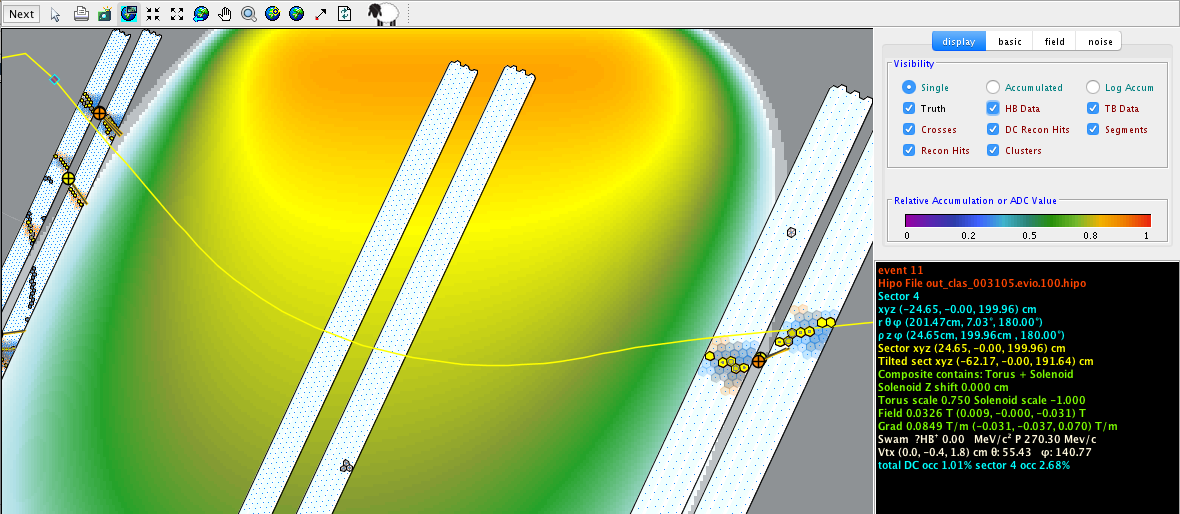
Using *ced*, the cLAS eVENT dISPLAY

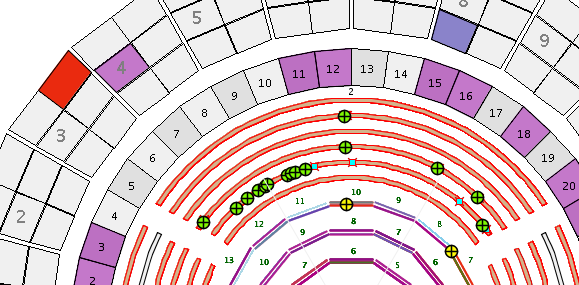
Based on *ced* version 1.04

May 5, 2018

David Heddle

Christopher Newport University





Introduction 2

Requirements 2

Obtaining *ced* 2

Quick Start Guide 3

Anatomy of a View 3

Basic Features 4

Advanced Features 4

Meaningless Features 4

# 

# Introduction

*ced* is a diagnostic graphical application for displaying CLAS events.

The primary element of *ced* is the *view*. A view is a graphical representation of CLAS in its entirety or a subset of detector packages.

*ced* contains multiple *views* of CLAS, some geometrically faithful and some not. It has both 2D and 3D views. Some views are purely informational, typically in the form of tables.

## Requirements

*ced* is 99.9% JAVA. The 0.1% non-JAVA pieces are platform specific libraries (hidden inside of jar files) for the 3D connections to OPENGL. Consequently, the one feature of *ced* that sometimes doesn’t work (especially on some linux platforms) is the 3D. *ced* is still quite useful even when the 3D doesn’t work. In fact, to first order 3D is provided so that those who say any event display worthy of the title “must” have 3D can try it and then realize that the 2D views have denser and more helpful information content. Nevertheless, there is a side project to redo the *ced* 3D using JavaFx, which will eliminate the platform specific 3D libraries.

The other requirement is that your laptop or desktop has JAVA 1.8 or above installed.

## Obtaining *ced*

You can download *ced* from this location: <https://userweb.jlab.org/~heddle/ced/builds/>

After uncompressing the tar.gz file, you should have a directory named *cedbuild*:

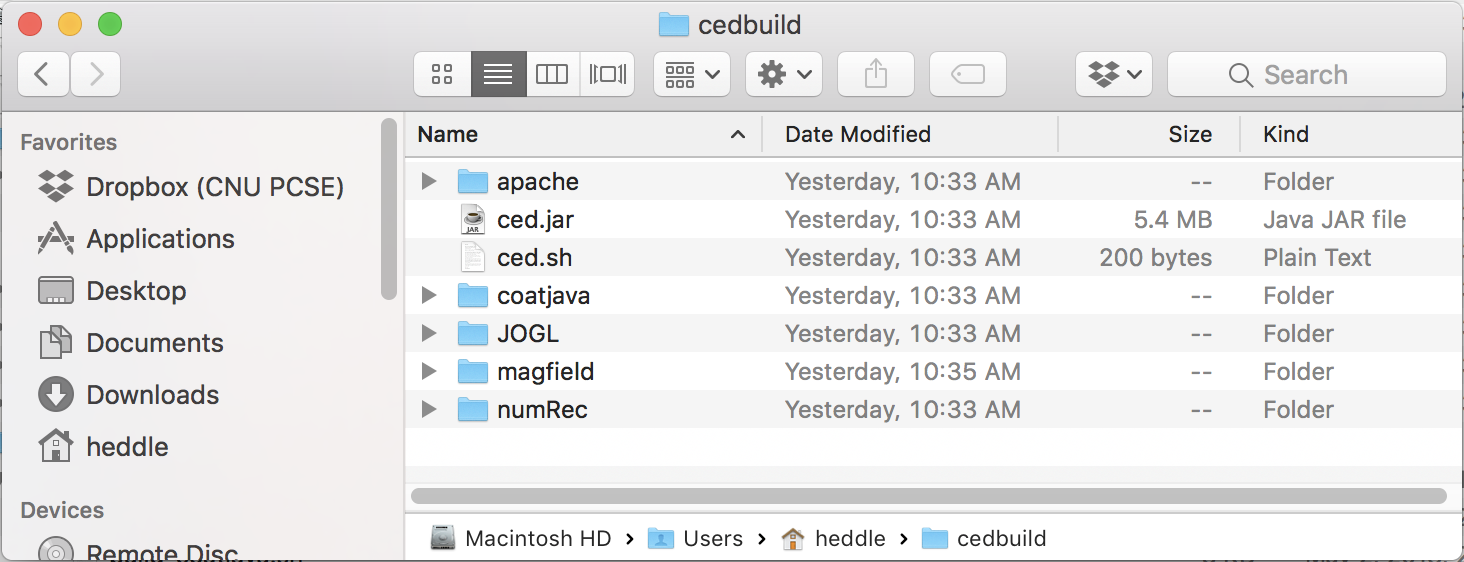


Figure The ced build directory. Double-clicking on the ced.jar file.

Usually *ced* can be launched by double clicking on the file *ced.jar*. If that doesn’t work then from a terminal try running the *ced.sh* launch script i.e., enter source ced.sh. A final alternative, also from the command line, is to enter java –jar ced.jar. The latter two methods are especially useful if *ced* doesn’t launch properly, as they will contain console messages that will likely give clues as to the problem.

# Quick Start Guide

Assuming *ced* launches properly you will see a series of annoying images on a splash screen as *ced* is downloading from the CLAS geometry database. If all goes well you should eventually get to the starting desktop as show in Figure 2:

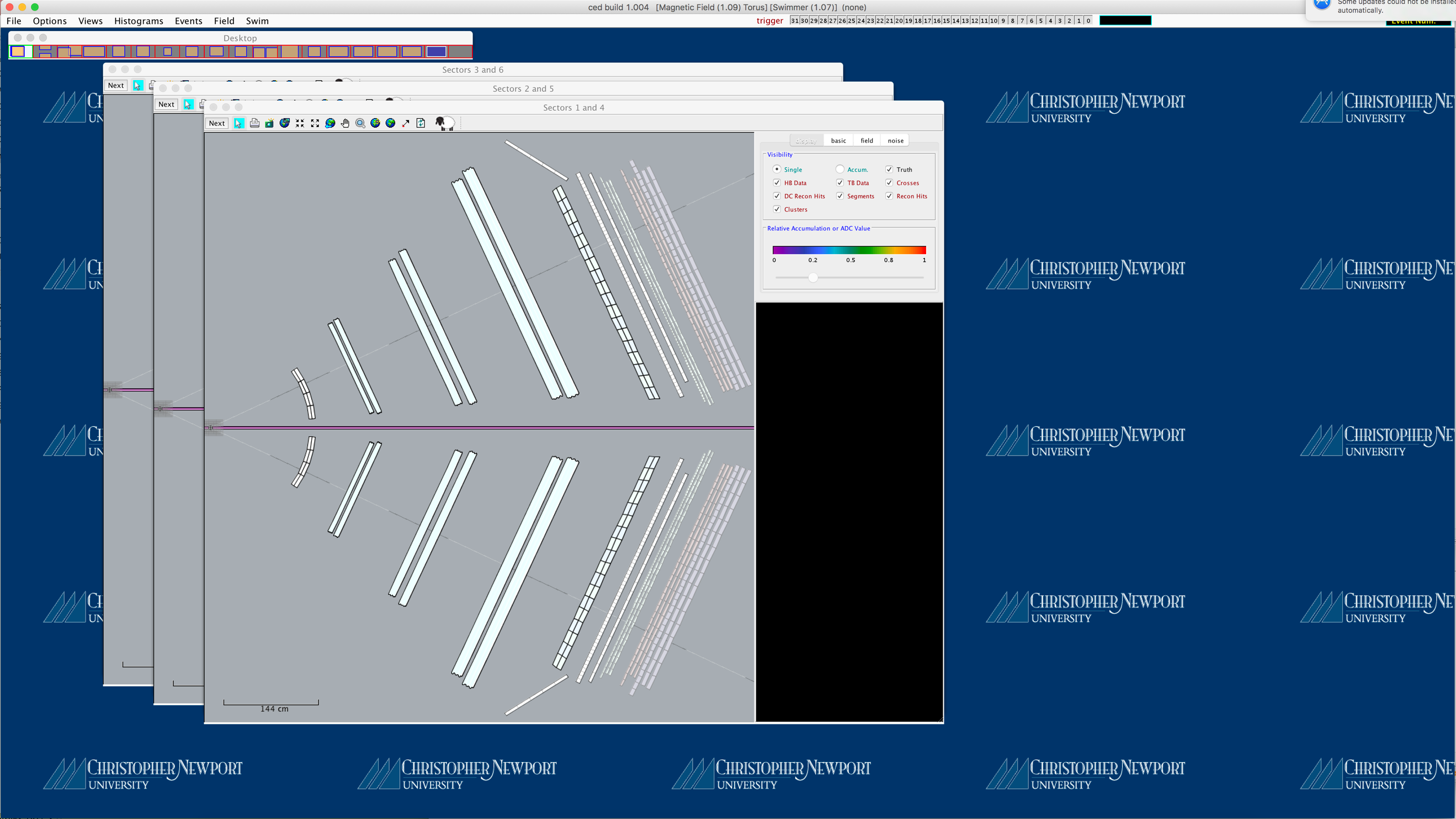


Figure The ced desktop at startup.

The content part of the window shows the three canonical *ced* views: Three 2D slices through forward detectors in opposing CLAS sectors. Views are windows that while confined to the desktop can be independently dragged, resized, and zoomed.

On the upper left of the content area is a special view, the Virtual Desktop View. It should look roughly like Figure 3:

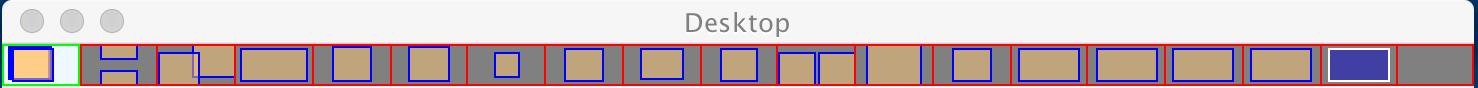


Figure The ced Virtual Desktop View.

As of *ced* v1.004, there are 19 rectangles (with red borders), each representing another desktop with different views. The off-screen views have a gray background, while the active desktop (in the 0th desktop) has a white background. Inside each of the 10 desktops are one or more blue-bordered rectangles, each representing the views on that desktop. Double-clicking on a gray virtual desktop you activate it, and you’ll then see the views it contains. That is not the only way to get to a view—the more common method is to select it from the View menu, to be discussed later. As you mouse-over the different virtual desktops, the title of the Virtual View will change to provide some context information.

Hint: A little known feature is that the blue rectangles, which represent views, can be dragged from one virtual view to another.

## Anatomy of a View

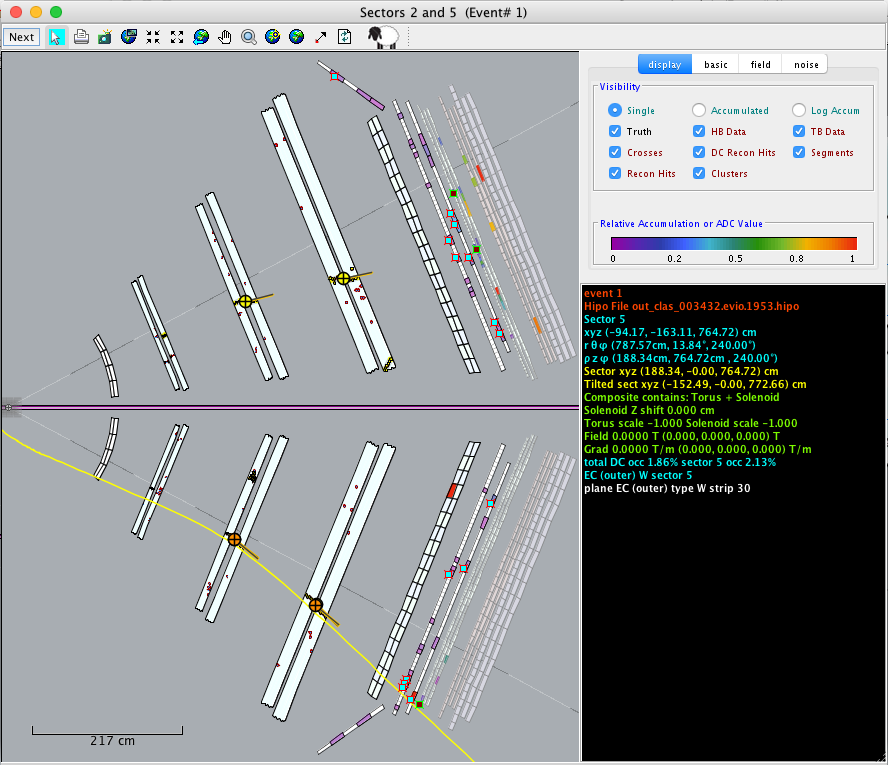


Figure 4 This is a typical ced view. It is a 2D slice through CLAS.

Figure 1 shows a typical *ced* view. It is a geometrically faithful 2D slice through sectors 2 and 5. Along the top is a toolbar with standard manipulation buttons that we assume are familiar and self-explanatory. They may include buttons to select, print, rectangle-zoom, fixed zoom, pan, magnify, restore, place an anchor, refresh and (the Dolly the Sheep button) clone the view. Not all views contain all buttons—some buttons are not appropriate on some views. We will not explain the buttons in detail—we believe they are self-explanatory.

The right-hand side of the view is the control panel. It generally consists of two parts. A display panel on top and a black (with colored text) mouse-over feedback panel that is used for context information.

The display panel may have multiple tabs (as shown in Figure 4). In any case the primary component will be a check box that controls what is being displayed. Again, we are going to punt and declare this self-explanatory.

The black feedback area is dense with information. It updates as you move the mouse over the content area of the view. It gives (depending on the view) locations, magnetic field information, ADCs and TDCs of hits. It updates instantly—it does not require hovering.

Hint: A little known feature is that holding down the control button will stop the updating. This is useful if you want to get a screen shot with information of interest displayed.

View Zoology

# Basic Features

# Advanced Features

# Meaningless Features