

Picking with CED and Marlin

Hauke Hoelbe, supervisor Frank Gaede

May 2010

1. Introduction

Picking means that it is possible to get information on a particle by clicking on a hit or track in CED. The information will be shown in the Marlin window. Based on the work of Stewart Martin-Haugh a general technique has been designed to enable picking.

2. How to use it

First the following code has to be inserted into the processEvent function of the viewer (for example CEDViewer):

```
CEDPickingHandler &pHandler=CEDPickingHandler::getInstance();  
pHandler.update(evt);
```

The purpose of this code is that the singleton class CEDPickingHandler knows all objects (particles,etc.) of the event and therefore assigns a print function for every object. There are predefined default output functions which can be overwritten by the user. To use your own output functions it is necessary to register them before calling the update function. That can be done by calling the registerFunction function. The first argument is a collection name or type and the second argument is a pointer to the print function.

Example:

```
CEDPickingHandler &pHandler=CEDPickingHandler::getInstance();  
pHandler.registerFunction(LCIO::MCPARTICLE, &MyMCParticlePrintFunction);  
//...more printfunctions for other types  
pHandler.update(evt);
```

Remark: For a proper use of picking it is essential that the IDs of every object drawn in CED are communicated to CED. This is done by using ced_line_ID instead of ced_line and ced_hit_ID instead of ced_hit.

LCIO Output Functions

During the project, print functions have been designed for a number of LCIO objects. The work of Jan Engels served as a basic principle. This allows to send LCIO objects directly to output streams. It is possible to print LCIO objects in a short or long style.

The long form:

```
Vertex *vertex = (Vertex *) YourLCIOVertexObjekt;  
cout << vertex;
```

The short form:

```
Vertex *vertex = (Vertex *) YourLCIOVertexObjekt;  
cout << lc_short(vertex);
```

The short form allows to print objects as tables:

```
Vertex *vertex1 = (Vertex *) YourLCIOVertexObjekt1;  
Vertex *vertex2 = (Vertex *) YourLCIOVertexObjekt2;  
cout << header(vertex) << tail(vertex) << lshort(vertex1) << lshort(vertex2)  
    << tail(vertex);
```

or:

```
cout << header(EVENT::Vertex) << tail(EVENT::Vertex) << lshort(vertex1)  
    << lshort(vertex2) << tail(EVENT::Vertex);
```

These ways of printing are available for the following LCIO types: MCParticle, TrackerHit, SimTrackerHit, CalorimeterHit, SimCalorimeterHit, ReconstructedParticle, Track and Cluster.

3. Changes in CED

- Some small changes were made in CED:
- The “100.000-event bug” in CED was fixed.
- It is possible to select objects by mouse click.
- Zooming by mouse wheel is available.
- CED supports different background colors.

4. Details

MarlinUtil::MarlinCED

- The new class CEDPickingHandler has been added. This class manages a map, that contains the following information: LCIO Objekt id, Objekt pointer, Print Funktion pointer. This information is added by update(Event). Registered objects can be printed by printID(ID).

LCIO

- The data file Operators.cc contains the new operators that allow the print of the LCIO objects.

CED

- The function ced_selected_id_noblock has been added to ced.c. This function is necessary for picking.
- In glced.c there has been added the background color feature.
- The "100.000 Event Bug" in glut_socks is fixed.

CEDViewer

- The CEDViewer has been made ready for picking as described above.