# Introduction to the IST hit structure & storage scheme

## StIstHit overview

The IST consists of one layer of 24 ladders with 6 sensors each. The StIstHit class therefore provides the two methods getLadder() and getSensor() which return the appropriate 'hardware' coordinate. Note that the numbering starts at 1. In addition the class has member functions to return the hit local position, max Adc time bin and the cluster size, respectively. The IST hits are stored in a tree organized similar to the SSD (Fig. 1).

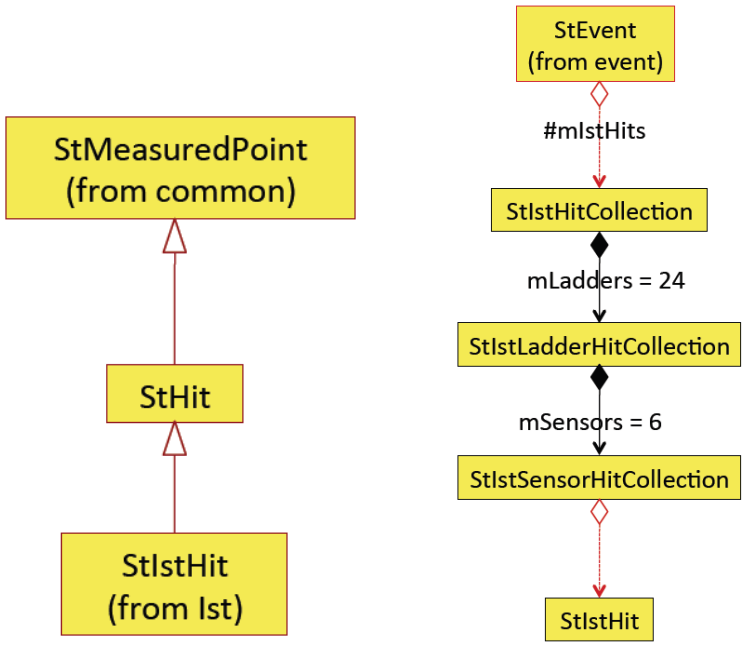


Fig. 1 **Left**: StIstHit class and its superclass. **Right**: Class diagrams of the IST hit storage scheme: ladder/sensor

The hits are stored per sensor. The following is the equivalent example for the IST hit storage scheme:

```
const int iladder = 2;
const int isensor = 3;
cout      << "The IST has "
          << event->istHitCollection()->numberOfHits()
          << " hits" << endl;
StSPtrVecIstHit& theHits = event->istHitCollection()->ladder(iladder)->sensor(isensor).hits();
cout      << "ladder " << iladder
          << ", sensor " << isensor << " contains "
          << theHits.size() << " hits" << endl;
for (int i=0; i<theHits.size(); i++) {
    cout << theHits[i]->position() << endl;
    cout << theHits[i]->charge() << endl;
    assert(theHits[i]->ladder() == iladder);
    assert(theHits[i]->sensor() == isensor);
}
```

# StIstHit class

**Summary**

**Synopsis**

```
#include "StIstHit.h"
class StIstHit;
```

**Description**

**Related Classes**

**Public Constructors**

```
StIstHit( unsigned char ladder=-1, unsigned char sensor=-1,
    float Charge=0., float ChargeErr=0., unsigned char maxTB=0,
    unsigned char nRawHits=1, unsigned char nRawHitsZ=0,
    unsigned char nRawHitsRPhi=0, float meanColumn=0., float meanRow=0);
StIstHit( const StThreeVectorF& position, const StThreeVectorF& error,
    unsigned int hwPosition, float charge, unsigned char trackRefCount = 0);
```

**Public Member Functions**

```
unsigned char    getLadder() const; //return ladder number in the range 1-24
unsigned char    getSensor() const; //return sensor number in the range 1-6
float            getMeanColumn() const; //return cluster average column
float            getMeanRow() const; //return cluster average row
unsigned char    getMaxTimeBin() const; //max Adc (charge) time bin index
float            getChargeErr() const;
unsigned char    getClusteringType() const; //clustering algorithm flag
unsigned char    getNRawHits() const; //cluster size
unsigned char    getNRawHitsZ() const; //cluster size in Z direction
unsigned char    getNRawHitsRPhi() const; //cluster size in r-phi direction
float        localPosition(unsigned int ) const; //local position
float        localPositionErr(unsigned int) const; //local position uncertainty

void     setMaxTimeBin(unsigned char tb);
void     setChargeErr(float chargeErr);
void     setClusteringType(unsigned char clusteringType);
void     setNRawHits(unsigned char nRawHits);
void     setNRawHitsZ(unsigned char nRawHitsZ);
void     setNRawHitsRPhi(unsigned char nRawHitsRPhi);
void     setLocalPosition(float, float, float);
```

**Members**

```
UChar_t mMaxTimeBin;          // max charge time bin
Float_t   mChargeErr;         // charge uncertainty
UChar_t mClusteringType;      // clustering algorithm size
UChar_t mNRawHits;            // nRawHits: cluster size
UChar_t mNRawHitsZ;           // cluster size in Z direction
UChar_t mNRawHitsRPhi;        // cluster size in r-phi direction
Float_t   mLocalPosition[3];  // local X, 0. or defined by sensor
                              // surface correction, local Z
StDetectorId mDetectorId;
```

# StIstHitCollection class

**Summary**

**Synopsis**

```
#include "StIstHitCollection.h"
class StIstHitCollection;
```

**Description**

**Related Classes**

**Public Constructors**

```
StIstHitCollection();
```

**Public Member Functions**

```
bool addHit(StIstHit*);
unsigned int numberOfHits() const;
unsigned int numberOfLadders() const;
StIstLadderHitCollection* ladder(unsigned int);
const StIstLadderHitCollection* ladder(unsigned int) const;
```

**Members**

```
enum { mNumberOfLadders = 24 };
StIstLadderHitCollection mLadders[mNumberOfLadders];
```

# StIstLadderHitCollection class

**Summary**

**Synopsis**

```
#include "StIstLadderHitCollection.h"
class StIstLadderHitCollection;
```

**Description**

**Related Classes**

**Public Constructors**

```
StIstLadderHitCollection();
```

**Public Member Functions**

```
unsigned int numberOfHits() const;
unsigned int numberOfSensors() const;
StIstSensorHitCollection* sensor(unsigned int);
const StIstSensorHitCollection* sensor(unsigned int) const;
```

**Members**

```
enum { mMaxNumberOfSensors = 6 };
StIstSensorHitCollection    mSensors[mMaxNumberOfSensors];
```

# StIstSensorHitCollection class

**Summary**

**Synopsis** #include "StIstSensorHitCollection.h"

class StIstSensorHitCollection;

**Description**

**Related Classes**

**Public Constructors** StIstSensorHitCollection ();

**Public Member Functions**

StSPtrVecIstHit& hits();

const StSPtrVecIstHit& hits() const;

**Member**

StSPtrVecIstHit mHits;

## Files to be reviewed:

**Location:**
**`/star/u/ypwang/disk02/istOfflineChain/istHitStructure_review`**

1) IST hit structure and its containers can be found: StRoot/StEvent/
   StIstHit.cxx
   StIstHit.h
   StIstHitCollection.cxx
   StIstHitCollection.h
   StIstLadderHitCollection.cxx
   StIstLadderHitCollection.h
   StIstSensorHitCollection.cxx
   StIstSensorHitCollection.h

2) IST hit maker can be found: StRoot/StIstHitMaker/
   StIstHitMaker.h
   StIstHitMaker.cxx

3) IST global constants can be found: StRoot/StIstUtil/
   StIstConsts.h

4) several basic classes are updated locally for IST:
   StRoot/StEvent/StEvent.h : lines 206, 268-269, 339
   StRoot/StEvent/StEvent.cxx : lines 223, 783-797, 1247-1251, 1367, 1393
   StRoot/StEvent/StEventType.h : lines: 278-281
   StRoot/StEvent/StContainers.h : lines 162, 220
   StRoot/StEvent/StContainers.cxx : lines 146, 204
   StRoot/StEvent/StEventClusteringHints.cxx : lines 165, 189

   StRoot/StEventUtilities/StEventHitIter.cxx : lines 20, 450-525, 750
   StRoot/StEventUtilities/StuFixTopoMap.cxx: lines 126-131