Updated April 2021

[1. Helpful links](#_e6d1himnao9y)

[2. General procedure](#_sm6q81m7jgpv)

[3. Alternative for LS2](#_qkrtu18jolu5)

[4. Examining high deviation channels](#_26j83327rxi)

[5. COOL ROBOT tags](#_9uyts96plfp3)

[6. Converting sqlite file tags](#_1ye00cto16oh)

# 1. Helpful links

[TileCisCalibration Twiki](https://twiki.cern.ch/twiki/bin/view/Atlas/TileCisCalibration): Overview of CIS calibration

[TileCisCalibrationProcedure Twiki](https://twiki.cern.ch/twiki/bin/viewauth/Atlas/TileCisCalibrationProcedure): Information about CIS updates and database tags

[PrimaryCisTools Twiki](https://twiki.cern.ch/twiki/bin/view/Atlas/PrimaryCisTools): Detailed description of useful macros and plotting tools.

[2018 CIS guide](https://drive.google.com/file/d/1vojCK_cZTw7vfyHZnvRJ4DlovwmpdBey/view?usp=sharing): Detailed overview of CIS tech work, including an explanation of updates and databases.

[TUCS CIS plots guide](https://docs.google.com/document/d/14RxNsx61A0L8cnDrbXUuIAbqMpTlqutxSXcQyy_V3zc/): Working execution examples for plotting procedures, more narrow but inspired by PrimaryCisTools.

Sasha (Alexander) Solodkov and Henric Wilkens will be your primary contacts for CIS updates.

# 2. General procedure

(not updated past step 7 since June 2020)

1. **Ssh into lxplus account**

* ssh -XY USER@lxplus.cern.ch

1. **Setup environment**
   1. Athena is the environment we use and it has a bunch of different versions
   2. You can do:

* asetup 20.3.3,here → older version for old code you might use
* asetup Athena 22.0.36 → Newer version

1. **Go to TUCS**

* This is where all the macros & workers are

1. **Remove any tileSqlite.db file in TUCS/results directory**

* Otherwise the new database file you obtain will not be useful

1. **Run CIS\_DB\_Update.py macro**

* macros/cis/CIS\_DB\_Update.py --date '-28 days' (siphone output using >> to find date)
* The number of days can change depending on the date of the last update
* This will give you three sets of files: a bunch of plots, a .db file in the results directory, and a .txt file in the results directory

1. **Examine Plots**

* When you see unstable channel or averages far from the detector average make a note to add a Bad CIS status to that ADC
* If you see any weird point (a point is a run) you might need to run the investigate macro to examine that specific point.

1. **Presentation**

* You will show your results during the 3:30 pm Monday meeting
* Include the histogram showing the # of ADCs with greater than 0.7% change
* Pull plots and histogram using *scp* command from a new terminal instance:
  + *scp -r eoakes@lxplus.cern.ch:~/public/Tucs/plots/latest/cis/ ~/Desktop/*
* Use the txt file in the results directory to get the number of ADCs that were updated and other information
* See Section 4 to examine channels with high DB deviation (>5%), include these in your presentation
* Show the plots for the channels that you want to add / remove flags for

1. **Change the comment and tag in the DB file** (highlighted = change this to be the name of your sqlite file)

* Be sure to change the comment of the sqlite file first:
  + Navigate to directory containing WriteDBMultipleIOV.py worker (Tucs/workers/) and edit the comment at the bottom
* Change the tag from UPD1 to UPD4 before uploading

**$ AtlCoolCopy "sqlite://;schema=tileSqlite.db;dbname=CONDBR2" "sqlite://;schema=tileSqlite\_with\_UPD4.db;dbname=CONDBR2" -ot TileOfl02CalibCisLin-RUN2-UPD4-16 -create**

* You can also read the contents of your sqlite file with the following:

**$ ReadCalibFromCool.py --schema='sqlite://;schema=tileSqlite\_with\_UPD4.db;dbname=CONDBR2'**

1. **Send the file to Yuri Smirnov and upload constants to database**

* Access robot from any of the links below. Choose CISLIN\_ALL
* <https://atlas-tile-calib-dev2.cern.ch/>
* <https://atlas-tile-calib-dev2.cern.ch/v1/>
* ...
* <https://atlas-tile-calib-dev2.cern.ch/v9>

1. **COOL Robot Database Update**

* navigate to one of the following links:

<https://atlas-tile-calib-dev2.cern.ch/>

<https://atlas-tile-calib-dev2.cern.ch/v1/>

…

<https://atlas-tile-calib-dev2.cern.ch/v9>

* Click on robot icon, then wait
* If nobody is logged in, you will be able to start a new session
* Click on the green CISLIN-ALL button at the button right
* Either upload a SQLite file with the updated constants and UPD4 tag or enter changes manually in the fields provided
* Submit changes and make sure to **exit the session**, otherwise others will not be able to access the Robot

1. **COOL status update script (don’t need this during LS2)**

* You will keep track of all the COOL Flag changes (BadCis, NoCis etc)
* There is a python script that /afs/cern.ch/user/a/amattill/public/
* If you can’t access it go to Hazal’s: /afs/cern.ch/user/h/hgoksu/public/TileCalorimeter/TileCalib/Tucs/CIS\_COOL\_UpdatesWr
* Copy that script and edit it, it has instructions
* Test the script by running the below command. It should create an sqlite file if you did everything correct. You can delete that file later  
   **WriteBchToCool.py --execfile=whatever\_you\_name\_thefile.py**
* Once you are sure the script works, send it to Yuri Smirnov yet again!

# 3. Alternative for LS2

1. **Read db into text file**

Go to results folder in Tucs

* 1. ReadCalibFromCool.py --schema="sqlite://;schema=tileSqlite.db;dbname=CONDBR2" --folder=/TILE/OFL02/CALIB/CIS/LIN --tag=UPD1 | grep -v miss > cis.txt

1. **Make changes to problematic channels**
   1. Can use --region flag in standard update to check changing constants (don’t overwrite database!) (or just use StudyFlag.py as outlined in the [TUCS plots doc](https://docs.google.com/document/d/14RxNsx61A0L8cnDrbXUuIAbqMpTlqutxSXcQyy_V3zc/edit))
   2. Make one file for channels already in update (corr1.txt), one for channels not in update (corr2.txt). Channels that are not in the update will have plots labeled as such. Channels that are in the update can also be seen in the CIS\_DB\_Update file.

i.e.: Drawer Chan Gain Value (don’t include header)

EBA01 1 0 1.27

1. **Use WriteCalibToCool.py to update local db**
   1. WriteCalibToCool.py --schema="sqlite://;schema=tileSqlite.db;dbname=CONDBR2" --txtfile=corr1.txt --folder=/TILE/OFL02/CALIB/CIS/LIN --tag=UPD1 --run=XXXXXX
      1. --run=XXXXXX from CIS\_DB\_update.txt
   2. WriteCalibToCool.py --inschema=COOLOFL\_TILE/CONDBR2 --outschema="sqlite://;schema=tileSqlite.db;dbname=CONDBR2" --update --txtfile=corr2.txt --folder=/TILE/OFL02/CALIB/CIS/LIN --tag=UPD1 --run=XXXXXX
   3. Check constants with ReadCalibFromCool.py as in 8a
2. **Run calib\_to\_oracle to generate UPD1-, UPD4-, and ONL-tagged databases**
   1. ~solodkov/scripts/calib\_to\_oracle CALIB/CIS/LIN tileSqlite.db
3. **Use ROBOT to upload db**
   1. <https://atlas-tile-calib.cern.ch/>
   2. Upload UPD4-tagged sqlite to CISLIN\_ALL
4. **Use WriteBchToCool.py to update COOL flags**
   1. Change update\_cis.py to reflect desired changes
   2. WriteBchToCool.py --execfile=update\_cis.py --online --upd4
   3. Upload tileSqlite\_upd4.db to ADC\_UPD4\_UPD1

Check the folder, tag, and IOV with AtlCoolConsole.py (?)

# 4. Examining high deviation channels

For channels with high deviation from the databases (>5%), it may be necessary to compare reconstructed laser amplitudes with the CIS constant updates. To look at laser amplitudes, we plot the ratio of signals in the channel of interest against a stable channel in the same module, where a “stable” channel is simply a channel with a signal and CIS constant that was not updated.

1. **Select channel for comparison** 
   1. Compare two odd or two even PMTs
      1. Laser signal sent to all odd or all event PMTs, so ratio should be consistent for same-parity PMTs on the same fibre
   2. For EBA/EBC, use ch. 16 and 17 (D5). If 16 and 17 aren’t stable, use ch. 37 and 38 (D6). If none are stable, it’s okay to use other channels for simple validation, but avoid ch. 0, 1, 12, and 13 (E-cells).
   3. For LBA/LBC, use ch. 13 and 14 (D1). If 13 and 14 aren’t stable, use ch. 24 and 25 (D2).
2. **Double-check your indices and channel numbers!** 
   1. LBA=0, LBC=1, EBA=2, EBC=3
   2. Module number is zero-indexed, i.e. EBA13 -> EBA12
   3. Be careful with channel-to-PMT indexing - check conversion [here](http://zenis.dnp.fmph.uniba.sk/tile.html). We want same-parity PMTs, not channels.
3. **Make plots in ROOT or with /afs/cern.ch/user/e/eoakes/public/show\_las.py**
   1. Show when CIS constants were updated with a line
   2. Use a time frame longer than a single CIS update
   3. Filter laser signals at zero, also use wheelpos==8 for high gain channels
4. **Compare with CIS constants**
   1. Look at the jumps in the laser signal around the CIS update times
   2. If the laser signal returns to stability, good update
   3. If the laser signal deviates more, or the CIS constant is being dragged away from the real value, bad update
      1. Try to adjust the CIS constant by hand as in step 9 above

Alternatively, talk to Henric Wilkens and the laser team about their plots, i.e. <http://hwilkens.web.cern.ch/hwilkens/Tucs/plots/latest/LBA.pdf>. These plots show both gains with response corrected, but are difficult to use because of their size and long date range.

# 5. COOL ROBOT tags

<https://atlas-tile-calib.cern.ch/>

* ADCUPD4: update UPD4 only
* ADC\_UPD4\_UPD1: update UPD4+UPD1 simultaneously
* ADC\_UPD1\_ONL: update UPD1 and ONL simultaneously (cannot be used with single sqlite file yet, since very different statuses for UPD1 and ONL)
* ADC\_UPD4\_UPD1\_ONL: update all 3 but avoid atm for similar reasons
* ADCtrigger: update ONL only (needed for trigger during data taking)

# 6. Converting sqlite file tags

Using ~solodkov/scripts/calib\_to\_oracle (untested):

1. upd1='CheckTagAssociation.py --folder=/TILE/OFL02/CALIB/CIS/LIN --globaltag=CURRENTES | tail -1 | awk '{print $NF}''
2. upd4='CheckTagAssociation.py --folder=/TILE/OFL02/CALIB/CIS/LIN --globaltag=CURRENT | tail -1 | awk '{print $NF}''
3. AtlCoolCopy "sqlite://;schema=tileSqlite.db;dbname=CONDBR2" "sqlite://;schema=tileSqlite\_upd4.db;dbname=CONDBR2" -create -folder /TILE/OFL02/CALIB/CIS/LIN -tag $upd1 -outtag $upd4

In the case of CIS/LIN the full UPD1 and UPD4 tags are TileOfl02CalibCisLin-RUN2-HLT-UPD1-00 and TileOfl02CalibLasLin-RUN2-UPD4-16, but next year they'll probably be -RUN3- etc.