072                 digiCollection->insert(newDigi);  
00073             }  
00074     }  
00129     StoreDigiCollection(digiCollection);
```

# Digitizer

- Digitizer creates the digit collection, similarly to hits, these are identified by a collection name

```
00021 SiDigitizer::SiDigitizer(G4String aName) :  
00022     G4VDigitizerModule(aName)  
00043 {  
00044     collectionName.push_back( digiCollectionName );  
00047 }
```

- Digits are created and added to the collection:

```
00049 void SiDigitizer::Digitize()  
00050 {  
00052     SiDigiCollection * digiCollection = new SiDigiCollection("SiDigitizer",digiCollectionName);  
00053     //Create a empty collection with one digits for each strip  
00055     const G4int numPlanes = 3; //Number of Si detectors  
00056     const G4int numStrips = 48; //Number of strip per plane  
00067     for ( G4int plane = 0 ; plane < numPlanes ; ++plane ) {  
00068         for ( G4int strip = 0 ; strip < numStrips ; ++strip )  
00069             {  
00070                 SiDigi* newDigi = new SiDigi(plane,strip);  
00072                 digiCollection->insert(newDigi);  
00073             }  
00074     }  
00129     StoreDigiCollection(digiCollection);
```

# Digitizer

- Digitizer creates the digit collection, similarly to hits, these are identified by a collection name

```
00021 SiDigitizer::SiDigitizer(G4String aName) :  
00022     G4VDigitizerModule(aName)  
00043 {  
00044     collectionName.push_back( digiCollectionName );  
00047 }
```

- Digits are created and added to the collection:

```
00049 void SiDigitizer::Digitize()  
00050 {  
00052     SiDigiCollection * digiCollection = new SiDigiCollection("SiDigitizer",digiCollectionName);  
00053     //Create a empty collection with one digits for each strip  
00055     const G4int numPlanes = 3; //Number of Si detectors  
00056     const G4int numStrips = 48; //Number of strip per plane  
00067     for ( G4int plane = 0 ; plane < numPlanes ; ++plane ) {  
00068         for ( G4int strip = 0 ; strip < numStrips ; ++strip )  
00069             {  
00070                 SiDigi* newDigi = new SiDigi(plane,strip);  
00072                 digiCollection->insert(newDigi);  
00073             }  
00074     }  
00129     StoreDigiCollection(digiCollection);
```



# Retrieving Digits

- 📌 Digits collection can be retrieved, by name, **via the DigiManager singleton**

```
00051 G4DigiManager * digiManager = G4DigiManager::GetDMpointer();  
00062 G4int digiCollID = digiManager->GetDigiCollectionID( digitsCollName );  
00063 const SiDigiCollection* digits = static_cast<const SiDigiCollection*>  
( digiManager->GetDigiCollection(digiCollID) );
```

- 📌 Remember retrieval is always a two-step process:

name (string)  $\Rightarrow$  ID (integer)  $\Rightarrow$  collection (pointer)

- 📌 **Since IDs do not change during a run you can (should) optimize your code: do the first search only once**

# Note

- 📌 Hits, being a user defined class, can contain whatever information is needed, but it is a good idea to keep detector read-out information separated (in digits) from true energy deposits (hits):
  - It is possible to define more than one digit collection from the same hits collection: different read-out parameters can be tried at the same time
  - Simulate detector malfunctioning: what is the effect of 1% dead cells in my calorimeter? What is the energy lost?