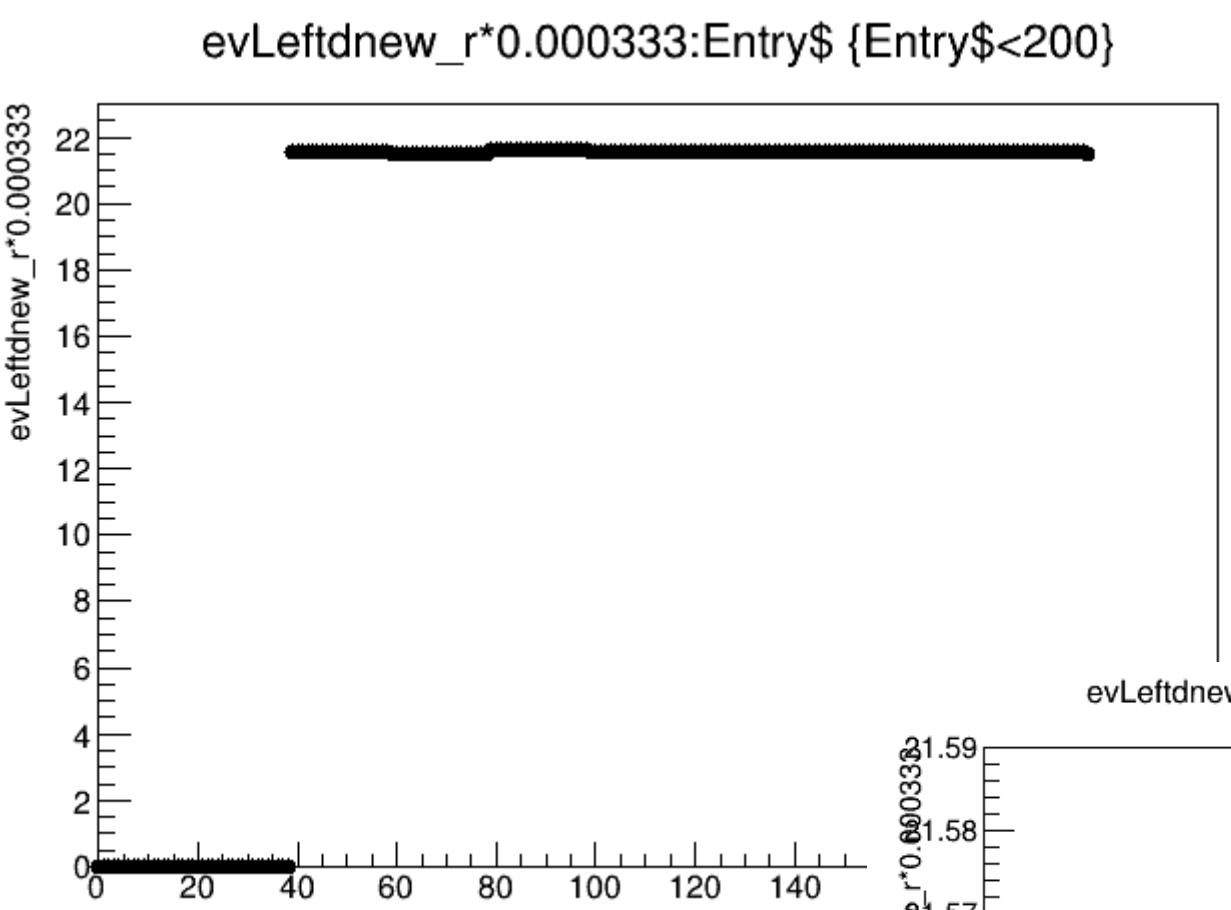
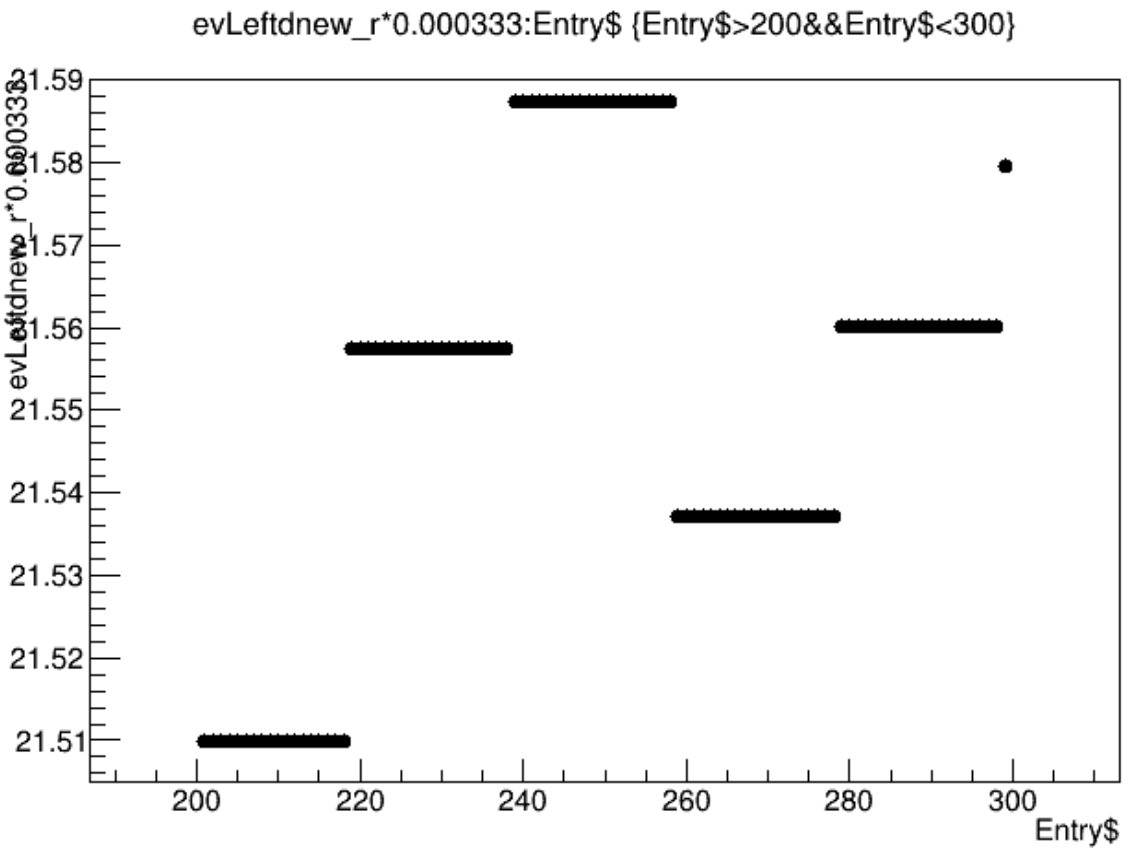
BCM offset


evLeftdnew_r*0.000333:Entry$ {Entry$<200}
evLeftdnew_r*0.000333
0
2
4
6
8
10
12
14
16
18
20
22
0
20
40
60
80
100
120
140


evLeftdnew_r*0.000333:Entry$ {Entry$>200&&Entry$<300}
21.51
21.52
21.53
21.54
21.55
21.56
21.57
21.58
21.59
200
220
240
260
280
300
Entry$

GenScaler/BCM class

- void GenScaler::LoadRates() {
- if (IsDecoded()) {
- Double_t dtime = GetTimeSincePrev();
- if (dtime==0) {
- memset(fRate, 0, fWordsExpect*sizeof(Double_t));
- return;
- }
- for (Int_t i=0; i<fWordsExpect; i++) {
- // Check for scaler overflow
- UInt_t diff;
- if(fDataArray[i] < fPrevData[i]) {
- diff = (kMaxUInt-(fPrevData[i]-1)) + fDataArray[i];
- } else {
- diff = fDataArray[i]-fPrevData[i];
- }
- ***fRate[i] = diff/dtime;***
- }
- }
- }

- TString bname[8] ={"u1","u3","u10","unew","d1","d3","d10","dnew"};
- bcm_name[i] = TString::Format("%s%s %s",scaler.Data(),arm.Data(),bname[i].Data());

  bcm_name_R[i] = TString::Format("%s%s %s_r",scaler.Data(),arm.Data(),bname[i].Data());
- //cout << bcm_name[i].Data() <<endl;
- }TString varnameR = bcm_name_R[i];
- Rate[i] = gHaVars->Find(varnameR.Data()
- bcms[i] = var[i]->GetValue();
- bcms_R[i] = Rate[i]->GetValue();
- // Calculate the charge and current if scalar is renewed
- count++;
- bcms_diff[i] = bcms[i]-bcm_old[i];
- average_current_event+= bcms_R[i]*gain[i]+ off[i];
- charge[i] = bcms_diff[i]*gain[i] + off[i]*time_sec;
- current[i] =bcms_R[i]*gain[i] + off[i];
- total_charge_event[i]+=charge[i]; //bcms[i]*gain[i] + off[i]*time_sec
- bcm_old[i]=bcms[i];
- }

Bcm offset

Entry#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
isrenewed	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
count	0	0	0	0	0	10	10	10	10	10	16	16	16	16	16	16	16	16	16	16	24	24	24	24	24	24	24
time	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0
rate	0	0	0	0	0	1	1	1	1	1	0.6	0.6	0.6	0.6	0.6	0	0	0	0	0	0.8	0.8	0.8	0.8	0.8	0	0
current	1	1	1	1	1	1	0.6	0.6	0.5	0.5	0.3	0.2	0	0	0	0	0.4	0.8	0.9	0.9	1	0	0	0	0	0	0
avg cur						1					0.5					0.04					0.8					0	

Bcm offset

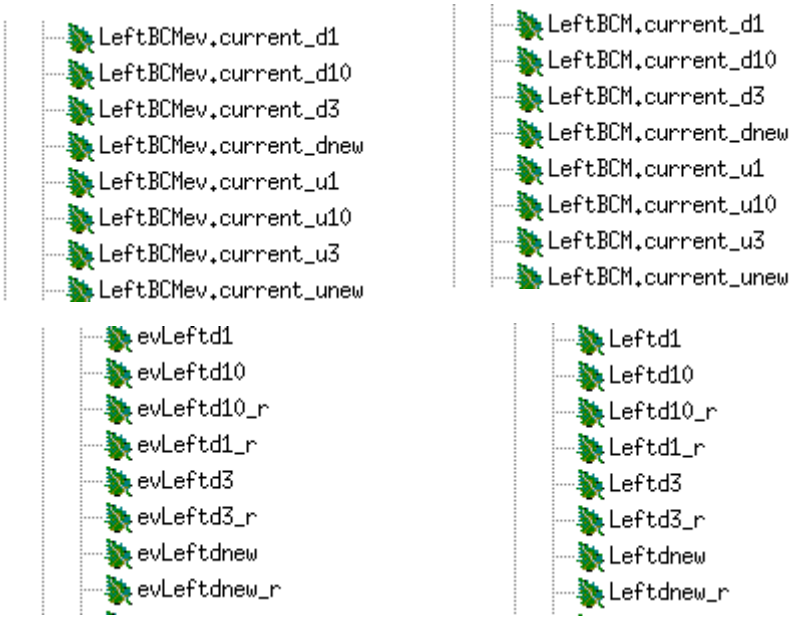
Entry#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
isrenewed	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
count	0	0	0	0	0	10	10	10	10	10	16	16	16	16	16	16	16	16	16	16	24	24	24	24	24	24	24
time	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0
rate	0	0	0	0	0	1	1	1	1	1	0.6	0.6	0.6	0.6	0.6	0	0	0	0	0	0.8	0.8	0.8	0.8	0.8	0	0
current	1	1	1	1	1	1	0.6	0.6	0.5	0.5	0.3	0.2	0	0	0	0	0.4	0.8	0.9	0.9	1	0	0	0	0	0	0
avg cur						1					0.5					0.04					0.8					0	

Bcm offset

Entry#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
isrenewed	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
count	0	0	0	0	0	10	10	10	10	10	16	16	16	16	16	16	16	16	16	16	24	24	24	24	24	24	24
time	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0
rate	0	0	0	0	0	1	1	1	1	1	0.6	0.6	0.6	0.6	0.6	0	0	0	0	0	0.8	0.8	0.8	0.8	0.8	0	0
current	1	1	1	1	1	1	0.6	0.6	0.5	0.5	0.3	0.2	0	0	0	0	0.4	0.8	0.9	0.9	1	0	0	0	0	0	0
avg cur						1					0.5					0.04					0.8					0	

BCM offset

Entry#	0	1	4	5	6	9	10	11	14	15	16	19	20	21	24	25
isrenewed	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
count	0	0	0	10	10	10	16	16	16	16	16	16	24	24	24	24
time	0	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10
rate	0	0	0	1	1	1	0.6	0.6	0.6	0	0	0	0.8	0.8	0.8	0
current	1	1	1	1	0.6	0.5	0.3	0.2	0	0	0.4	0.9	1	0	0	0
shift	1	1	1	0.6	0.6	0.6	0	0	0	0.8	0.8	0.8	0	0	0	0
avg	0.5	0.5	0.5	0.8	0.8	0.8	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0

Tree structure

- Leave old branch and add new
- Replace old with new.
- Make new Tree
- Make new file.
- Should I do this with all bcm current variable