Updating L1 calibrations

New information (late 2022): Updating the L1 calibration database

Sometimes the instrument specialists give us new calibrations. Usually just a dead strip list for TKR, 3 (asym, MeV/DAC, peds) at a time for CAL. They need to be put in a standard place, registered in the calibration DB, and set active. This requires picking a time when validity switches from the previous set to the new one. We want this to happen between runs. The gaps between non-SAA runs are short, and CALDB doesn't do leap seconds, so I put the transitions near the middle of an SAA passage - the first one after the last run that we have data for.

If there is not already an existing issue for the update, start one in the JIRA PII tracker. Here is an example that can be used as a template: ⚠ PII-454 - Jira project doesn't exist or you don't have permission to view it. based off the ones they use for on-board hot strip masking updates (e.g., ⚠ OBCONF-204 - Jira project doesn't exist or you don't have permission to view it.).

We have adopted the workflow that one person (the primary) will do the update and another (the secondary) will check it and sign off after verifying that the start time is valid and the file names in the database are correct. They can either work on it together or have the secondary check later but before the calibration is scheduled to take effect. Hopefully, this will avoid situations in the past where updates were done incorrectly (e.g., the file names were mistyped) that resulted in having to cleanup the processing afterwards.

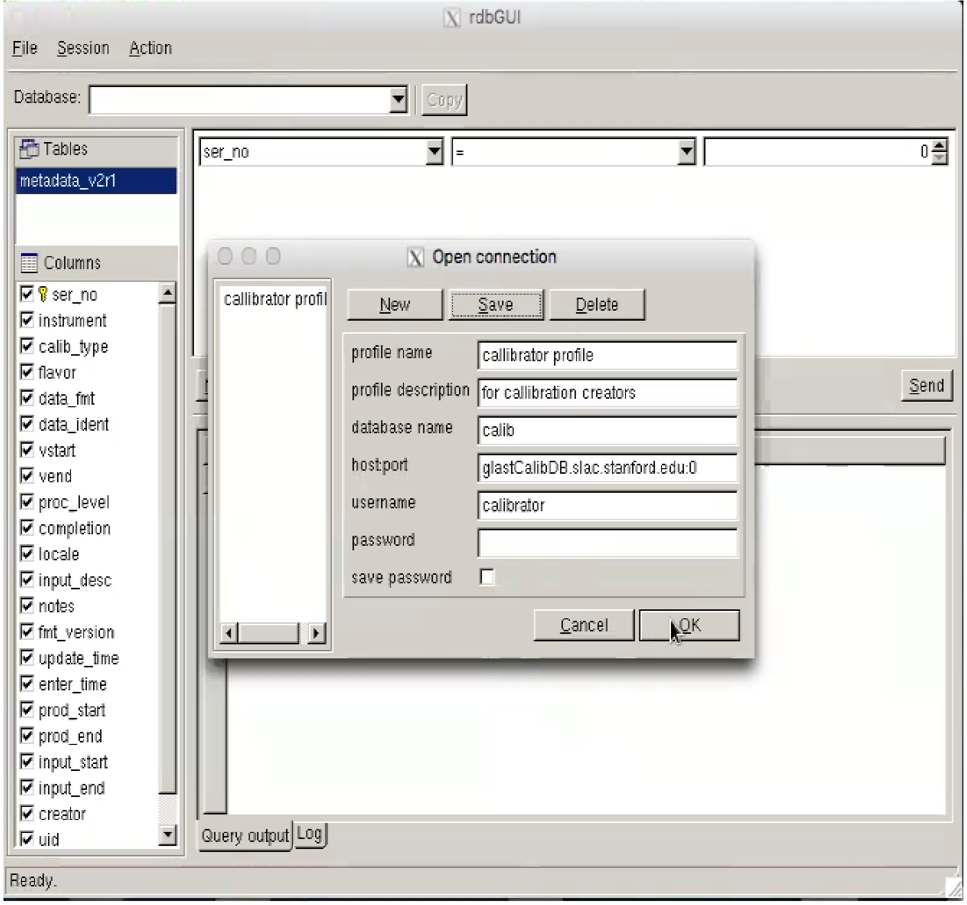
In general, it is probably better to do the updates early in the week rather than later.

Step-by-step guide

1. Make sure your environment is setup correctly:
    a. Log into a SLAC machine.
        i. If you are using ssh, you will need to have X11 forwarding enabled (the -X or -Y option if not enabled by default).
        ii. Using the necessary GUI via X11 can be very slow. SLAC has a remote desktop service called FastX that should work better. It has web and desktop clients. See their web page for setup and usage instructions.
            1. The GUI will not run on a FastX server. From a terminal session on a FastX server, you will need to "ssh -Y rhel6-64".
            2. From within your terminal session on RHEL6-64, you can quickly/easily check/confirm X-window GUI sharing with (for example) the "xeyes" or "xclock" commands.
            3. **NOTE: rdbGUI does not run on centos7**
        iii. The GUI does not run on fermilnx01 or fermilnx02. It will fail with a message like: *error while loading shared libraries*. If you get this, try running on a different machine. The rhel6-64 pool machines seem to work fine.
    b. If one is not already sourced in your .bashrc or .cshrc then:
        i. bash: `source /afs/slac.stanford.edu/g/glast/ground/scripts/group.sh`
        ii. (t)csh: `source /afs/slac.stanford.edu/g/glast/ground/scripts/group.cshrc`
    c. This will set environment variables like $GLASTROOT and the path to the GUI you will use below.
    d. Make sure that you have write permissions to the $LATCalibRoot directory.
        i. If not, ask Tom Glanzman.
2. Copy files to the "normal" place (if the instrument people haven't already):
    a. TKR: $LATCalibRoot/TKR
    b. CAL: $LATCalibRoot/CAL/p7repro
3. Figure out the transition time:
    a. Go to the data processing page.
        i. Find the most recent run. It will be at the top, unless deliveries have arrived out of order.
        ii. Click on the run number.
        iii. Note the start time.
    b. Go to the mission planning timeline
        i. Find the first SAA passage that is still in the future processing-wise.
        ii. Pick a time that is in the middle of the SAA passage. That is when we will start the new calibration.
        iii. You may also want to note the start time of the next run after the SAA. You will want to send that information to the data monitors list (see below) when you inform them of the new calibration.
4. Start the `rdbGUI`: type "rdbGUI" at the command prompt in your FastX terminal.
5. Go to the "File" menu and select "Open DB Schema".
6. Navigate to the directory `/afs/slac/g/glast/applications/dbSchemas/calib`
    a. You can paste this into the `File Name` box. **NOTE: to "paste" from clipboard into a Qt GUI window, using a Mac: ctrl-click-v**
    b. You can bookmark this location for future sessions by clicking on the small red flag icon, and retrieve the bookmark by clicking on the red flag icon.
7. Open the calib.xml file

    a. select the file "calib.xml" and click on "OK".
    b. then click on `metadata_v2r1` under the Tables box.

8. Under the `Session` menu, click on "Open connection". This will open a pop up box.
9. Fill in the box like the image below. The "profile name" and "profile description" boxes can be anything. The password is the same as the user name but with an "8" in place of the second "a". If you click "Save", then the connection information will be stored as a profile (e.g., "calibrator profile") that you can load next time from the box on the left instead of typing everything in.



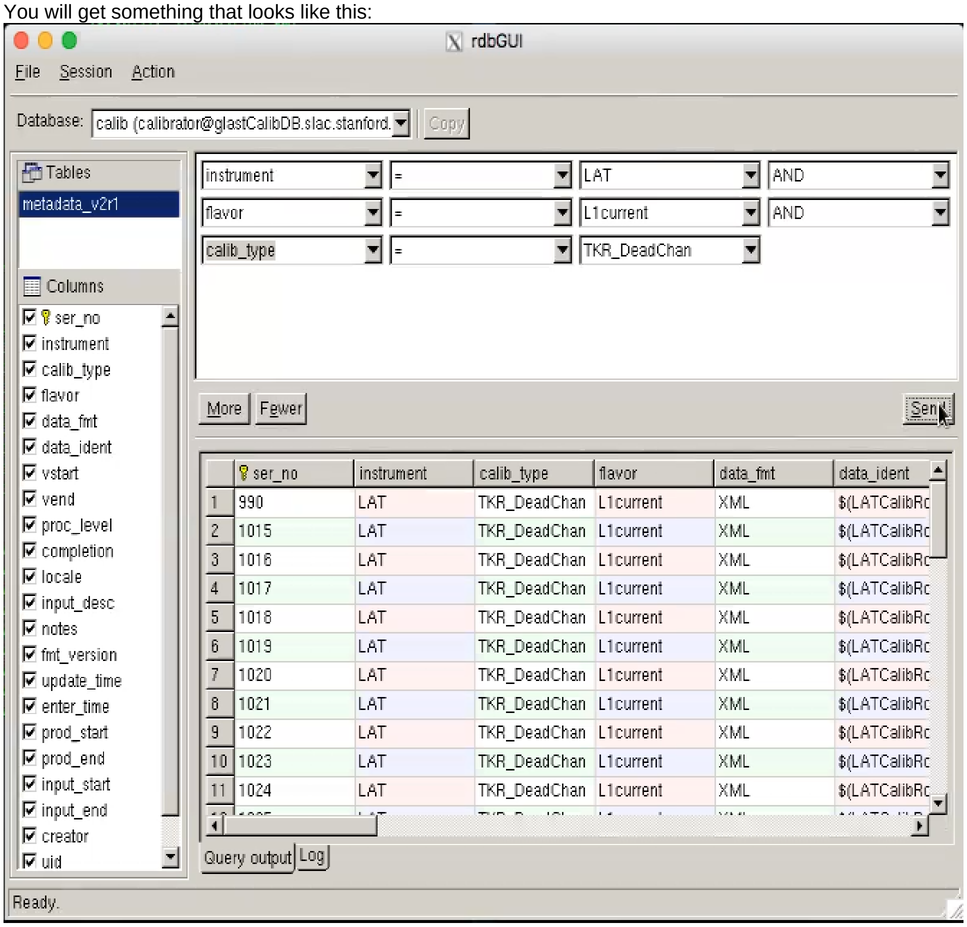
profile name	callibrator profile
profile description	for callibration creators
database name	calib
host:port	glastCalibDB.slac.stanford.edu:0
username	calibrator
password	
save password	□

10. Click "OK". Then verify that the chosen database "calib" appears in the Database window in the top left corner of the GUI.
11. Once connected, click on the drop down where is says "ser_no" (serial number) and set it to "instrument". It will fill in "LAT" in the box on the right.
12. Click on the "More" button at the bottom left corner of the upper box. This will add another drop down menu row.
13. Select "flavor" from the "ser_no" pull-down menu.
14. In the right hand box (that may say "vanilla"), type in "L1current". There is no menu choice for it.
15. Click on the "More" button again.
16. Select "calib_type" (calibration type) from the "ser_no" pull-down menu.
17. In the right hand box, select the type, e.g., 'TKR_DeadChan'.
    a. If you don't know the type that corresponds to the file you have, you can pick an option from the menu and hit "Send" to see what the current/previous calibration files are used for that type.
    b. For easier reference, here's a table with some example files and the calib_type for them.

file name example	**calib_type**	**Notes**
fit_gcrhists_lkhd_568m_572m_bigsum.gcr_asym_hist.xml	CAL_Asym	the "568m_572m" should increment by 6m for each new calibration
fit_proton_calib_568m_572m_bigsum.calMPD.xml	CAL_MevPerDac	the "568m_572m" should increment by 6m for each new calibration
pedavr_568m_572m.xml	CAL_Ped	the "568m_572m" should increment by 6m for each new calibration
LAT_BadStrips_45.xml	TKR_DeadChan	the "45" should increment by 1 for each new calibration

18. Then click the "Send" button at the bottom right corner of the upper box, i.e. on the right hand middle of the main rdbGUI window.

19. You will get something that looks like this:



Database: calib (calibrator@glastCalibDB.slac.stanford.

Tables: metadata_v2r1

Columns:

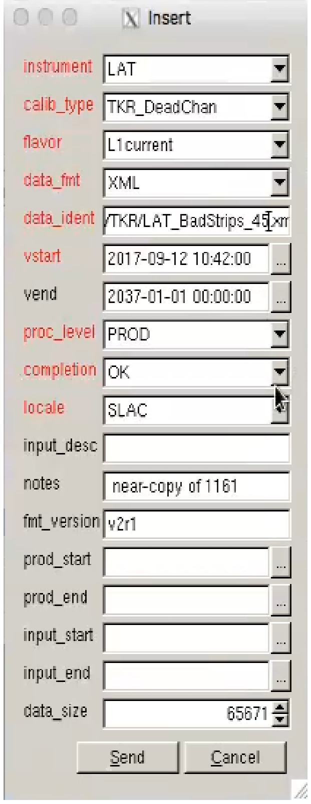
- ☑ ser_no
- ☑ instrument
- ☑ calib_type
- ☑ flavor
- ☑ data_fmt
- ☑ data_ident
- ☑ vstart
- ☑ vend
- ☑ proc_level
- ☑ completion
- ☑ locale
- ☑ input_desc
- ☑ notes
- ☑ fmt_version
- ☑ update_time
- ☑ enter_time
- ☑ prod_start
- ☑ prod_end
- ☑ input_start
- ☑ input_end
- ☑ creator
- ☑ uid

instrument	=	LAT	AND
flavor	=	L1current	AND
calib_type	=	TKR_DeadChan	

	ser_no	instrument	calib_type	flavor	data_fmt	data_ident
1	990	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
2	1015	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
3	1016	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
4	1017	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
5	1018	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
6	1019	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
7	1020	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
8	1021	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
9	1022	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
10	1023	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc
11	1024	LAT	TKR_DeadChan	L1current	XML	$(LATCalibRc

Ready.

20. Add the new calibration file.
    a. Select the last row in the bottom box.
    b. Right-click on a field (not the row number) in the last row and select "Copy Latest" from the menu. **NOTE: to right-click on a Mac laptop, do a 2-finger tap on the trackpad, or use a 3-button mouse**
    c. In the pop-up box that comes up, change the "vstart" time to the middle of the SAA passage chosen earlier.

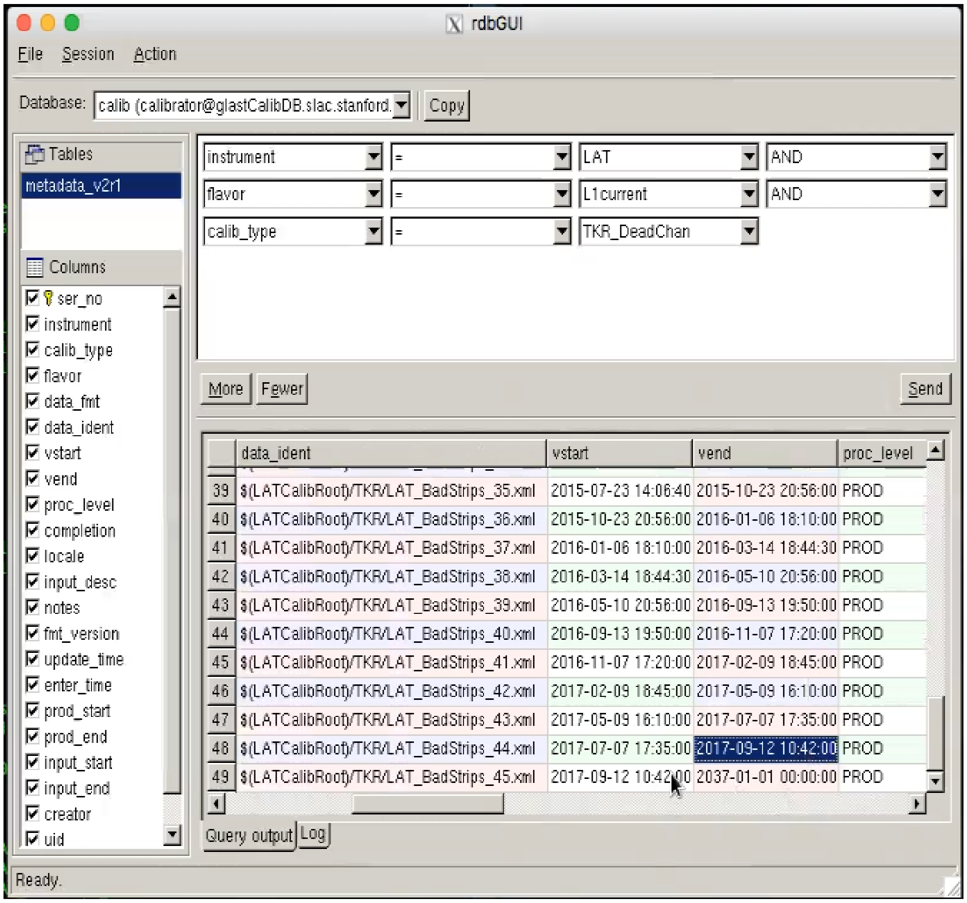


instrument	LAT
calib_type	TKR_DeadChan
flavor	L1current
data_fmt	XML
data_ident	/TKR/LAT_BadStrips_45.xm
vstart	2017-09-12 10:42:00
vend	2037-01-01 00:00:00
proc_level	PROD
completion	OK
locale	SLAC
input_desc	
notes	near-copy of 1161
fmt_version	v2r1
prod_start	
prod_end	
input_start	
input_end	
data_size	65671

    d. Change "data_ident" field to the file name for the XML file:

   e. Click "Send" on the tall and narrow "insert" pop-up and it will disappear.
   f. To show the new record in the calibration database that you have just added: re-click the "Send" button on the right hand middle of the main rdbGUI window.

21. There should be a new last line. Scroll over and check that the "vstart" time is correct and that the "vend" field of the previous line is the same.



Database: calib (calibrator@glastCalibDB.slac.stanford.

Tables: metadata_v2r1

instrument	=	LAT	AND
flavor	=	L1current	AND
calib_type	=	TKR_DeadChan	

Columns:
☑ ser_no
☑ instrument
☑ calib_type
☑ flavor
☑ data_fmt
☑ data_ident
☑ vstart
☑ vend
☑ proc_level
☑ completion
☑ locale
☑ input_desc
☑ notes
☑ fmt_version
☑ update_time
☑ enter_time
☑ prod_start
☑ prod_end
☑ input_start
☑ input_end
☑ creator
☑ uid

	data_ident	vstart	vend	proc_level
39	$(LATCalibRoot)/TKR/LAT_BadStrips_35.xml	2015-07-23 14:06:40	2015-10-23 20:56:00	PROD
40	$(LATCalibRoot)/TKR/LAT_BadStrips_36.xml	2015-10-23 20:56:00	2016-01-06 18:10:00	PROD
41	$(LATCalibRoot)/TKR/LAT_BadStrips_37.xml	2016-01-06 18:10:00	2016-03-14 18:44:30	PROD
42	$(LATCalibRoot)/TKR/LAT_BadStrips_38.xml	2016-03-14 18:44:30	2016-05-10 20:56:00	PROD
43	$(LATCalibRoot)/TKR/LAT_BadStrips_39.xml	2016-05-10 20:56:00	2016-09-13 19:50:00	PROD
44	$(LATCalibRoot)/TKR/LAT_BadStrips_40.xml	2016-09-13 19:50:00	2016-11-07 17:20:00	PROD
45	$(LATCalibRoot)/TKR/LAT_BadStrips_41.xml	2016-11-07 17:20:00	2017-02-09 18:45:00	PROD
46	$(LATCalibRoot)/TKR/LAT_BadStrips_42.xml	2017-02-09 18:45:00	2017-05-09 16:10:00	PROD
47	$(LATCalibRoot)/TKR/LAT_BadStrips_43.xml	2017-05-09 16:10:00	2017-07-07 17:35:00	PROD
48	$(LATCalibRoot)/TKR/LAT_BadStrips_44.xml	2017-07-07 17:35:00	2017-09-12 10:42:00	PROD
49	$(LATCalibRoot)/TKR/LAT_BadStrips_45.xml	2017-09-12 10:42:00	2037-01-01 00:00:00	PROD

22. Do this for each of the calibration files that need to be updated. You are basically done.
23. To quit from the rdbGUI, select "Quit" from the "File" pull-down menu at the top left of the GUI window.
24. Send an email to whoever created the calibration files and cc the data monitor list (datamonlist@glast4.stanford.edu) to let them know when the calibration should take effect.
    a. Get the ID of the next run by using XTime or similar tool to convert start time of the next run after the SAA pass into MET.
    b. Include the start time and first run in the email, e.g., "vstart is 2018-09-28 03:43. First run should be 559799706."
25. Update the JIRA issue with the same information.
26. **Later** check that the new calibration has been used.
    a. Go to data processing page and click on the L1Proc bar for the run after the change should have happened.
    b. Go to the "Substreams" table in the middle of the page and click on one of the streams for the "doChunk" task.
    c. Go down to the "Substreams" table and click on any one of the "doCrumb" streams.
    d. Click on the log file link for the "recon" process.
    e. Search in the log for appropriate XML file(s):
       i. you can search for the string "BadStrips" in the log file, to find the TKR bad strips XML file.
       ii. NOTE: the XML file(s) are selected by a command of the form:
       SELECT ser_no FROM metadata_v2r1 WHERE ((completion="OK") AND (instrument="LAT") AND (calib_type="TKR_DeadChan") AND (flavor="L1current")
       AND ("2021-05-22 12:57:20">=vstart) AND ("2021-05-22 12:57:20"<vend) AND ("PROD"=PROC_LEVEL)) ORDER BY update_time desc
       so in the event that the "vend" field of the previous record does not update, then the XML file might be selected by the "update_time" value. So you might want to confirm that the desired new XML calibration file has the latest "update_time".
27. Assuming everything is good, update and close the JIRA issue.

Troubleshooting

This section is to help troubleshoot any errors.

L1Proc failing

If the L1Proc deliveries start to fail after the calibration change takes effect, look at the logs mentioned above and search for any errors.   If you see something like:

> *XmlBaseCnv FATAL Unable to parse document $(LATCalibRoot)/CAL/p7repro/fit_proton_calib_634m_638m_bigsum.calMPD.xml*
> *aka /afs/slac/g/glast/ground/releases/calibrations//CAL/p7repro/fit_proton_calib_634m_638m_bigsum.calMPD.xml*

then that calibration file was not actually copied to the TKR or CAL directories described above. Copy it there and then either contact someone to rollback the process or do it yourself following the instructions in Things to know while on-call for Data Processing. In this case, a command line rollback of delivery 210427007:

> */afs/slac.stanford.edu/u/gl/glast/pipeline-II/prod/pipeline -m PROD rollbackStream --minimum 'L1Proc[210427007]'*

was all that was needed to successfully process the delivery.

Duplicate rows

During one update, step 20 was done twice creating two rows in the database, and the vend value was not updated for the "old" calibration file. The calibrator uses does not have permission to delete rows. What you can do is edit the value in the table to manually set either using the rdbGUI (like in step 20) or logging into the database on the command line.

Here's what was done on the command line. First, log into the mysql database. I think this works from pretty much any SLAC machine. The password is the same as the one used for rdbGUI given above:

> *mysql -h glastCalibDB.slac.stanford.edu -u calibrator -p*

Then change to the calib database and look at the table. Below are the last few entries for the TKR calibration files:

```
MySQL [(none)]> use calib;
MySQL [calib]> select ser_no,flavor,data_ident,vstart,vend,update_time from metadata_v2r1 where calib_type =
'TKR_DeadChan' and ser_no > 1256;
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------+
| ser_no | flavor    | data_ident                               | vstart              | vend                |
update_time         |
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------+
|   1263 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_58.xml | 2020-08-03 15:30:00 | 2020-11-05 04:23:00 |
2020-11-05 02:28:20 |
|   1267 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_59.xml | 2020-11-05 04:23:00 | 2021-02-03 15:41:00 |
2021-02-03 20:28:28 |
|   1271 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_60.xml | 2021-02-03 15:41:00 | 2037-01-01 00:00:00 |
2021-02-03 20:28:28 |
|   1278 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_61.xml | 2021-05-04 18:29:00 | 2037-01-01 00:00:00 |
2021-05-04 22:09:54 |
|   1279 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_61.xml | 2021-05-04 18:30:00 | 2037-01-01 00:00:00 |
2021-05-04 21:45:51 |
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------
```

I then updated the vend times the "old" one and the initial duplicate. For good measure, I also changed the flavor of the 1278 row to "test" from "L1current" since the pipeline selects only "L1current" files (see the note in step 25e).

```
MySQL [calib]> update metadata_v2r1 set vend = '2021-05-04 18:30:00' where ser_no = 1271;
MySQL [calib]> update metadata_v2r1 set vend = '2021-05-04 18:30:00' where ser_no = 1278;
MySQL [calib]> update metadata_v2r1 set flavor='test' where ser_no = 1278;
MySQL [calib]> select ser_no,flavor,data_ident,vstart,vend,update_time from metadata_v2r1 where calib_type =
'TKR_DeadChan' and ser_no > 1256;
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------+
| ser_no | flavor    | data_ident                               | vstart              | vend                |
update_time         |
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------+
|   1263 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_58.xml | 2020-08-03 15:30:00 | 2020-11-05 04:23:00 |
2020-11-05 02:28:20 |
|   1267 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_59.xml | 2020-11-05 04:23:00 | 2021-02-03 15:41:00 |
2021-02-03 20:28:28 |
|   1271 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_60.xml | 2021-02-03 15:41:00 | 2021-05-04 18:30:00 |
2021-05-25 07:00:51 |
|   1278 | test      | $(LATCalibRoot)/TKR/LAT_BadStrips_61.xml | 2021-05-04 18:29:00 | 2021-05-04 18:30:00 |
2021-05-25 07:01:55 |
|   1279 | L1current | $(LATCalibRoot)/TKR/LAT_BadStrips_61.xml | 2021-05-04 18:30:00 | 2037-01-01 00:00:00 |
2021-05-04 21:45:51 |
+--------+-----------+------------------------------------------+---------------------+---------------------
+---------------------
```

A later check of the pipeline confirmed that it is using the 1279 entry.

Further troubleshooting instructions will be added as issues come up.