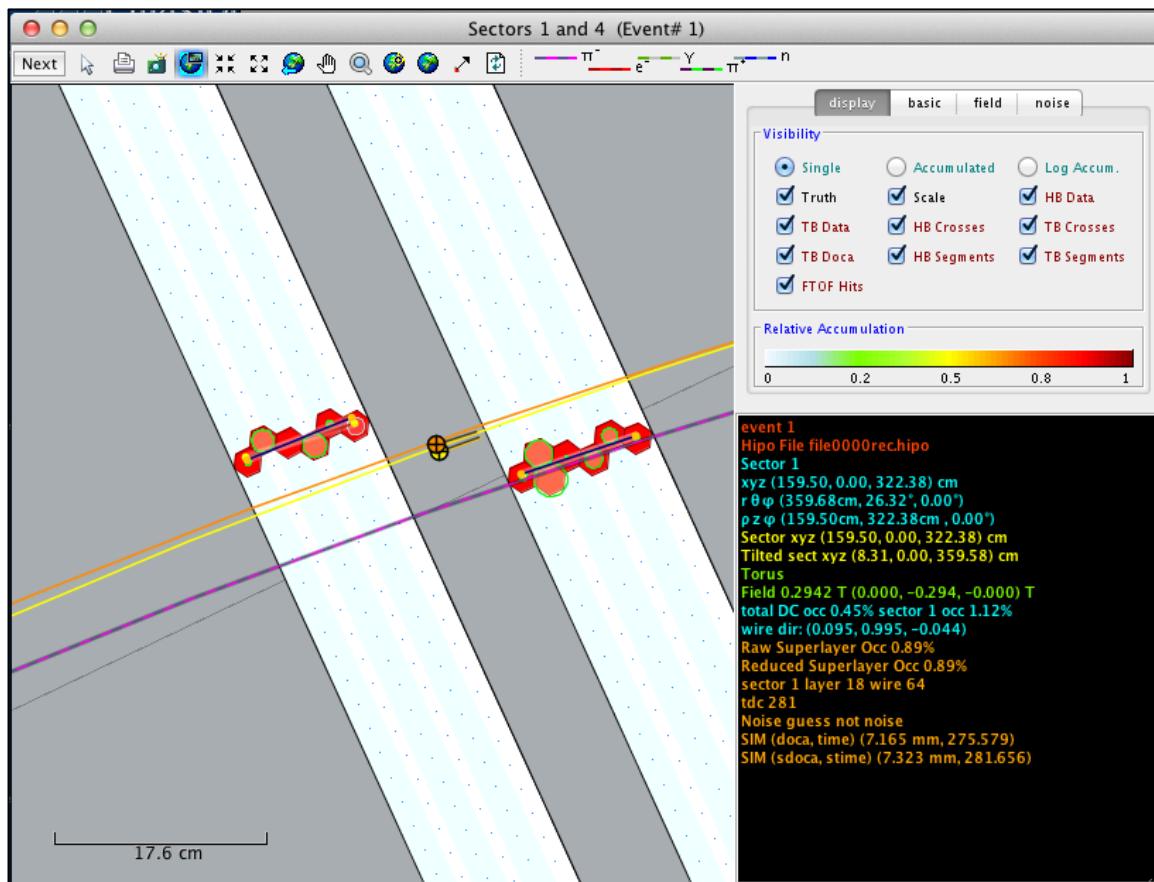


ced

Counting House Manual

Document version 0.1

All “real” detectors are made of pixels



Introduction

This manual describes the use of *ced* (“see-e-dee”), which stands for CLAS Event Display¹. *ced* provides a variety of “views” of CLAS detectors and of CLAS data. Most are 2D visualizations, some are 3D, and some are not visualizations at all but rather histograms or tables. *ced* can show a single event or can accumulate events for visual histograms e.g., drift chamber occupancy. *ced* can read archived events from Hipo formatted files, or stream live events from either Hipo or ET rings.

Starting *ced*

In the counting house, login as *clasrun* using the top-secret immutable password.² It is claimed that someone will have done all the magic necessary that all you have to do from a counting house workstation is enter the command *ced*. Seriously, I’ve been told: *don’t put in this manual where ced is or mention launch scripts, environment variables, etc. Just tell them to enter “ced”*. So there you have it.

If all goes well, you’ll see a splash screen something like this:³

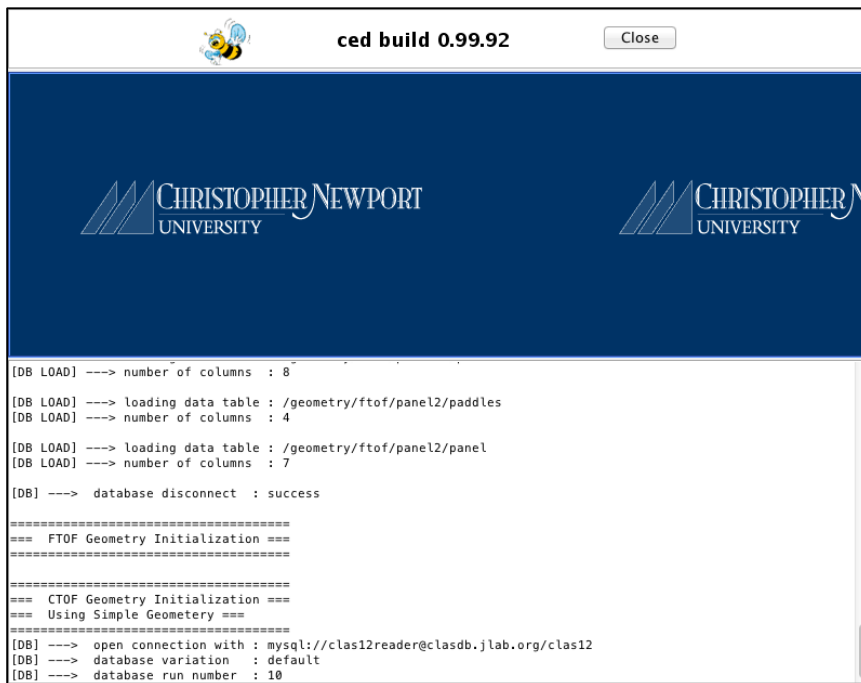


Figure 1. The *ced* splash screen.

¹ *ced* is for CLAS Event Display. *CLAS* is for CEBAF Large Acceptance Spectrometer. *CEBAF* is for Continuous Electron Beam Accelerator Facility. So *ced* actually stands for “Continuous Electron Beam Accelerator Facility Large Acceptance Spectrometer Event Display”, providing a respectable compression of 30:1.

² If the immutable password ever, um, mutates, nothing will ever work. Ever. Move to Hall D.

³ It may not have an animated bumblebee. But it’ll have *something*.

This screen (Figure 1) indicates the *ced* is initializing. The version number may be of interest, but the text scrolling by probably is not. It is mostly giving information about the geometry that *ced* is trying to obtain from the common-tools geometry service.

If *ced* initializes properly, then after about 15 seconds the splash screen should go away and you should see the *ced* initial configuration (Figure 2):

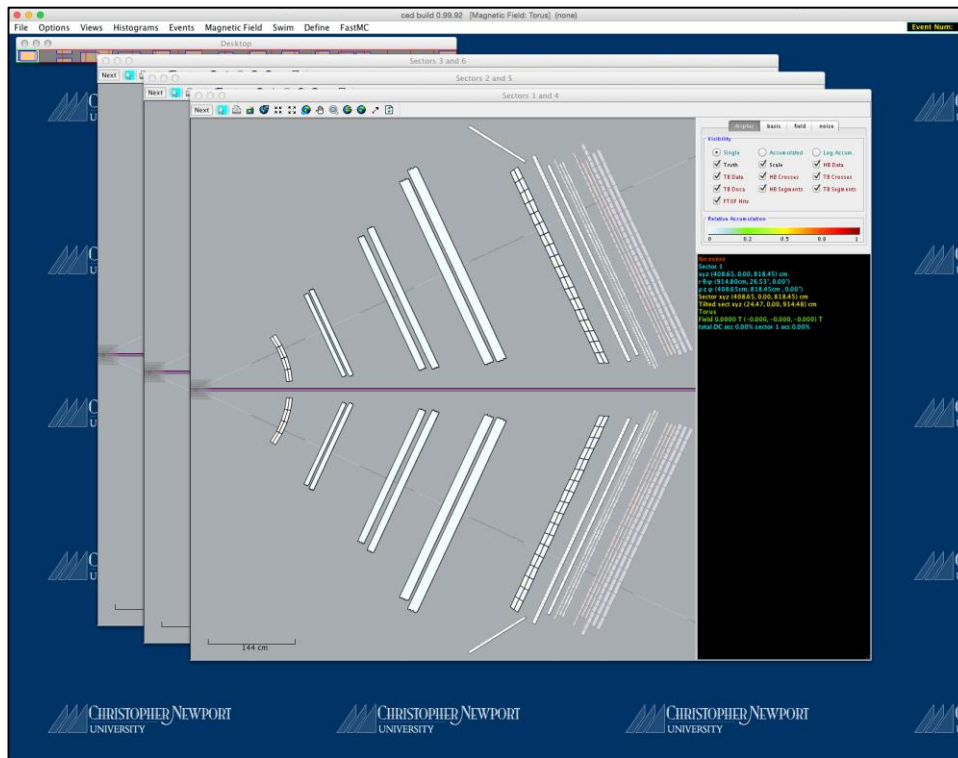


Figure 2. The startup configuration.

ced is what is known as a *multiple document interface* (mdi) application. This means it has one main window, emblazoned with Christopher Newport University propaganda⁴, and independent smaller windows (called views) that you can resize and drag about. This is in contrast to each view being an independent window.⁵ This (mdi) is the interface used by AOL in 1985, so you now it's a good paradigm.

⁴ Yeah, well, other university groups put their logos on stickers which they paste on detectors. We get to be obnoxious too.

⁵ This was a design decision. We encourage you to report bugs (to heddle@jlab.org). We encourage you to submit suggestions, feature requests, and corrections to this document to the same address. But as for critiques of the basic design, you should enter the command `$cat > /dev/null`, then enter your criticisms, and terminate with `^D`. We'll get to 'em as soon as we read 'em.

Views	Histograms	Events	N
Desktop			
Current Event			
Sectors 3 and 6			
Sectors 2 and 5			
Sectors 1 and 4			
Monte Carlo Events			
Reconstructed Tracks			
All Drift Chambers			
Central Z			
Central XY			
FTCal XY			
Drift Chambers XY			
Projected Drift Chambers			
EC			
PCAL			
Forward Detectors 3D View			
Central Detectors 3D View			
FTCal 3D View			
Log			
sPlot			

Views

The views you see at startup are three 2D slices through the detectors, through the midplanes of sectors 1 and 4, sectors 2 and 5, and sectors 3 and 6. These are but three of (at the time of this writing) 19 views (24 if you count histogram grids). Where are all these views? We will talk about menus later, but we'll point out here that all the views are reachable through the **Views** menu. Selecting the view "take you" to the view, it won't plop the view in front of what you are presently viewing. That is because *ced* lays out its views on a virtual desktop,

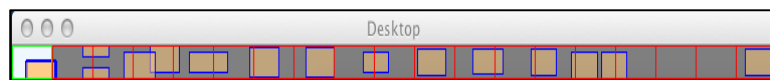


Figure 3. The Views menu and the Virtual Desktop view.

which itself is a sort of view. Each red grid rectangle on the virtual desktop is a "screen", and each screen has one or more views, represented by smaller, tan rectangles. You can double click to jump to another screen. You can even drag the views from screen to screen using the virtual desktop. Why, it's virtually a desktop.

Toolbar

Most of the views have a toolbar across the top.

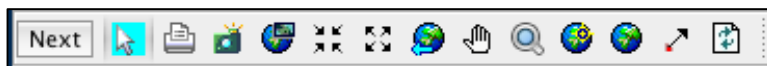


Figure 4. A view toolbar for zooming, panning, etc

These buttons can be used to print, zoom, magnify, pan, re-center, etc. It is easier just to tell you to experiment with them than to explain them. They do have tool-tips which may or may not be helpful. The "next" button is just a convenient way to go to the next event from any view.

The Menus

We will go through them menus, explaining those items relevant for on-line use carefully and glossing over those that are frivolous or intended for off-line use.

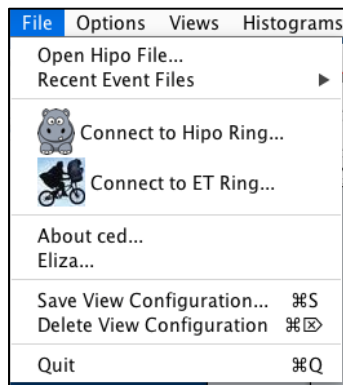


Figure 5. The File Menu.

File Menu

The file menu is perhaps the most important of all the *ced* menus. This is where you specify the source of the displayed events. This is accomplished with the first four items in the menu.

Open Hipo File...

Use this item to open a Hipo file. You will be presented with a familiar file selection dialog filtering on files with a *.hipo* extension. If the file opens, you will be able to select “next” for next event from a number of locations including the toolbar on most views and the Events menu (discussed below).

Recent Event Files

This will give you a short-cut selection of the most recently opened files.

Connect to Hipo Ring

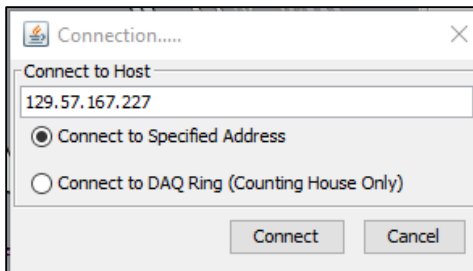


Figure 6. Connection to a Hipo ring.

From here you will get a dialog asking you to specify the IP address of the Hipo ring. Another option is to select “*Connect to DAQ Ring (Counting House Only)*”. This is supposed to “discover” a Hipo ring on a clondaq machine. At the time of this writing it hasn’t been tested. Surely it will work.

If you successfully connect to the ring, then all other actions will be the same as if you opened a file (except you won’t be able to select “previous event” or “go to event.” Those are for file sources only.

Connect to ET Ring

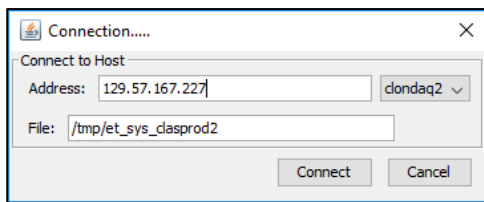


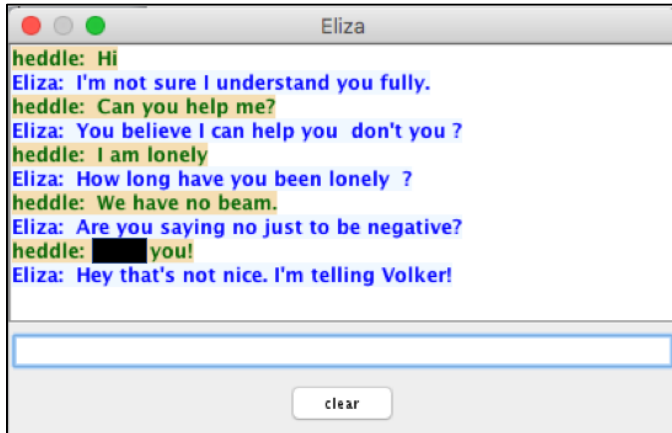
Figure 7. Connect to an ET ring.

This is similar to connecting to a Hipo ring, except you connect to a “rawer” (more raw?) ET ring where you might find detector hits (you should!) but you will *not* find reconstructed banks. You specify an IP address and a session (Called here, *File*. That might change.) The correct values, if different from the defaults, will hopefully be posted.⁶

⁶ As with CLAS 6, important notifications will probably be “posted” in some barely legible handwriting on a yellow post-it note (hence *posted*) of questionable stickiness, so look on the floor.

About ced
Uninteresting propaganda.

Eliza...



Eliza is a psychoanalyst.

You will be tempted to swear at her. Do not swear at her. It's not Eliza's fault everyone is home sleeping while you are running a shift at 3:00 am, and that if you had studied a little harder, you could have been a *real* doctor.

Figure 8. A free Eliza session.

Save View Configuration/Delete View Configuration

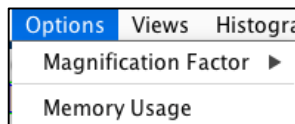
These control saving (and deleting) a configuration. This is a way to save the state of *ced* with the views just the way you like them. It hasn't been tested since 2012. It might still work.

Quit

You can experiment with that to see how it works.

Options Menu

The Options Menu is least important menu for *ced*. I forgot it was even there until I started writing this document. It allows you to control the magnification factor used by the magnify tool on a view's toolbar. (It's the button that looks like a magnifying glass.)



Views Menu

This is the menu we saw above (Figure 3). It allows you to jump to any of the *ced* views. I don't have the endurance to go through each view. I'll just mention one, the most important view that is not a detector view.

Current Event View

This (Figure 9) is a tabular view of the contents of the current event. It is extremely useful for debugging. All the known banks in the CLAS bankosphere are shown on the right. The banks in red are present in the current event, the banks in gray are not. Each row in the table corresponds to a bank column. If you click on the row, the column data will appear as a vertical array in the white area on the left of the view.

If you *double* click on a red bank name, a table (Figure 10) showing *all* columns is displayed. You can turn off columns you don't want to see using the checkboxes at the bottom. A technical tidbit: the main view (titled "Current Event") is really a *ced* view. It lives somewhere on the virtual desktop and will be hidden if you go to a different virtual screen. The secondary "view" (in this example, titled "ECAL::tdc") is *not* a view at all. It is a *dialog*. It will be visible even if you change virtual screens. This is a feature, not a bug. You can bring up a bank and then go to the relevant detector view and compare what is in the bank to what is being displayed. If they don't agree you have found a bug and you can send me a nasty email.

The screenshot shows the 'Current Event' window with the following components:

- Source:** /Users/heddle/git/cnuphys/ced/data/HippoData/test.hipo
- Buttons:** prev, next, Go to #, event # 1, count 2000, run # 11
- Table:**

Name	Type	Count
DC::doca.LR	byte	238
DC::doca.doca	float	238
DC::doca.sdoca	float	238
DC::doca.stime	float	238
DC::doca.time	float	238
DC::tdc.TDC	int	238
DC::tdc.component	short	238
DC::tdc.layer	byte	238
DC::tdc.order	byte	238
DC::tdc.sector	byte	238
ECAL::adc.ADC	int	49
ECAL::adc.component	short	49
ECAL::adc.layer	byte	49
ECAL::adc.order	byte	49
ECAL::adc.ped	short	49
ECAL::adc.sector	byte	49
ECAL::adc.time	float	49
ECAL::tdc.TDC	int	49
ECAL::tdc.component	short	49
ECAL::tdc.layer	byte	49
ECAL::tdc.order	byte	49
ECAL::tdc.sector	byte	49
FTOF::adc.ADC	int	102
FTOF::adc.component	short	102
FTOF::adc.layer	byte	102
FTOF::adc.order	byte	102
FTOF::adc.ped	short	102
FTOF::adc.sector	byte	102
FTOF::adc.time	float	102
FTOF::tdc.TDC	int	102
FTOF::tdc.component	short	102
FTOF::tdc.layer	byte	102
FTOF::tdc.order	byte	102
FTOF::tdc.sector	byte	102
MC::Particle.pid	int	20
MC::Particle.px	float	20
- Vertical Array (Left):** Shows data for the selected bank (ECAL::tdc.layer): [204] 10, [205] 19, [206] 20, [207] 21, [208] 22, [209] 23, [210] 24, [211] 25, [212] 26, [213] 27, [214] 28, [215] 29, [216] 30, [217] 31, [218] 31, [219] 32, [220] 33, [221] 34, [222] 35, [223] 36, [224] 1, [225] 1, [226] 2, [227] 3, [228] 4, [229] 5, [230] 6, [231] 7, [232] 8, [233] 9, [234] 10, [235] 11, [236] 12, [237] 33, [238] 26.
- Bank List (Right):**
 - CTOF::adc
 - CTOF::hits
 - CTOF::adc
 - DETECTOR::hits
 - ECAL::clusters
 - ECAL::peaks
 - FTCAL::adc
 - FTOF::clusters
 - FTOF::hits
 - FTOF::rawhits
 - HTCC::adc
 - HTCC::tdc
 - HitBasedTrkg::HBCrosses
 - HitBasedTrkg::HBSegmentTrajectory
 - HitBasedTrkg::HBTracks
 - LTC::adc
 - MC::Particle
 - RAW::tdc
 - REC::Particle
 - REC::Particle
 - RUN::config
 - SVT::adc
 - TimeBasedTrkg::TBClusters
 - TimeBasedTrkg::TBCrosses
 - TimeBasedTrkg::TBSegments
 - CTOF::dgz
 - CTOF::rawhits
 - DC::dgz
 - DC::tdc
 - ECAL::adc
 - ECAL::dgz
 - ECAL::tdc
 - FTOF::adc
 - FTOF::dgz
 - FTOF::matchedclusters
 - FTOF::tdc
 - HTCC::rec
 - HitBasedTrkg::HBClusters
 - HitBasedTrkg::HBHits
 - HitBasedTrkg::HBSegments
 - RAW::adc
 - REC::Detector
 - REC::Detector
 - RUN::config
 - SVT::adc
 - TimeBasedTrkg::TBCovMat
 - TimeBasedTrkg::TBSegments
 - TimeBasedTrkg::TBSegments
 - TimeBasedTrkg::TBSegments

Figure 9. The Current Event view.

	sector	layer	component	order	TDC
1	4	3	37	1	0
2	3	2	43	1	41432
3	3	1	29	1	26778
4	3	3	46	1	42678
5	3	3	54	1	0
6	4	3	29	1	162581
7	4	1	39	1	158827
8	4	2	59	1	162579
9	4	2	58	1	162689
10	4	3	30	1	162975
11	4	2	60	1	163245
12	4	3	31	1	0
13	4	1	37	1	158775
14	4	1	40	1	159213
15	4	1	38	1	0
16	6	1	53	1	152580
17	6	2	26	1	157824
18	6	3	49	1	157871
19	6	2	25	1	158022
20	1	1	44	1	0
21	6	1	21	1	151024
22	1	2	11	1	152397
23	1	1	62	1	152077
24	1	1	45	1	150876
25	1	2	8	1	153265
26	1	1	67	1	163107
27	1	3	19	1	150130
28	1	2	52	1	161931
29	1	2	50	1	161084
30	5	9	30	1	0
31	4	4	15	1	158649
32	4	5	35	1	164945
33	4	4	14	1	158515

Visibility

- ☒ sector
- ☒ layer
- ☒ component
- ☒ order
- ☒ TDC

Figure 10. All the columns for the ECAL::tdc bank.

Histograms Menu

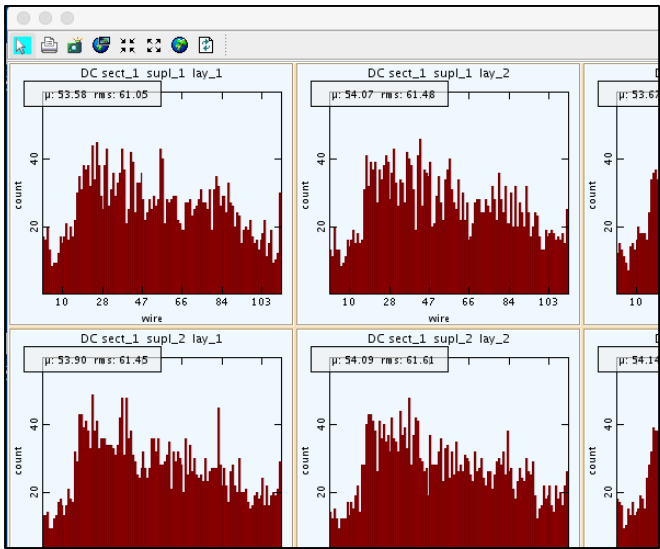
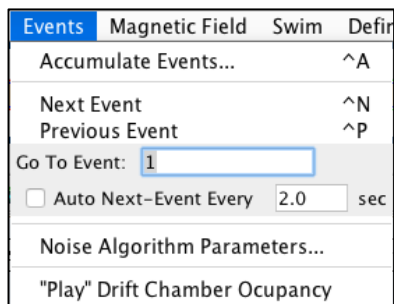


Figure 11. Example of the ced histograms.

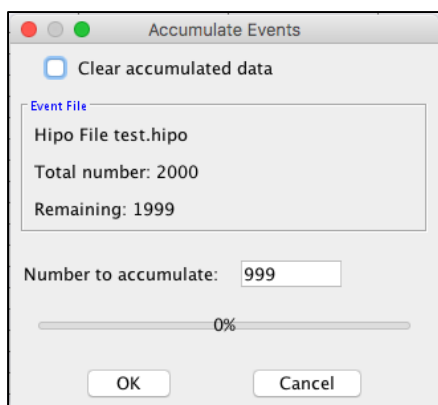
The Histograms menu shows histograms for some of the detector packages, viz. DC, FTOF, BST, PCAL and EC. After an accumulation (described below in the Events menu) these histograms get filled. Each histogram can be zoomed using the rectangle zoom tool. I don't know how useful that is, but it's kind of cool.



Events Menu

This is an important menu. It has self-explanatory items for going to the next event, the previous event, and an arbitrary event. The latter two are not available for streaming sources (the Hipo and ET rings.)

Figure 12. The Events menu.



Accumulate Events...

This allows you, through a dialog, to accumulate events for creating normal and visual histograms (occupancy plots) as well as user-defined plots as described in the section of the *Define* menu below. You simply select the number of events you want to accumulate and hit the OK button. Each detector view has in its control panel (in the upper right) a way to select whether you want the view to display a single event or accumulated data (Figure 14).

Figure 13. The Accumulation dialog.

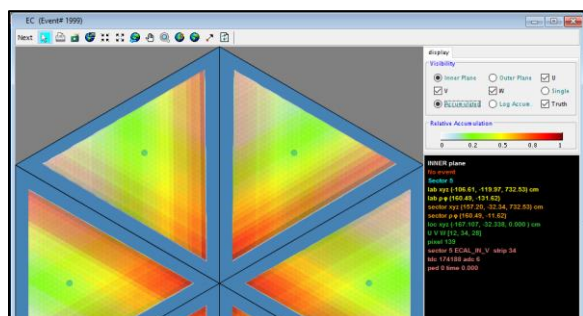


Figure 14. An example of displaying accumulated data.

Noise Algorithm Parameters

This controls (through a dialog) what *ced* thinks is uncorrelated drift chamber noise. It is not important here, so I won't discuss it.⁷

"Play" Drift Chamber Occupancy.

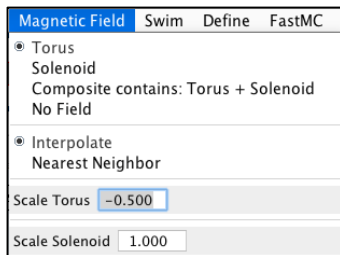
Selecting this will cause *ced* to emit a short tone whose pitch is related to overall drift chamber occupancy. So you can listen to the drift chambers. It seems like a good idea to

me to leave this on, but some who lack vision worry that it might get annoying. So you have to turn it on.⁸ After that it can't be turned off. Just kidding. (Or am I?)

⁷ Just to be clear, *ced* is not affecting what data is recorded by the DAQ system. So if you change the noise algorithm parameters, it won't cause more or less data to be archived. Not sure why *ced* is not trusted enough...

⁸ Being bitter about the youth of grad students, I considered using "under thirty" frequencies, those high frequencies that most people stop hearing after the age of 30. At full volume. So that when grad

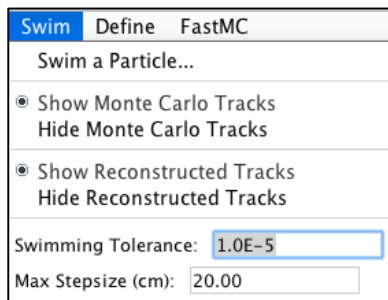
Magnetic Field Menu



The Magnetic Field menu allows you to specify whether the magnetic field is the torus, the solenoid, or both. It also allows you to choose scale factors. In the counting house this is likely not important unless you are showing reconstructed tracks. In that case you want the swimmer to swim with the right field. Note: Hipo data should include a *run* bank which specifies the field

configuration. When *ced* encounters the run bank, it changes its field settings to match. You can also change the algorithm for finding field values from interpolation (tri-linear) to nearest neighbor. But there is no need to. You won't notice the difference.

Swim Menu



The Swim menu allows you to swim a particle through CLAS. It also controls whether tracks from any particles found in MC (Monte Carlo) banks are shown (obviously not relevant for the counting house) and whether reconstructed tracks are shown. It also has controls for the adaptive step-size Runge Kutta swimmer. Change those at your own peril. I'm personally afraid to touch them.

Define Menu

The Define Menu allows you to plot anything in the data stream.⁹ Seriously, just about anything.¹⁰ Any column from a bank against any other column. Whether it is meaningful or not. You define what you want to plot. Then, to fill the plot, you accumulate as described earlier. You can create cuts. You can define expressions.

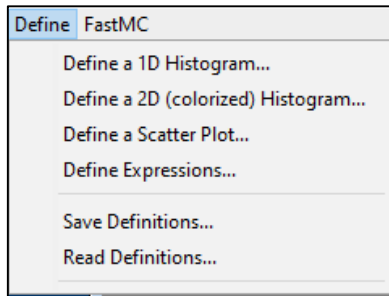
students cover their ears and demand that the incessant pinging stop, I can say: "I don't hear nothing."

⁹ It was broken from some unknown time until *ced* version 0.99.94. I believe about a year ago someone (maybe Mac) told me it wasn't working. I said: "I'll look into it" which is software-speak for "It used to work, I'm sure it's operator error." Kind of a corollary of "yes I tested it" which is software-speak for "it compiles." Anyway this exercise has converted me from a documentation apostate to a documentation zealot. If I hadn't needed pictures for this manual (demanded by my cruel overlords) I would never have tested it. For the next 48 hours (being a zealot has a short shelf life) I'll remain convinced that the best unit-testing ever is writing a manual.

¹⁰ You could, for example, produce a scatter plot of DC wire number vs. PCAL adc values. If you see a correlation, that would be *awesome*.

This is a somewhat complex feature¹¹. Let's just take a simple case, defining a scatter plot of DC tdc value vs. DC smeared doca.

We start by selecting *Define a Scatter Plot...* This brings up the plot definition dialog (Figure 15).



Notice I selected the desired banks and columns. That's all there is to it. You can't choose the name of the plot; the name is derived from the data selected. After I have defined the plot, it shows up on the *Define* menu (Figure 16) so that that you can make it visible. It will be empty until you do an accumulation.

After accumulation I achieved the result shown in Figure 17 . More complicated

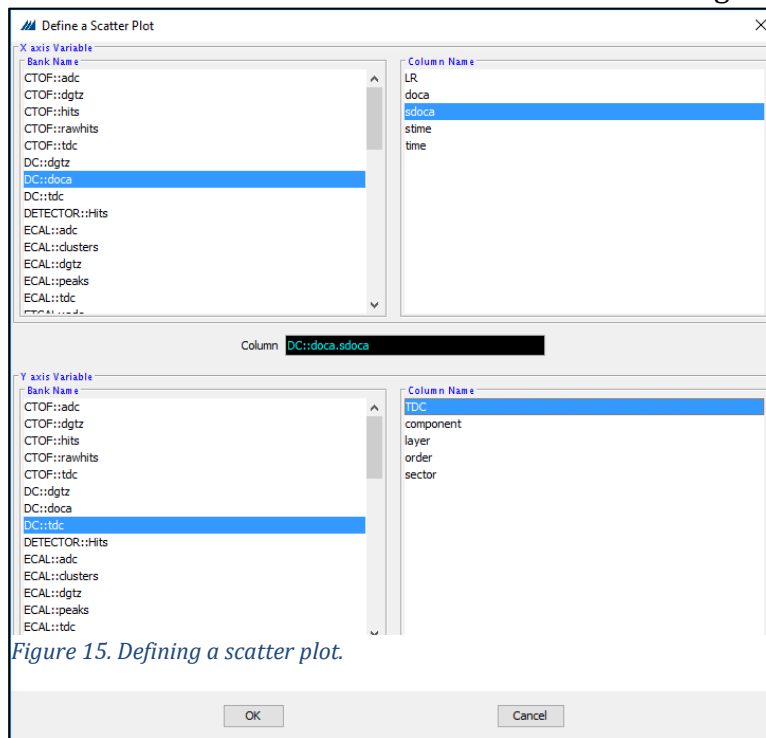


Figure 15. Defining a scatter plot.

features, such as defining expressions and making cuts... I am going to assume that if you need those features you'll let me know and I'll write up some more instructions (as soon as I remember how to use them.)

¹¹ Given that this was broken for a long time and I heard almost nothing (see previous footnote) the software misery index (SMI) of this feature is potentially enormous. SMI is computed by: $SMI = \text{complexity} / (1.0e-16 + \text{usefulness})$ where the $1.0e-16$ is to preclude the very real possibility of dividing by zero.

Fast MC Menu

This menu is not relevant (as far as I can imagine) for counting house usage. It is primarily for reading and swimming tracks from a Lund file. They are run through a fast Monte Carlo (FastMC¹²) which creates purely geometric hits.

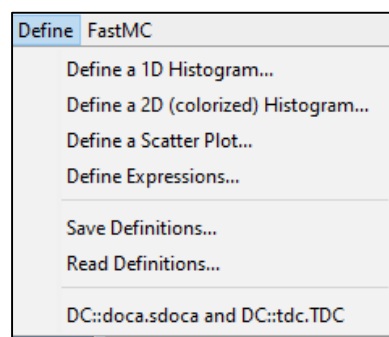


Figure 16. The Define menu after the scatter plot is defined.

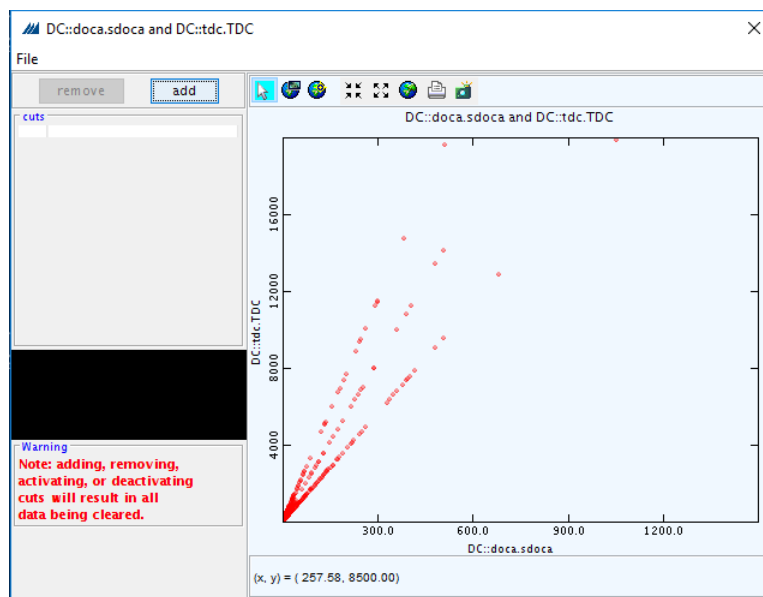


Figure 17. A sample user-defined scatter plot.

¹² Which, being fast but not instantaneous, was actually (no kidding) going to be called *Half-fast MC*. But if you say that (try it in your head) it sends a message that was not intended.