gamecock

Purpose

The intension of gamecock is to walk the user through all SC (time-of-flight) calibration steps using a unified look-and-feel surface. While gamecock has all necessary ***fit*** procedures built in, it uses external programs to ***create*** these histograms. These external programs are described in the SC-calibration documentation (cvs: docs/tof_calib/tof_calib.ps.gz), if you have problems at this step refer to this document.

Getting started with gamecock:

System requirements: gcc 3.x and root 3.05-05 or higher. The whole package has been developed and tested under Redhat 9, which comes with gcc3.2 and glibc2.3.

Checkout the cvs tree:

Type: cvs co packages/utilities/sc_calib/gamecock

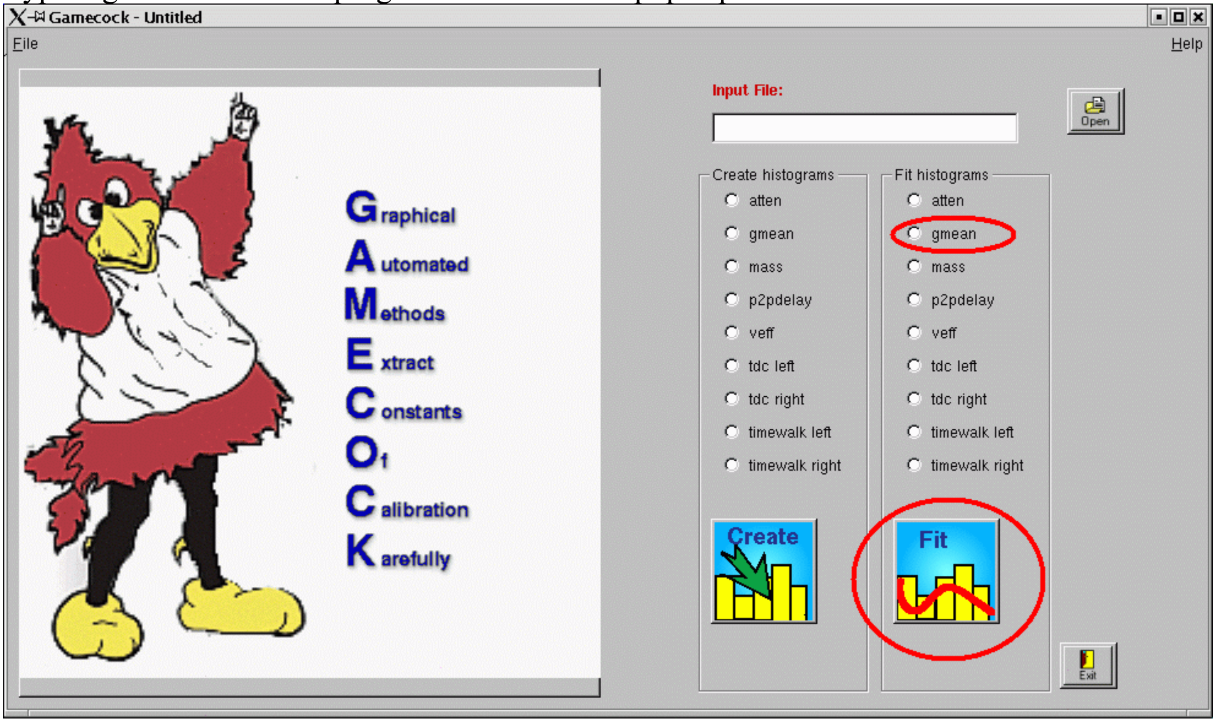
Make the program

Type: Make

Tutorial: Run the demo to get familiar with program features

The sample root histogram for the tutorial is located in the subdirectory root. It is part of the installation, when the tree is checked out. This subdirectory is the location for the default files, which are used when you omit the histogram file name on starting a fit procedure.

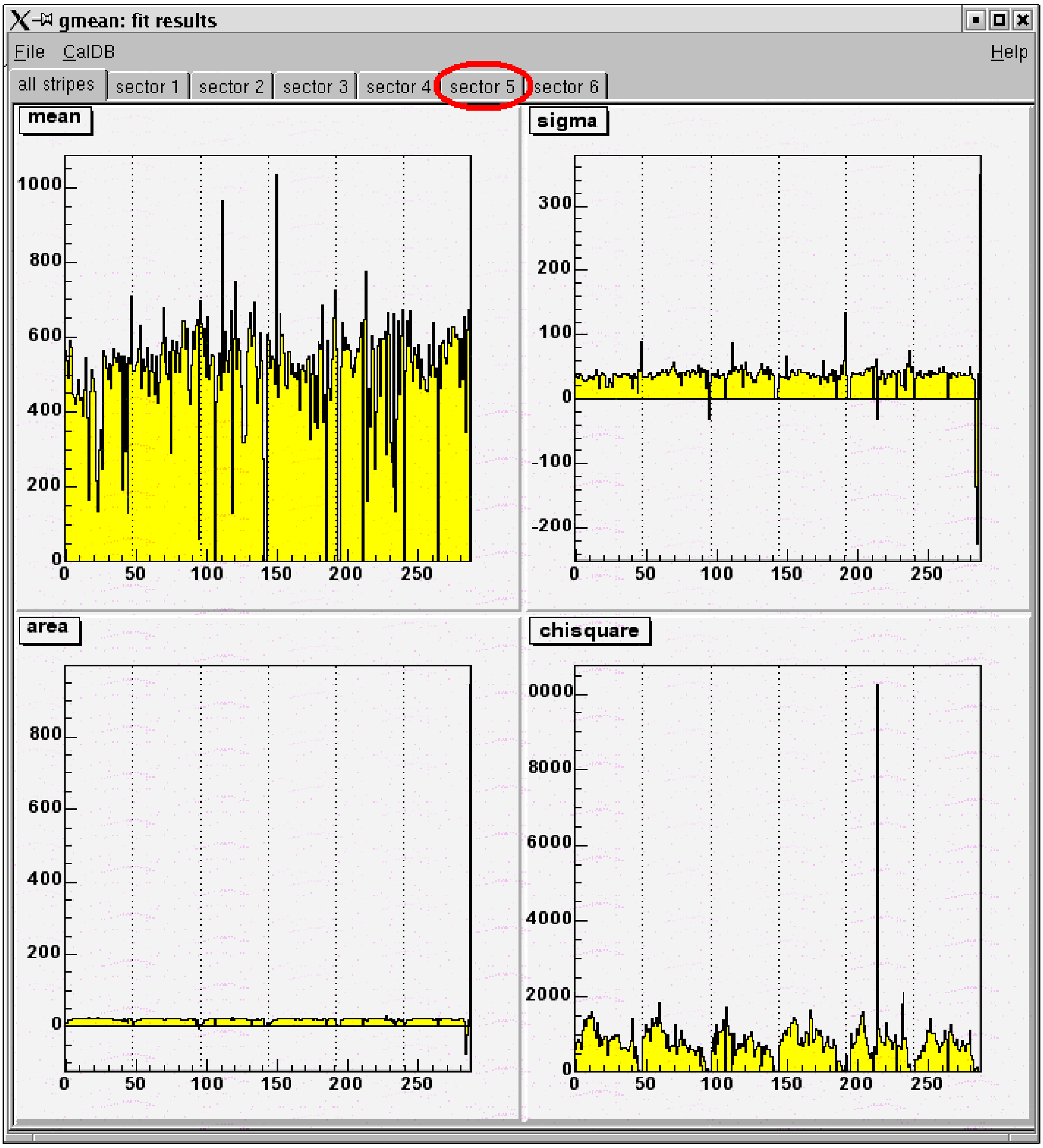
Type: gamecock   the program main window pops up



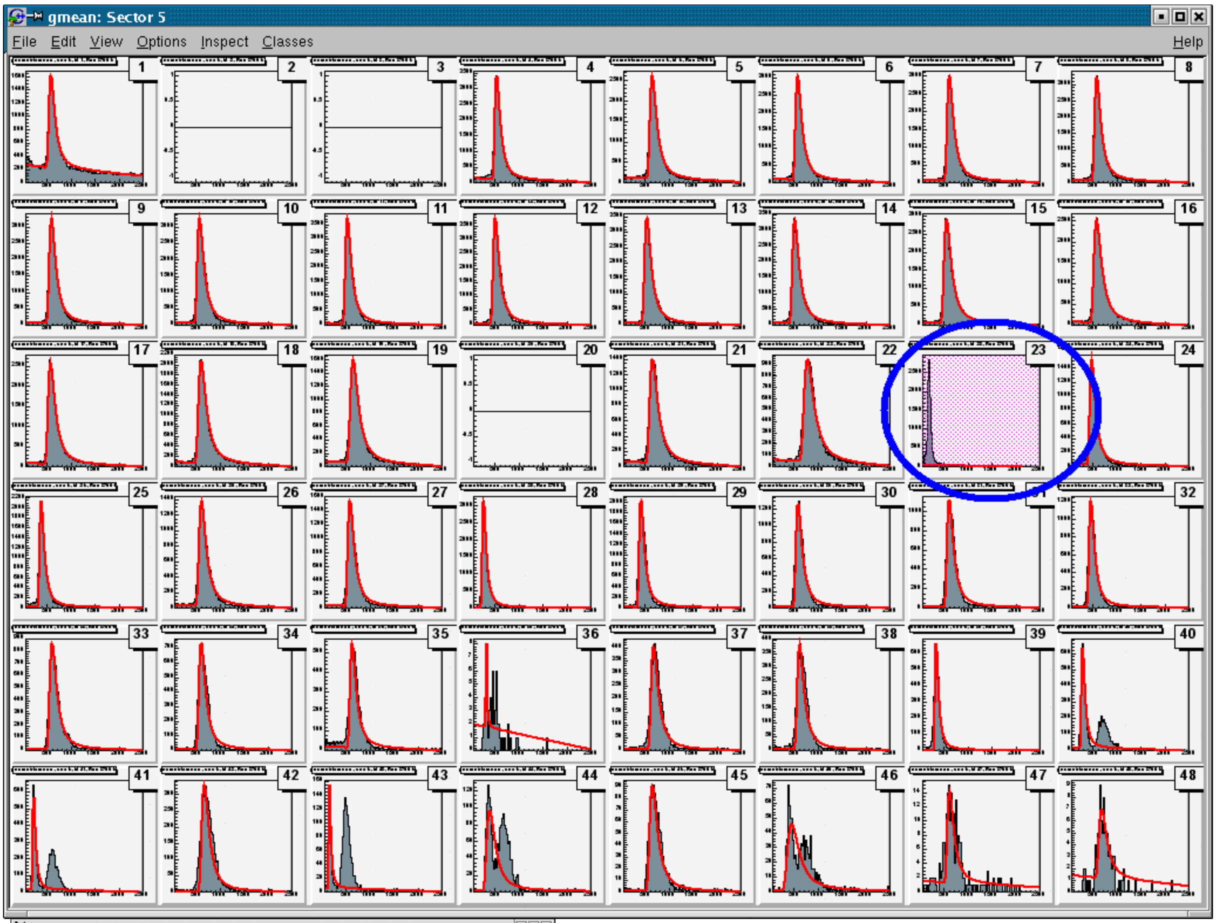
Click on the radio button gmean in the "Fit histograms" column to select the gmean calibration step. In this calibration step the ADC are calibrated using the pion MIP.
Click on the large fit button. The programs loads the default histogram file root/gmean.root and starts the fit procedure for the 288 SC channels. Eventually a fit will fail (e.g. due to the fact, that there is no data for a broken PMT available) and the user gets a warning message stating the sector and the stripe number of this channel. At the end, two new windows will

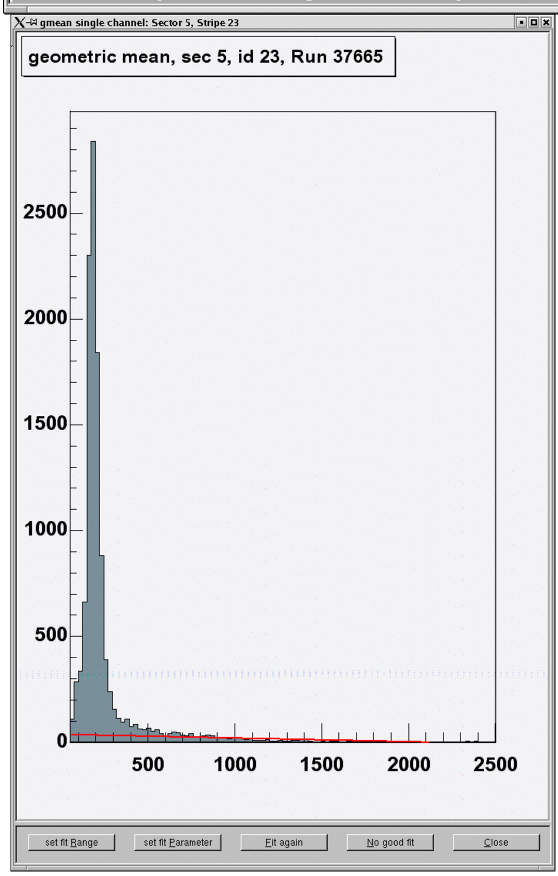
pop up, one shows the fit results (mean, sigma, area under the fit/histogram entries, chisquare), the other one thumbnails for all the histograms and their fits for one sector.

Click on the "sector 5" tags in the fit results window. The fit result window will zoom all histograms so that only the selected sector is shown. The thumbnail "sector" window switches to the selected sector as well.



Lets have a look on the "gmean: Sector 5" window. A lot of thumbnails are showing good fits to the data. In some plots (like stripe 2, 3, 20, 36), there is no or sparse data due to hardware problems in this particular run. Please note that the peculiar double peaks (e.g. Channel 40 and 41) are caused by hardware as well, there are two PMT coupled to one ADC and one TDC. To decouple the PMT can only be done in the underlaying program which creates the histograms.





In SC stripe 23 the data distribution is reasonable, however, the fit routine failed. After clicking on this thumbnail a new window showing this particular channel pops up. If this window was already visible on the screen, it switches to the selected channel. A background pattern in thumbnail indicates the current selection.

Tutorial: Improve the calibration for a channel with reasonable data distribution
If the data distribution looks good and the fit function is not giving satisfactory results, the fit can be improved by either using better start parameters or by limiting the fit function to a range.

Set fit Range: This method is most likely to help if there are widespread background entries in the histogram (especially in two dimensional histograms) which might have a bad influence on the fit. The "Set Range" options gives the user a handle to get rid of these entries, by selecting a particular interval in a one dimensional histogram, or a two dimensional polygon in the two dimensional case. After pushing the button, two left mouse clicks defining the interval are necessary in the histogram shown above. A reasonable choice in our case would be x_0=50 and x_1=350 (y-position of mouse click disregarded in the one dimensional case) marking the interval which comprises the peak. Vertical dashed lines are added to the plot to show the selected range.

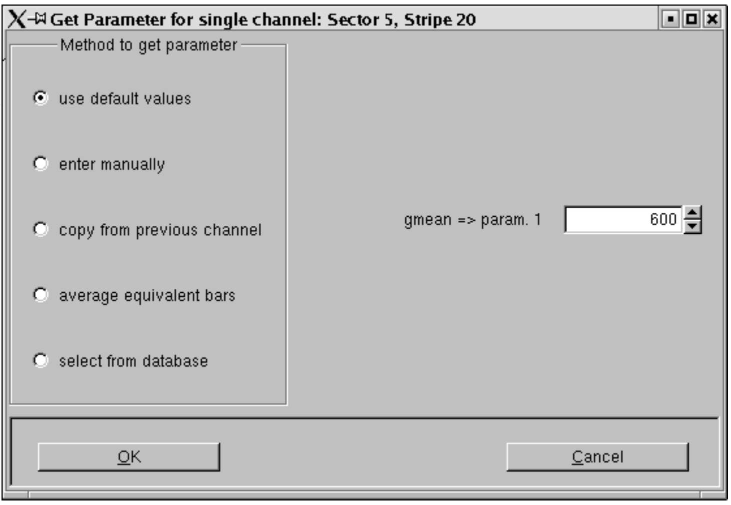
The selection of the range works a little bit different in the two dimensional case (not shown in the tutorial). First it needs n left mouse clicks to define the polygon area with n edges, followed by a single right click to indicate that no more points are going to be added. Due to limitations of the root graphics editor, the borders of the polygon are first shown after the final right click. In any case: you are free to repeat this procedure if you are not satisfied with the selected range.

Set fit Parameter: All fit functions try to minimize the chisquare. However, there might be a local minimum which have to be avoided by using a good set of start parameters. Upon pushing this button, a slider box to set the parameters manually pops up. The parameter will be used for the next fit if you press OK at the end.

Fit Again: Useful after defining either a range for the fit function or after setting the start parameters manually. After performing the fit the new function will be shown in the single-histogram window as well as in the thumbnail. The plot showing the fit parameter are updated with a different color to enable a comparison between the values before and after the new fit.

Tutorial: How to deal with channels without a reasonable data distribution

No good fit: If the calibration constants can't be extracted from a fit (e.g. sparse data for channel, fit is not converging etc.) this button pops up a menu which gives the user the following options:



Get Parameter for single channel: Sector 5, Stripe 20

Method to get parameter

● use default values

○ enter manually

○ copy from previous channel

○ average equivalent bars

○ select from database

gmean => param. 1: 600

Use default values: There is a default value for all calibration items, which is usually a rounded average over all channels. This value is normally a poor choice and should only be used if other methods fail.

Enter manually: If you think you know the correct value, you can enter it in the text field.

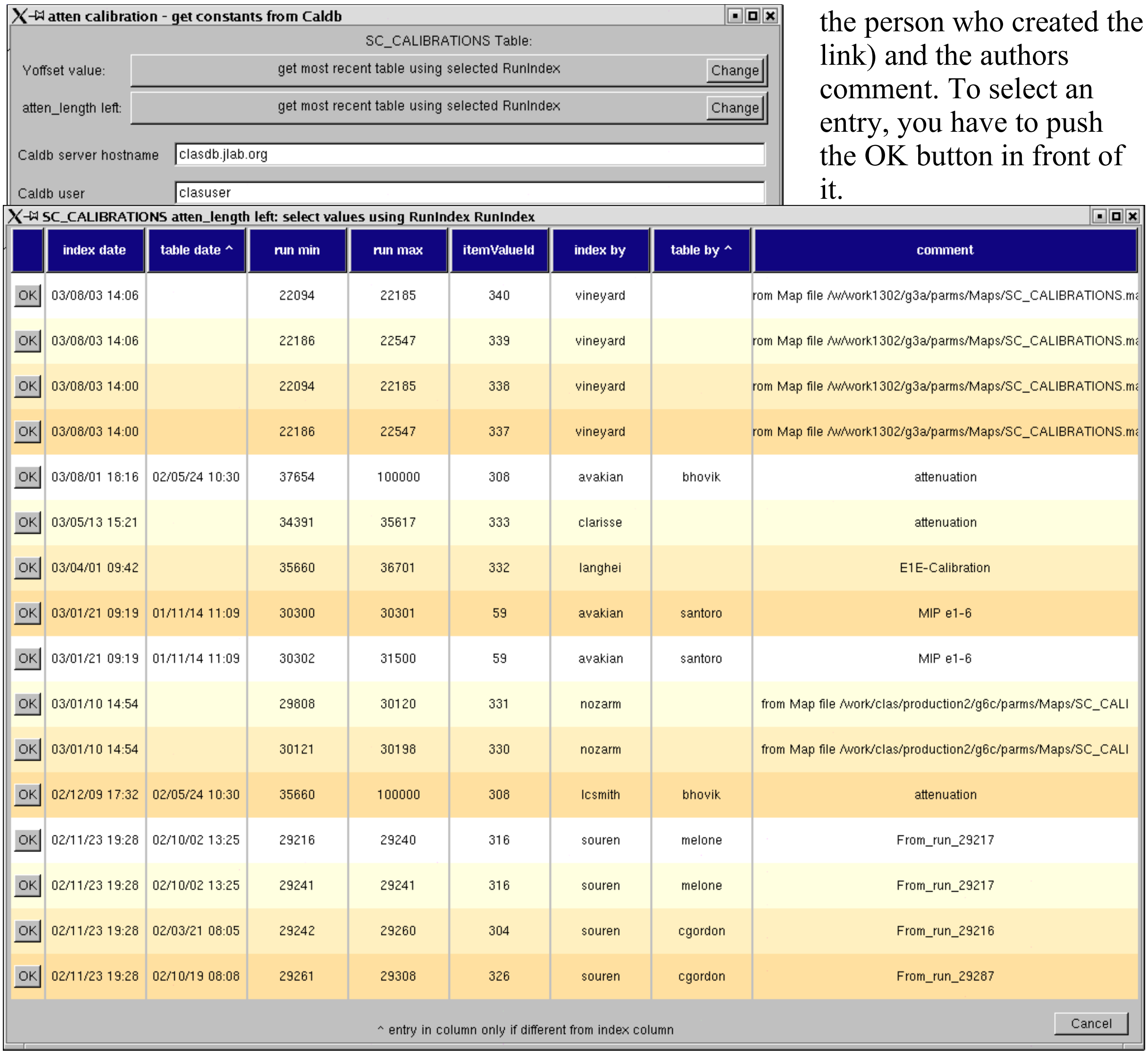
Copy from previous channel: Self explanatory, but not recommended.

Average equivalent bars: Uses the phi symmetry of the CLAS detector and assumes that the values for the geometrical equivalent channel in a different sector would be a good guess. This is true for items which depend strongly on the geometry like v_{eff}.

Select from database: The recommended method in our case. We try to locate the value from a recent calibration in the database.

Select from database: (the recommended method)

To access the database containing the results from previous calibrations, a runindex table (default: RunIndex)and a run number (in our case type 37665) are needed. The most commonly used server is clasdb.jlab.org (user clasuser, no password) but you are free to enter the name of one of the mirrors or of a private calibration database server of your own. After the form is completed and after pushing the OK button, gamecock tries to obtain the most recent value from the calibration database for this item. However, if this value is unreasonable you might want to go further back in history. To do so, you have to push the change button to change your current selection from the database. A new selection tool pops up, showing the date of the calibration (with two differnt values only if the date of creating the table differs from the date of linking the table to the runindex), the range of validity, the index number, the autor (showing two different names only if the author is not identical with the person who created the link) and the authors comment. To select an entry, you have to push the OK button in front of it.



X atten calibration - get constants from Caldb

SC_CALIBRATIONS Table:

Yoffset value:	get most recent table using selected RunIndex	Change
atten_length left:	get most recent table using selected RunIndex	Change
Caldb server hostname	clasdb.jlab.org	
Caldb user	clasuser	

X SC_CALIBRATIONS atten_length left: select values using RunIndex RunIndex

	index date	table date ^	run min	run max	itemValueId	index by	table by ^	comment
OK	03/08/03 14:06		22094	22185	340	vineyard		rom Map file /w/work1302/g3a/parms/Maps/SC_CALIBRATIONS.ma
OK	03/08/03 14:06		22186	22547	339	vineyard		rom Map file /w/work1302/g3a/parms/Maps/SC_CALIBRATIONS.ma
OK	03/08/03 14:00		22094	22185	338	vineyard		rom Map file /w/work1302/g3a/parms/Maps/SC_CALIBRATIONS.ma
OK	03/08/03 14:00		22186	22547	337	vineyard		rom Map file /w/work1302/g3a/parms/Maps/SC_CALIBRATIONS.ma
OK	03/08/01 18:16	02/05/24 10:30	37654	100000	308	avakian	bhovik	attenuation
OK	03/05/13 15:21		34391	35617	333	clarisse		attenuation
OK	03/04/01 09:42		35660	36701	332	langhei		E1E-Calibration
OK	03/01/21 09:19	01/11/14 11:09	30300	30301	59	avakian	santoro	MIP e1-6
OK	03/01/21 09:19	01/11/14 11:09	30302	31500	59	avakian	santoro	MIP e1-6
OK	03/01/10 14:54		29808	30120	331	nozarm		from Map file /work/clas/production2/g6c/parms/Maps/SC_CALI
OK	03/01/10 14:54		30121	30198	330	nozarm		from Map file /work/clas/production2/g6c/parms/Maps/SC_CALI
OK	02/12/09 17:32	02/05/24 10:30	35660	100000	308	lcsmith	bhovik	attenuation
OK	02/11/23 19:28	02/10/02 13:25	29216	29240	316	souren	melone	From_run_29217
OK	02/11/23 19:28	02/10/02 13:25	29241	29241	316	souren	melone	From_run_29217
OK	02/11/23 19:28	02/03/21 08:05	29242	29260	304	souren	cgordon	From_run_29216
OK	02/11/23 19:28	02/10/19 08:08	29261	29308	326	souren	cgordon	From_run_29287

^ entry in column only if different from index column

----------------- end of tutorial --

Running gamecock on a histogram file of my own.

Make sure that the histogram file has been created with one of the standard calibration programs or that it comprises the same histograms. There are two ways to select the file offered by the gamecock startup page, either by typing its name in th "input file" field or by using the file selection dialog button. If the file is a root file (recommended), it has to have the suffix .root in its name. Otherwise the file is assumed to be a hbook file, and the program tries to convert it to root using the tool $ROOTSYS/bin/h2root.

Running gamecock on a data file, creating the histograms.

If you want to use the program as a launcher, you have to install the standard calibration software at the usual location, e.g. $CLAS_PACK/utilities/sc_calib/atten/tof_calib to create the atten histograms. Select one or multiple (by using the wildcard characters like *,?) files using the text entry or the file selection dialog as mentioned above.

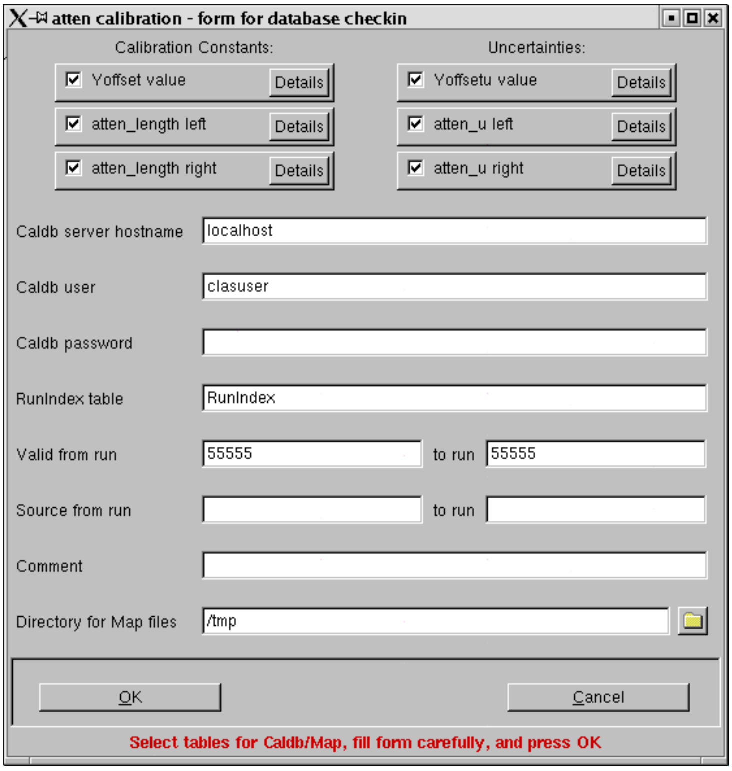
If the selected file has the suffix .set it is assumed to be a work set, containing a list of filenames to be worked through (about to be implemented).

Understanding the survey histograms

The fourth histogram shows the chisquare of the fit as reported by the fit function. The other three histograms are showing either fit parameter, the error of the fit or a magnitude calculated from the fit parameter, for example the product of the sigma of the peak times the scaling constant divided by the number of entries in the histogram. This magnitude should be constant if the majority of the entries can be found in the peak and the fit function describes the peak fairly well.

Check calibration constants into the Caldb (MySQL) database
You should be familiar with the Caldb database before you proceed with this step. A useful introduction is http://clasweb.jlab.org/caldb/caldb/
It is a good idea to doublecheck, that the calibration table doesn't contain zeros or unreasonable values before proceeding.
In most cases, there is a one to one relation in between the fit parameter and the constants to check into the database. There are however exceptions, where additional information from input files is needed to calculate the constants. In this case a query dialog pops up, asking for the location of these files. In the case of the gmean calibration the Y0offset is needed to calculate the constants. Unless you want to play with the program to check its functionality you have to do the atten calibration prior to the gmean calibration.

To proceed, click on CalDB->Check in. The main form to fill looks like this:



atten calibration - form for database checkin

Calibration Constants:

☑ Yoffset value Details
☑ atten_length left Details
☑ atten_length right Details

Uncertainties:

☑ Yoffsetu value Details
☑ atten_u left Details
☑ atten_u right Details

Caldb server hostname: localhost
Caldb user: clasuser
Caldb password:
RunIndex table: RunIndex
Valid from run: 55555 to run: 55555
Source from run: to run:
Comment:
Directory for Map files: /tmp

Select tables for Caldb/Map, fill form carefully, and press OK

Caldb server hostname: the IP address of the MySQL server to upload the constants. Most people might want to use the main server clasdb.jlab.org. Some people have a copy of the complete database on their local host which is also the default to avoid that the main server gets messed up inadvertently.
Caldb user: the MySQL user name. It is a good policy to use your login name rather than "clasuser" for check in.
Caldb password: The password to access the MySQL database if required. Leave field blank if no password required.
RunIndex table: See Caldb documentation. Use the public RunIndex to make your constants to be the default for all user, a private index like calib_user.RunIndex_xxx otherwise.
Valid from run ... to run: These pair of integer numbers defines the range of run numbers the calibration values are valid for
Source from run ... to run: The range of runs used for the calibration. Very often a single file has been used and "from" and "to" are the same numbers. These numbers are used for documentation only.
Comment: You are expected to enter something in this field, like "e1f calibration" or "test of new calibration tool". The comment is written to the table as well as to the runindex. For the runindex, system, subsystem and item name are added automatically in front of the comment, you don't have to type this information.