

# Modifications to the Threshold Calculator Application

Matti Kalliokoski

15<sup>th</sup> BLM Thresholds WG Meeting

26/05/2015



**CERN**  
CH-1211 Geneva 23  
Switzerland



LHC Project Document No.  
**LHC-BLM-ES-0002 rev 0.1**

CERN Div./Group or Supplier/Contractor Document No.  
**BE-BI**

EDMS Document No.  
**1280100**

Date: 2013-04-11

## Technical Specification

# CALCULATION OF ABORT THRESHOLDS FOR THE BEAM LOSS MONITORING SYSTEM OF THE LARGE HADRON COLLIDER

### Abstract

*The Beam Loss Monitor system is made of about 4000 detectors spread around LHC ring. For each detector a set of operational parameters is prepared. Beam-abort thresholds are an example of safety-critical parameters. These parameters are calculated, modified and send to electronics when needed. This document explains step-by-step calculation procedure of safety-critical parameters and describes formulas, input parameters, functional requirements and graphical user interface needed to operate Beam Loss Monitoring abort threshold generator software.*

<i>Prepared by:</i> <b>Martin Nemcic</b> <b>Eduardo Nebot del Busto</b>	<i>Checked by:</i> <b>Lars Jensen</b> <b>Mariusz Sapinski</b> <b>Katarina Sigerud</b> <b>Jorg Wenninger</b> <b>Daniel Wollmann</b>	<i>Approval Leader:</i> <b>Bernd Dehning</b> <b>Eva Barbara Holzer</b> <b>Stephen Jackson</b> <b>Chris Roderick</b> <b>Markus Zerlauth</b>
---	---	---

## 2.1.3.1 Calculation of Maximum Number of Lost Protons for Warm Magnets

### Old Algorithm

*Linear weight for energy evolution*

$$\text{LinearWeight} = (\text{energy} - \text{eInjection}) / (\text{eColl} - \text{eInjection})$$

*Calculation of max number of protons in the first running sum at the given energy*

nShortInj and nShortInj are the given values of NPFAST at injection and collision energy

$$\text{nShort} = (\text{nShortInj} * (1. - \text{linearWeight})) + (\text{nShortColl} * \text{linearWeight})$$

*Calculation of max number of protons in the first running sum at the given energy dNdtLongInj and dNdtLongColl are the given values of PRMAX at injection and collision energy*

$$\text{nLong} = (\text{dNdtLongInj} * \text{time} * (1. - \text{linearWeight})) + (\text{dNdtLongColl} * \text{time} * \text{linearWeight})$$

*Logarithmic difference between last (83 s) and first (40us) integration time*

$$\text{dTTime} = \log(\text{time}[12]) - \log(\text{time}[1])$$

*Logarithmic weight for time evolution*

$$\text{timeWeight} = (\log(\text{time}) - \log(\text{time}[1])) / \text{dTTime}$$

*Calculation of max number of protons*

$$\text{Np} = ((1. - \text{timeWeight}) * \text{nShort}) + (\text{timeWeight} * \text{nLong})$$

### New Algorithm

*Calculation of max number of protons in the first running sum at the given energy:*

$$\text{nShort} = \text{interpolation}(\text{Ebeam})$$

*where the interpolation algorithm accepts positive and negative exponents (we need fits of the form aShort-1 x-1 + aShort1 x)*

$$\text{nLong} = \text{interpolation}(\text{Ebeam}) * \text{time}$$

*where the interpolation algorithm accepts positive and negative exponents (we need fits of the form aLong-1 x-1 + aLong1 x)*

*Logarithmic difference between last (83 s) and first (40us) integration time*

$$\text{dTTime} = \log(\text{time}[12]) - \log(\text{time}[1])$$

*Logarithmic weight for time evolution*

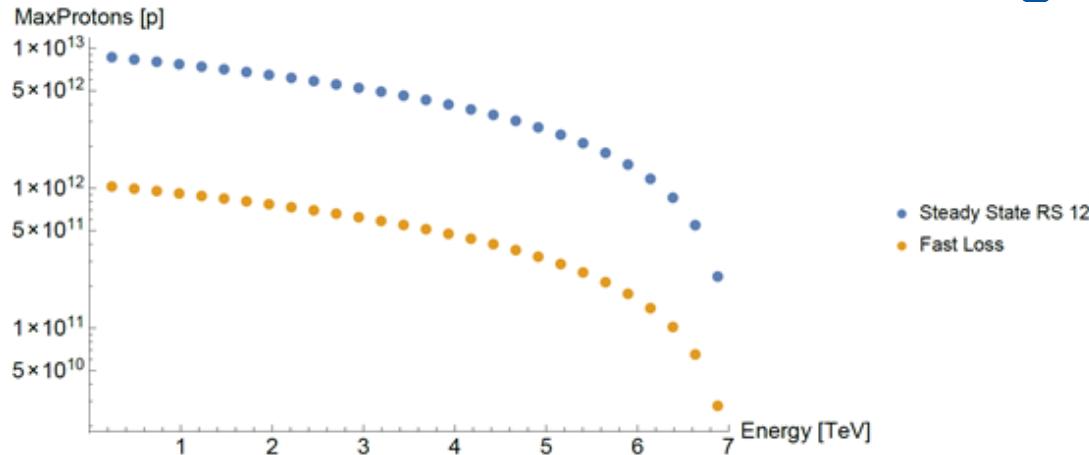
$$\text{timeWeight} = (\log(\text{time}) - \log(\text{time}[1])) / \text{dTTime}$$

*Calculation of max number of protons*

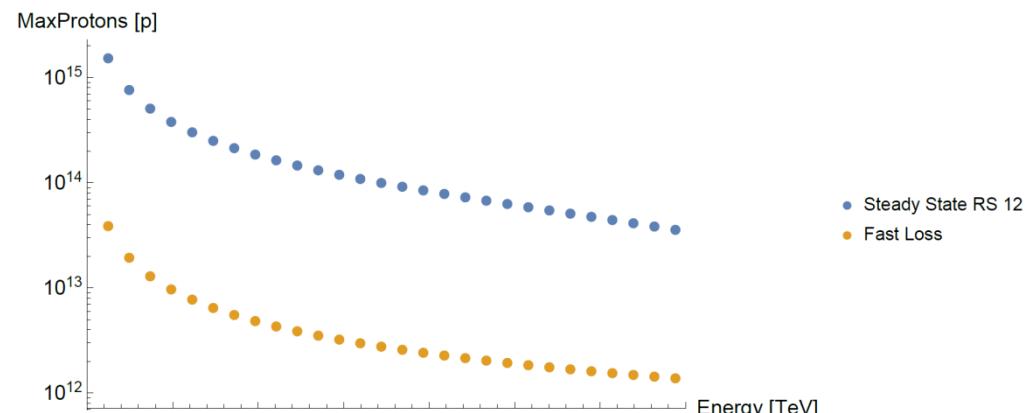
$$\text{Np} = ((1. - \text{timeWeight}) * \text{nShort}) + (\text{timeWeight} * \text{nLong})$$



## 2.1.3.1 Calculation of Maximum Number of Lost Protons for Warm Magnets



Old Algorithm



New Algorithm

# 2.1.4.1 Maximum Number of Lost Protons for Collimators

## Old Algorithm

```
- FAST LOSS (tfast = 1 s)
if (time < tfast){
    Calculating energy dependent slope and offset
    shortSlope = (nShortColl - nShortInj) / (eColl - elnjection)
    shortOffSet = nShortInj - (shortSlope * elnjection)
    Calculating raw max number of protons
    nProtF = shortOffSet + shortSlope * energy
    Ultra fast loss correction (only for the first 5 running sums)
    if (time == time[1])
        nProtF *= blmCorrUF[1]
    if (time == time[2])
        nProtF *= blmCorrUF[2]
    if (time == time[3])
        nProtF *= blmCorrUF[3]
    if (time == time[4])
        nProtF *= blmCorrUF[4]
    if (time == time[5])
        nProtF *= blmCorrUF[5]
    } end 'if fast losses

- SLOW LOSSES LOSS (tslow = 10 s)
if (time > tfast && time < tslow){
    Calculating energy dependent slope and offset
    mediumSlope = (dNdtMidColl - dNdtMidInj) / (eColl - elnjection)
    mediumOffset = dNdtMidInj - (mediumSlope * elnjection)
    Calculating raw max number of protons
    nProtF = (mediumOffset + mediumSlope * energy) * time
} end 'if slow losses

- STEADY STATE LOSS (tslow = 10 s)
if (time > tSteady){
    Calculating energy dependent slope and offset (slow loss case)
    mediumSlope = (dNdtMidColl - dNdtMidInj) / (eColl - elnjection)
    mediumOffset = dNdtMidInj - (mediumSlope * elnjection)
    Calculating energy dependent slope and offset (steady-state case)
    longSlope = (dNdtLongColl - dNdtLongInj) / (eColl - elnjection)
    longOffSet = dNdtLongInj - (longSlope * elnjection)
    Calculating raw max number of protons
    nProtF = (mediumOffset + mediumSlope * energy) * tslow
    nProtF += (time - tslow) * (longOffSet + longSlope * energy)
} end 'if steady state losses
```

## New Algorithm

*Calculating the loss rate for each energy level*

$$\text{LossRate500kW} = 500 \text{ kW / E}$$
$$\text{LossRate100kW} = 100 \text{ kW / E}$$

*Calculation of number of lost protons*

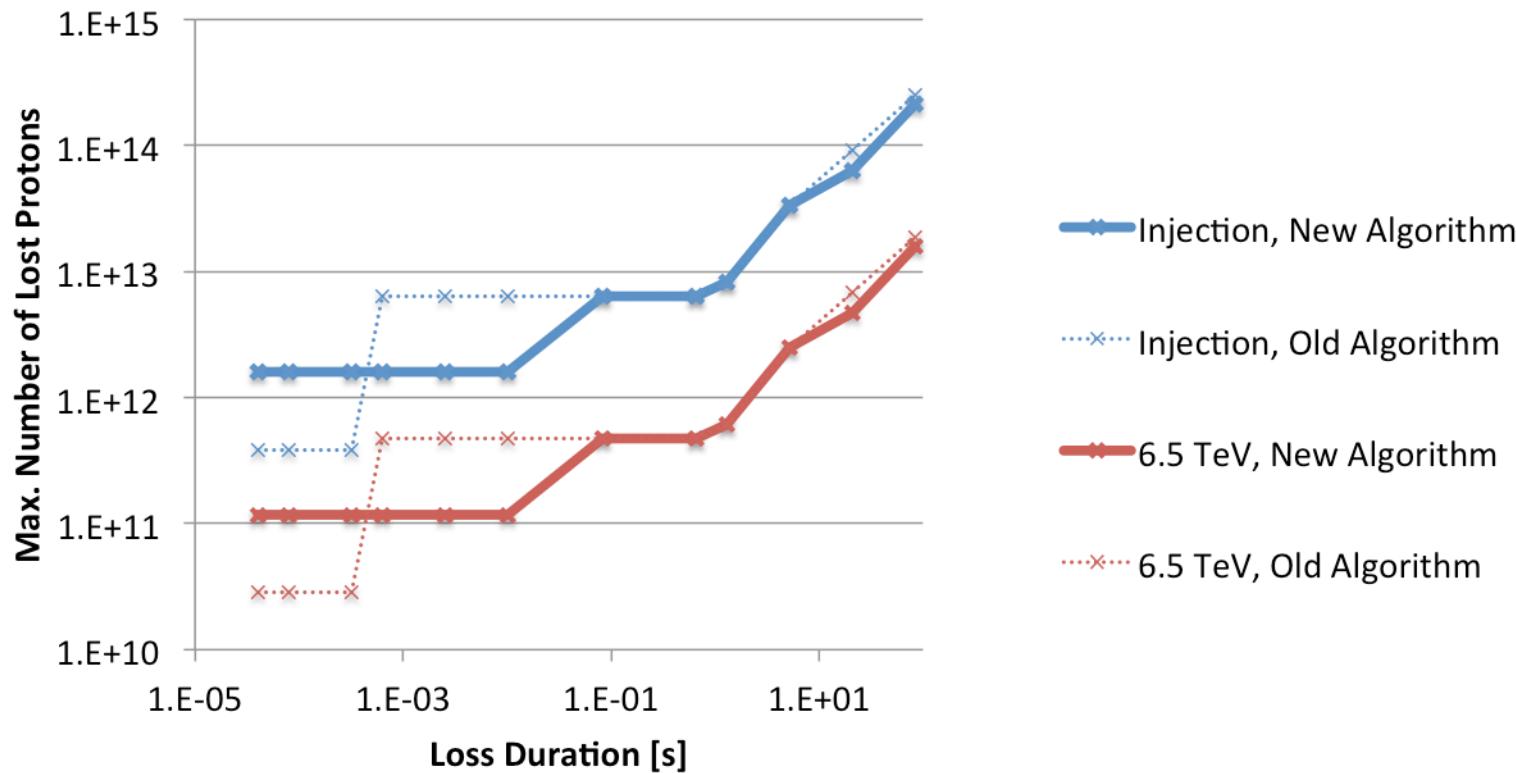
```
if time = time[12] // RS12
    nProtE = LossRate100kW * time

elseif time = time[9-11] // RS09-11
    nProtE = min(time, time[9-11]) *
    LossRate500kW

elseif time = time[7-8] // RS06-07
    nProtE = 1 s * LossRate500kW

else // RS01-05
    nProtE = 0.25 s * LossRate500kW
```

## 2.1.4.1 Maximum Number of Lost Protons for Collimators



## 2.3.1 Beam Energy Levels

Beam Energy	Value	Units	Type
Energy 1	0.24564	TeV	double
Energy 2	0.49140	TeV	double
Energy 3	0.73716	TeV	double
Energy 4	0.98292	TeV	double
Energy 5	1.22868	TeV	double
Energy 6	1.47400	TeV	double
Energy 7	1.72020	TeV	double
Energy 8	1.96596	TeV	double
Energy 9	2.21172	TeV	double
Energy 10	2.45748	TeV	double
Energy 11	2.70324	TeV	double
Energy 12	2.94900	TeV	double
Energy 13	3.19476	TeV	double
Energy 14	3.44052	TeV	double
Energy 15	3.68628	TeV	double
Energy 16	3.93204	TeV	double
Energy 17	4.17780	TeV	double
Energy 18	4.42356	TeV	double
Energy 19	4.66932	TeV	double
Energy 20	4.91508	TeV	double
Energy 21	5.16084	TeV	double
Energy 22	5.40660	TeV	double
Energy 23	5.65236	TeV	double
Energy 24	5.89812	TeV	double
Energy 25	6.14388	TeV	double
Energy 26	6.38964	TeV	double
Energy 27	6.63540	TeV	double
Energy 28	6.88116	TeV	double
Energy 29	7.12692	TeV	double
Energy 30	7.37268	TeV	double
Energy 31	7.61844	TeV	double
Energy 32	7.86420	TeV	double

- At the moment the table values start from 0.24564 TeV
- This value is not used for machine protection
- We would like to change the first value to match either the second energy level or fix the value to 450 GeV
- At the bottom the values should be fixed to 7 TeV
  - Would reduce noise issues

## 2.3.1 Beam Energy Levels

Beam Energy	Value	Units	Type
Energy 1	0.45000	TeV	double
Energy 2	0.49140	TeV	double
Energy 3	0.73716	TeV	double
Energy 4	0.98292	TeV	double
Energy 5	1.22868	TeV	double
Energy 6	1.47400	TeV	double
Energy 7	1.72020	TeV	double
Energy 8	1.96596	TeV	double
Energy 9	2.21172	TeV	double
Energy 10	2.45748	TeV	double
Energy 11	2.70324	TeV	double
Energy 12	2.94900	TeV	double
Energy 13	3.19476	TeV	double
Energy 14	3.44052	TeV	double
Energy 15	3.68628	TeV	double
Energy 16	3.93204	TeV	double
Energy 17	4.17780	TeV	double
Energy 18	4.42356	TeV	double
Energy 19	4.66932	TeV	double
Energy 20	4.91508	TeV	double
Energy 21	5.16084	TeV	double
Energy 22	5.40660	TeV	double
Energy 23	5.65236	TeV	double
Energy 24	5.89812	TeV	double
Energy 25	6.14388	TeV	double
Energy 26	6.38964	TeV	double
Energy 27	6.63540	TeV	double
Energy 28	6.88116	TeV	double
Energy 29	7.00000	TeV	double
Energy 30	7.00000	TeV	double
Energy 31	7.00000	TeV	double
Energy 32	7.00000	TeV	double

- At the moment the table values start from 0.24564 TeV
- This value is not used for machine protection
- We would like to change the first value to match either the second energy level or fix the value to 450 GeV
- At the bottom the values should be fixed to 7 TeV
  - Would reduce noise issues

# 3.3 IL Correction

Correction ID	ilCorrection
Input Parameters	timeConstant, norm, blmConvBit2Gy
Description	The function allows to increase the dump threshold up to norm (value Gy/s) in RS01. The allowed dose in longer running sums corresponds to an exponential decay (with decay time timeConstant). This correction is applied to specific monitors in the injection regions.

# 3.3 IL Correction

## Old version

### 1. Compute normalization.

```
norm /= (1.0 - exp(-1.0 * integrationTime[0] / timeConstant))
```

### 2. Energy [2] loop.

```
for (int energy = 0; energy < 2; energy++) {
```

### 3. Integration Time [12] loop.

```
for (int time = 0; time < 12; time++) {
```

### 4. Compute injection loss.

```
injectionLoss = (norm / blmConvBit2Gy) * (1.0 - exp(-  
integrationTime[it] / timeConstant))
```

### 5. Compare injection loss with computed thresholds.

```
if (injectionLoss > ThresholdsValue[energy][time]) {
```

### 6. If injection loss is bigger than threshold value, replace threshold value with injection loss.

```
ThresholdsValue[energy][time] = injectionLoss
```

### 7. End of 'if' statement.

```
}
```

## Fixed version

### 1. Compute normalization.

```
norm /= exp(-1.0 * integrationTime[0] / timeConstant)
```

### 2. Energy [2] loop.

```
for (int energy = 0; energy < 2; energy++) {
```

### 3. Integration Time [12] loop.

```
for (int time = 0; time < 12; time++) {
```

### 4. Compute injection loss.

```
injectionLoss = (norm / blmConvBit2Gy) * exp(-1.0 *  
integrationTime[time] / timeConstant) * integrationTime[time]
```

### 5. Compare injection loss with computed thresholds.

```
if (injectionLoss > ThresholdsValue[energy][time]) {
```

### 6. If injection loss is bigger than threshold value, replace threshold value with injection loss.

```
ThresholdsValue[energy][time] = injectionLoss
```

### 7. End of 'if' statement.

```
}
```

# 3.8-3.10 Ad-Hoc Corrections

1. *Integration Time [12] loop.*

```
for (int rsum = 0; rsum < 12; rsum++) {
```

2. *Energy [32] loop.*

```
for (int energyLevel = 0; energyLevel < 32; energyLevel++) {
```

3. *Checks if beam levels are specified.*

```
if (beamLevel.isEmpty) {
```

4. *Scaling all beam energy(32) levels by given factor.*

```
ThresholdsValue[energyLevel][rsum] *= scaleRS[rsum]
```

5. *Else if, scale only specified beam level positions.*

```
} else {
```

6. *Correct specified beam level positions and not all 32.*

7. *End of 'else if' statement.*

```
}
```

8. *End of Energy loop.*

```
}
```

9. *End of Integration time loop.*

```
}
```



# 3.8-3.10 Ad-Hoc Corrections

1. Integration Time [12] loop.

```
for (int rsum = 0; rsum < 12; rsum++) {
```

2. Energy [32] loop.

```
for (int energyLevel = 0; energyLevel < 32; energyLevel++) {
```

**3. Checks if beam levels are specified.**

```
if (beamLevel.isEmpty) {
```

**4. Scaling all beam energy(32) levels by given factor.**

```
ThresholdsValue[energyLevel][rsum] *= scaleRS[rsum]
```

**5. Else-if, scale only specified beam level positions.**

```
} else {
```

6. Correct specified beam level positions and not all 32.

**7. End of 'else if' statement.**

```
}
```

8. End of Energy loop.

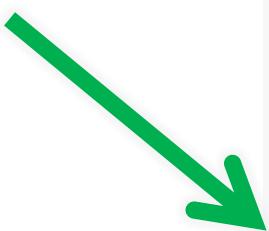
```
}
```

9. End of Integration time loop.

```
}
```

<input checked="" type="checkbox"/> Beam_lvl 0	<input checked="" type="checkbox"/> Beam_lvl 16	ScaleRS 0	4.0
<input checked="" type="checkbox"/> Beam_lvl 1	<input checked="" type="checkbox"/> Beam_lvl 17	ScaleRS 1	4.0
<input checked="" type="checkbox"/> Beam_lvl 2	<input checked="" type="checkbox"/> Beam_lvl 18	ScaleRS 2	4.0
<input checked="" type="checkbox"/> Beam_lvl 3	<input checked="" type="checkbox"/> Beam_lvl 19	ScaleRS 3	4.0
<input checked="" type="checkbox"/> Beam_lvl 4	<input checked="" type="checkbox"/> Beam_lvl 20	ScaleRS 4	4.0
<input checked="" type="checkbox"/> Beam_lvl 5	<input checked="" type="checkbox"/> Beam_lvl 21	ScaleRS 5	4.0
<input checked="" type="checkbox"/> Beam_lvl 6	<input checked="" type="checkbox"/> Beam_lvl 22	ScaleRS 6	2.0
<input checked="" type="checkbox"/> Beam_lvl 7	<input checked="" type="checkbox"/> Beam_lvl 23	ScaleRS 7	1.0
<input checked="" type="checkbox"/> Beam_lvl 8	<input checked="" type="checkbox"/> Beam_lvl 24	ScaleRS 8	1.0
<input checked="" type="checkbox"/> Beam_lvl 9	<input checked="" type="checkbox"/> Beam_lvl 25	ScaleRS 9	1.0
<input checked="" type="checkbox"/> Beam_lvl 10	<input checked="" type="checkbox"/> Beam_lvl 26	ScaleRS 10	1.0
<input checked="" type="checkbox"/> Beam_lvl 11	<input checked="" type="checkbox"/> Beam_lvl 27	ScaleRS 11	1.0
<input checked="" type="checkbox"/> Beam_lvl 12	<input checked="" type="checkbox"/> Beam_lvl 28		
<input checked="" type="checkbox"/> Beam_lvl 13	<input checked="" type="checkbox"/> Beam_lvl 29		
<input checked="" type="checkbox"/> Beam_lvl 14	<input checked="" type="checkbox"/> Beam_lvl 30		
<input checked="" type="checkbox"/> Beam_lvl 15	<input checked="" type="checkbox"/> Beam_lvl 31		

[Set all beam levels](#) [Unset all beam levels](#)



# 3.10 Ad-Hoc Bits Correction

RC\_CORRECTION  
IL\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
DECREASE\_CORRECTION  
MAX\_BITS\_CORRECTION  
MIN\_BITS\_CORRECTION  
**AD\_HOC\_FACTOR\_CORRECTION**  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION  
AD\_HOC\_FACTOR\_CORRECTION

Beam_lvl 0	Beam_lvl 16	ScaleRS 0	1.1017
Beam_lvl 1	Beam_lvl 17	ScaleRS 1	1.0
Beam_lvl 2	Beam_lvl 18	ScaleRS 2	1.0
Beam_lvl 3	Beam_lvl 19	ScaleRS 3	1.0
Beam_lvl 4	Beam_lvl 20	ScaleRS 4	1.0
Beam_lvl 5	Beam_lvl 21	ScaleRS 5	1.0
Beam_lvl 6	<input checked="" type="checkbox"/> Beam_lvl 22	ScaleRS 6	1.0
Beam_lvl 7	Beam_lvl 23	ScaleRS 7	1.0
Beam_lvl 8	Beam_lvl 24	ScaleRS 8	1.0
Beam_lvl 9	Beam_lvl 25	ScaleRS 9	1.0
Beam_lvl 10	Beam_lvl 26	ScaleRS 10	1.0
Beam_lvl 11	Beam_lvl 27	ScaleRS 11	1.0
Beam_lvl 12	Beam_lvl 28		
Beam_lvl 13	Beam_lvl 29		
Beam_lvl 14	Beam_lvl 30		
Beam_lvl 15	Beam_lvl 31		

MOVE UP   MOVE DN  
AD\_HOC\_BITS\_CORRECTION  
ADD   Delete  
COMMIT  
Set all beam levels   Unset all beam levels  
Description of correction: AD\_HOC\_FACTOR\_CORRF  
Order ID: 7   Save   Cancel

- Originally empty fields for RS were not allowed
  - Values were looped only over energy, not time
- Bits Correction could not be used
  - Repetition of other corrections were needed to do a simple change

# 3.10 Ad-Hoc Bits Correction

<input type="checkbox"/> Beam_lvl 0	<input type="checkbox"/> Beam_lvl 16	ScaleRS 0	2000
<input type="checkbox"/> Beam_lvl 1	<input type="checkbox"/> Beam_lvl 17	ScaleRS 1	2000
<input type="checkbox"/> Beam_lvl 2	<input type="checkbox"/> Beam_lvl 18	ScaleRS 2	
<input type="checkbox"/> Beam_lvl 3	<input type="checkbox"/> Beam_lvl 19	ScaleRS 3	
<input type="checkbox"/> Beam_lvl 4	<input type="checkbox"/> Beam_lvl 20	ScaleRS 4	
<input type="checkbox"/> Beam_lvl 5	<input type="checkbox"/> Beam_lvl 21	ScaleRS 5	
<input type="checkbox"/> Beam_lvl 6	<input type="checkbox"/> Beam_lvl 22	ScaleRS 6	
<input type="checkbox"/> Beam_lvl 7	<input type="checkbox"/> Beam_lvl 23	ScaleRS 7	
<input type="checkbox"/> Beam_lvl 8	<input type="checkbox"/> Beam_lvl 24	ScaleRS 8	
<input type="checkbox"/> Beam_lvl 9	<input type="checkbox"/> Beam_lvl 25	ScaleRS 9	
<input type="checkbox"/> Beam_lvl 10	<input type="checkbox"/> Beam_lvl 26	ScaleRS 10	
<input type="checkbox"/> Beam_lvl 11	<input checked="" type="checkbox"/> Beam_lvl 27	ScaleRS 11	
<input type="checkbox"/> Beam_lvl 12	<input checked="" type="checkbox"/> Beam_lvl 28		
<input type="checkbox"/> Beam_lvl 13	<input checked="" type="checkbox"/> Beam_lvl 29		
<input type="checkbox"/> Beam_lvl 14	<input checked="" type="checkbox"/> Beam_lvl 30		
<input type="checkbox"/> Beam_lvl 15	<input checked="" type="checkbox"/> Beam_lvl 31		

- Modification to the application was made
  - App passes a flag to API to allow empty fields in RS
  - Loop over RS was introduced for Bits correction

# Conclusions

- Series of modifications to the Threshold Calculator is to be made
- Some bug fixes have already been introduced
- Corresponding document LHC-BLM-ES-0002, EDMS No. 1280100 will be updated accordingly
- All the changes are also reported in JIRA:  
<https://issues.cern.ch/browse/BLMDM-69>