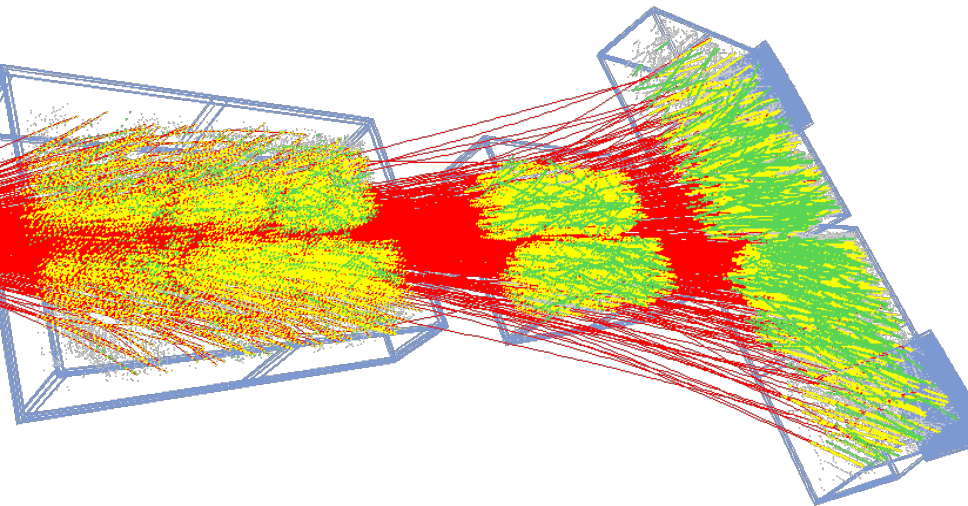


# Data Analysis with SHINE Offline



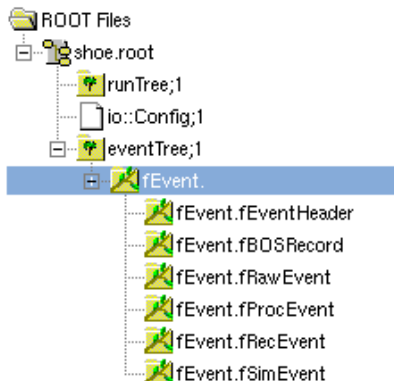
M. Unger for the SHINE developers

# Layout of the SHINE Offline Event (SHOE)

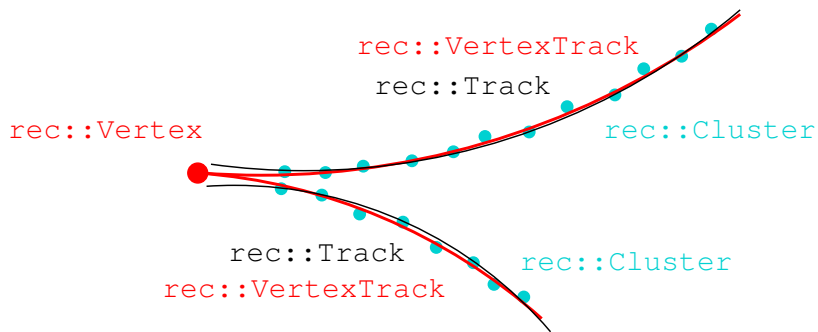
Event consists of

- **EventHeader**  
(id, run, time, ...)
- **BOSRecord**  
(BOS-banks)
- **RawEvent**  
(decoded BOS-banks, trigger, WFA, scalers, ...)
- **ProcEvent**  
(processed raw data)
- **RecEvent**  
(tracks, vertices, clusters...)
- **SimEvent**  
(tracks, vertices, hits...)

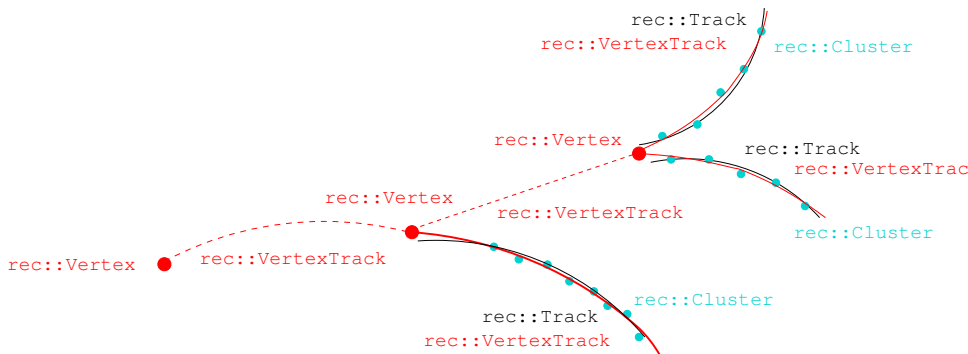
event view in ROOT's TBrowser:



# Tracks, Vertices, Vertex-Tracks



# Tracks, Vertices, Vertex-Tracks



## How to access SHOE Objects - a) looping

- most objects are stored as plain STL lists
- e.g. in `RecEvent`:

```
std::list<rec::Cluster> fClusterList;  
std::list<rec::Track> fTrackList;  
std::list<rec::VertexTrack> fVertexTrackList;  
std::list<rec::Vertex> fVertexList;  
std::list<rec::TOFMass> fTOFMassList;
```

- access via

```
RecEvent::Begin<T> ()  
RecEvent::End<T> ()  
RecEvent::GetNumberOf<T> ()
```

where T is the class name (Cluster, Vertex...)

## How to access SHOE Objects - a) looping

Example:

loop over all vertices and find  $\Xi$ -decay vertices:

```
typedef list<Vertex>::const_iterator VtxIterator;
for (VtxIterator iter = recEvent.Begin<Vertex>();
     iter != recEvent.End<Vertex>(); ++iter) {

    const Vertex& vertex = *iter;

    if (vertex.GetType() == VertexConst::eXi) {

        // ... do something

    }

}
```

## How to access SHOE Objects - b) object indices

SHOE equivalent of persistent pointers:

- each object has a unique index with which it can be directly accessed (random access)
- implemented as

```
Index<T> objectIndex;
```

where T is a class name (Cluster, Vertex...)

- retrieve object by index:

```
Index<Vertex> vertexIndex = ... ;  
const Vertex& vertex = recEvent.Get(vertexIndex);
```

# How to access SHOE Objects - b) object links

Vertex 1

Vertex 0

VertexTrack 0

VertexTrack 3

VertexTrack 1

VertexTrack 4

VertexTrack 2

Vertex 3

Track 0

Track 1

Track 2

VertexTrack 5

Track 3

VertexTrack 6

Track 4

Track 5

Cluster 0

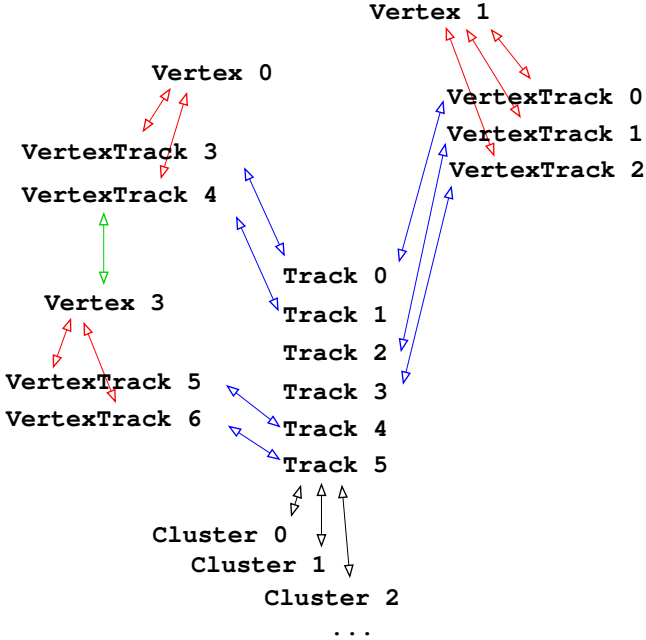
Cluster 1

Cluster 2

...



# How to access SHOE Objects - b) object links



## How to access SHOE Objects - b) object indices

Example: `VertexTrack` → `Track` → `Cluster`.

```
// start with vertex track
const VertexTrack& vertexTrack = ....;

// could be a V0, so better check if Track exists
if (vertexTrack.HasTrack()) {
    // get Track from RecEvent via Index<Track>
    const Index<Track> trackIndex = vertexTrack.GetTrackIndex();
    const Track& track = recEvent.Get(trackIndex);

    // loop over cluster indices associated to this track
    for (ClusterIndexIterator iter = track.ClustersBegin();
         iter != track.ClustersEnd(); ++iter) {
        // get cluster from RecEvent given Index<Cluster>
        const Cluster& cluster = recEvent.Get(*iter);

        // ... do something
    }
}
```

## How to access SHOE Objects - b) object indices (C++11)

Example: `VertexTrack` → `Track` → `Cluster`.

```
// start with vertex track
const auto& vertexTrack = ....;

// could be a V0, so better check if Track exists
if (vertexTrack.HasTrack()) {
    // get Track from RecEvent via Index<Track>
    const auto trackIndex = vertexTrack.GetTrackIndex();
    const auto& track = recEvent.Get(trackIndex);

    // loop over cluster indices associated to this track
    for (auto iter = track.ClustersBegin();
         iter != track.ClustersEnd(); ++iter) {
        // get cluster from RecEvent given Index<Cluster>
        const auto& cluster = recEvent.Get(*iter);

        // ... do something
    }
}
```

# How to Remove SHOE Objects

- set ROOT branch 'off' in `ShineFileExporter.xml`
  - preferred method to exclude a certain class of objects from streaming (e.g. all clusters)
- same can be done while reading or writing using `EventFile`
- removal of unwanted objects, e.g.

```
if (vertexTrack.GetStatus() != 0 )  
    recEvent.Erase(vertexTrack.GetIndex());
```

- easy to produce pre-selected data, i.e. your own custom nanoDST suited for your analysis needs.
- see `nanoSHOE.cc` in example area

## Comparison of File Sizes (run-009910x002)

raw data	1 GByte
SHOE DST*	74 MByte
SHOE miniDST**	3.2 MByte
SHOE nanoDST***	0.9 MByte

\* everything except raw data and covariance matrices

\*\* as DST, but without clusters

\*\*\* as miniDST, but only main vertex tracks with status==0

# File Formats

EventFile, EventFileChain and EventFileReaderSG can read and convert to SHOE

- SHOE (of course)
- BOS (raw data)
- DSPACK (legacy)
- HepMC
- unigen
- ASCII MC (proprietary particle list for GNA61)

and SHOE can be exported to


- SHOE (e.g. event selection)
- BOS (e.g. for embedding)
- ASCII MC (e.g. to feed GNA61)



# How to get started

- Documentation:
  - Shine Offline TWiki
  - doxygen
- example programs\*
  - DS2SHOE
  - compareShineFiles
  - nanoSHOE
  - selectEvents
  - lambdaAnalysis
  - recEventLoop
  - simpleSelection
  - simulationAnalysis

TWiki > NA61 Web > SHINEOfflineHome (22-Jan-2013, AndrasLaszlo)

 [Edit](#) [Attach](#) [PDF](#)



## SHINE Offline Framework

SHINE Offline is a unified Framework for data analysis, event reconstruction, detector simulation and online monitoring. Please, address questions to the [na61-soft mailing list](#).

### Beginners

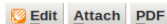
- [Getting started on Ixplus](#)
- [Data Structures](#) (a.k.a. SHOE, the SHINE Offline Event)
- [Data analysis](#)
- [Frequently Asked Questions](#)
- [EventBrowser](#)
- [Releases](#)
- [Class documentation](#)
- [Bug reports](#)
- [Installation](#)
- [Documentation](#) including articles and talks

### Advanced users

\* located in `install/share/shine-offline/apps/Examples/SimpleAnalysis`

# How to get started

TWiki > ■ NA61 Web > SHINEOfflineHome > SHOAnalysis  
(21-Nov-2012, MichaelUnger)



## ■ Documentation:

- ▶ Shine Offline TWiki
- ▶ doxygen

## ■ example programs\*

- ▶ DS2SHOE
- ▶ compareShineFiles
- ▶ nanoSHOE
- ▶ selectEvents
- ▶ lambdaAnalysis
- ▶ recEventLoop
- ▶ simpleSelection
- ▶ simulationAnalysis

On this page:

- ↓ [Data Analysis](#)
  - ↓ [Compiled Programs](#)
    - ↓ [Skeleton Programs](#)
    - ↓ [Geometry package](#)
    - ↓ [Units](#)
    - ↓ [Relativistic Calculations](#)
    - ↓ [Tracking](#)
  - ↓ [ROOT macros](#)
  - ↓ [Quick Plots with TTree::Draw\(\)](#)
  - ↓ [File Inspection using TTree::Scan\(\)](#)
    - ↓ [simple content browsing](#)
    - ↓ [following logical connections inside the event](#)
  - ↓ [Adding the SHINE Event library to ROOT:](#)

\* located in `install/share/shine-offline/apps/Examples/SimpleAnalysis`



# How to get started

- Documentation:
  - ▶ Shine Offline TWiki
  - ▶ doxygen
- example programs\*
  - ▶ DS2SHOE
  - ▶ compareShineFiles
  - ▶ nanoSHOE
  - ▶ selectEvents
  - ▶ lambdaAnalysis
  - ▶ recEventLoop
  - ▶ simpleSelection
  - ▶ simulationAnalysis

## Public Member Functions

<code>ClusterListIterator</code>	<code>ClustersBegin () const</code> cluster indices
<code>ClusterListIterator</code>	<code>ClustersEnd () const</code>
<code>void</code>	<code>Detach (RecEvent &amp;recEvent)</code> detach all indices
<code>int</code>	<code>GetCharge () const</code> particle charge
<code>const utl::CovarianceMatrix &amp;</code>	<code>GetCovarianceMatrix () const</code>
<code>const utl::Point &amp;</code>	<code>GetFirstPointOnTrack () const</code> first measured point on track
<code>double</code>	<code>GetFitProbability () const</code> P( $\chi^2$ , Ndf) of momentum fit.
<code>Index&lt; Track &gt;</code>	<code>GetIndex () const</code> get object index
<code>const utl::Point &amp;</code>	<code>GetLastPointOnTrack () const</code> last measured point on track
<code>const utl::Vector &amp;</code>	<code>GetMomentum () const</code> fitted momentum at point "MomentumPoint"
<code>const utl::Point &amp;</code>	<code>GetMomentumPoint () const</code> point at which momentum was fitted
<code>unsigned int</code>	<code>GetNumberOfClusters (const ETPC id) const</code> number of clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfClusters () const</code>
<code>unsigned int</code>	<code>GetNumberOfdEdXClusters (const ETPC id) const</code> number of dEdX clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfFitClusters (const ETPC id) const</code> number of fitted clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfSimVertexTracks () const</code>
<code>unsigned int</code>	<code>GetNumberOfVertexTracks () const</code>
<code>unsigned int</code>	<code>GetPotentialNumberOfClusters (const ETPC id) const</code> potential number of clusters in TPC id
<code>int</code>	<code>GetStatus () const</code> get status (e.g. NA49 iFlag)
<code>double</code>	<code>GetTruncatedMeanCharge (const ETPC id) const</code> truncated mean of cluster charges along track
<code>double</code>	<code>GetTruncatedMeanChargeVariance (const ETPC id) const</code> variance of truncated mean of cluster charges

\* located in `install/share/shine-offline/apps/Examples/SimpleAnalysis`

# How to get started

- Documentation:
  - ▶ Shine Offline TWiki
  - ▶ doxygen
- example programs\*
  - ▶ DS2SHOE
  - ▶ compareShineFiles
  - ▶ nanoSHOE
  - ▶ selectEvents
  - ▶ lambdaAnalysis
  - ▶ recEventLoop
  - ▶ simpleSelection
  - ▶ simulationAnalysis

## Public Member Functions

<code>ClusterListIterator</code>	<code>ClustersBegin () const</code> cluster indices
<code>ClusterListIterator</code>	<code>ClustersEnd () const</code>
<code>void</code>	<code>Detach (RecEvent &amp;recEvent)</code> detach all indices
<code>int</code>	<code>GetCharge () const</code> particle charge
<code>const utl::CovarianceMatrix &amp;</code> <code>const utl::Point &amp;</code>	<code>GetCovarianceMatrix () const</code> <code>GetFirstPointOnTrack () const</code> first measured point on track
<code>double</code>	<code>GetFitProbability () const</code> P( $\chi^2$ , Ndf) of momentum fit.
<code>Index&lt; Track &gt;</code>	<code>GetIndex () const</code> get object index
<code>const utl::Point &amp;</code>	<code>GetLastPointOnTrack () const</code> last measured point on track
<code>const utl::Vector &amp;</code>	<code>GetMomentum () const</code> fitted momentum at point "MomentumPoint"
<code>const utl::Point &amp;</code>	<code>GetMomentumPoint () const</code> point at which momentum was fitted
<code>unsigned int</code>	<code>GetNumberOfClusters (const ETPC id) const</code> number of clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfClusters () const</code>
<code>unsigned int</code>	<code>GetNumberOfdEdxClusters (const ETPC id) const</code> number of dEdx clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfFitClusters (const ETPC id) const</code> number of fitted clusters in TPC id
<code>unsigned int</code>	<code>GetNumberOfSimVertexTracks () const</code>
<code>unsigned int</code>	<code>GetNumberOfVertexTracks () const</code>
<code>unsigned int</code>	<code>GetPotentialNumberOfClusters (const ETPC id) const</code> potential number of clusters in TPC id
<code>int</code>	<code>GetStatus () const</code> get status (e.g. NA49 iFlag)
<code>double</code>	<code>GetTruncatedMeanCharge (const ETPC id) const</code> truncated mean of cluster charges along track
<code>double</code>	<code>GetTruncatedMeanChargeVariance (const ETPC id) const</code> variance of truncated mean of cluster charges

\* located in `install/share/shine-offline/apps/Examples/SimpleAnalysis`