

# Automatic run restarts



+



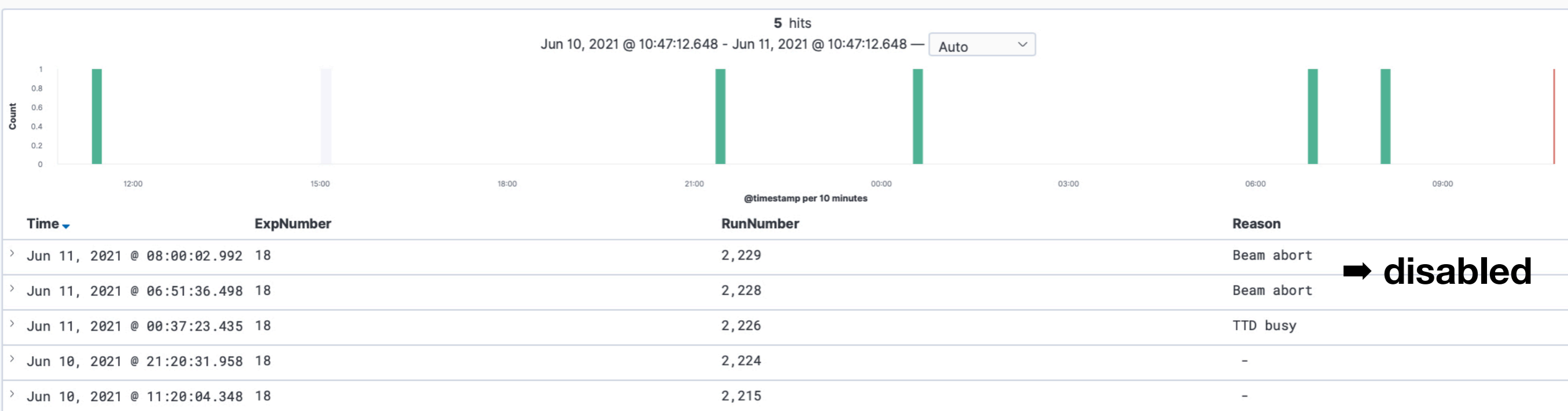
**Takuto KUNIGO**

June 11, 2021  
OTF meeting



# Activated

Finally the auto run-restart mode activated!



However, further tuning and debugging needed

# Automatic run-restarts

## Current scheme to fix problems on our data-taking:

➔ Try SALS as soon as possible if SALS cannot fix the problem, contact the relevant on-call experts

- Sound alarm helps the CR shifters to take prompt action
- Automatic run restarts

## Categories

### 1. Automatic restart for “inevitable” cases

- e.g. in case of the 8hour limit
- Automatic run stop at beam aborts, and/or automatic run start when HV permission granted can be categorised in this category

### 2. 1st trial of SALS

- i.e. 1st trial of SALS is requested/required most of the cases, we may trigger 1st trial of SALS automatically.
- However, we should not perform SALS against some errors which are not recoverable by SALS

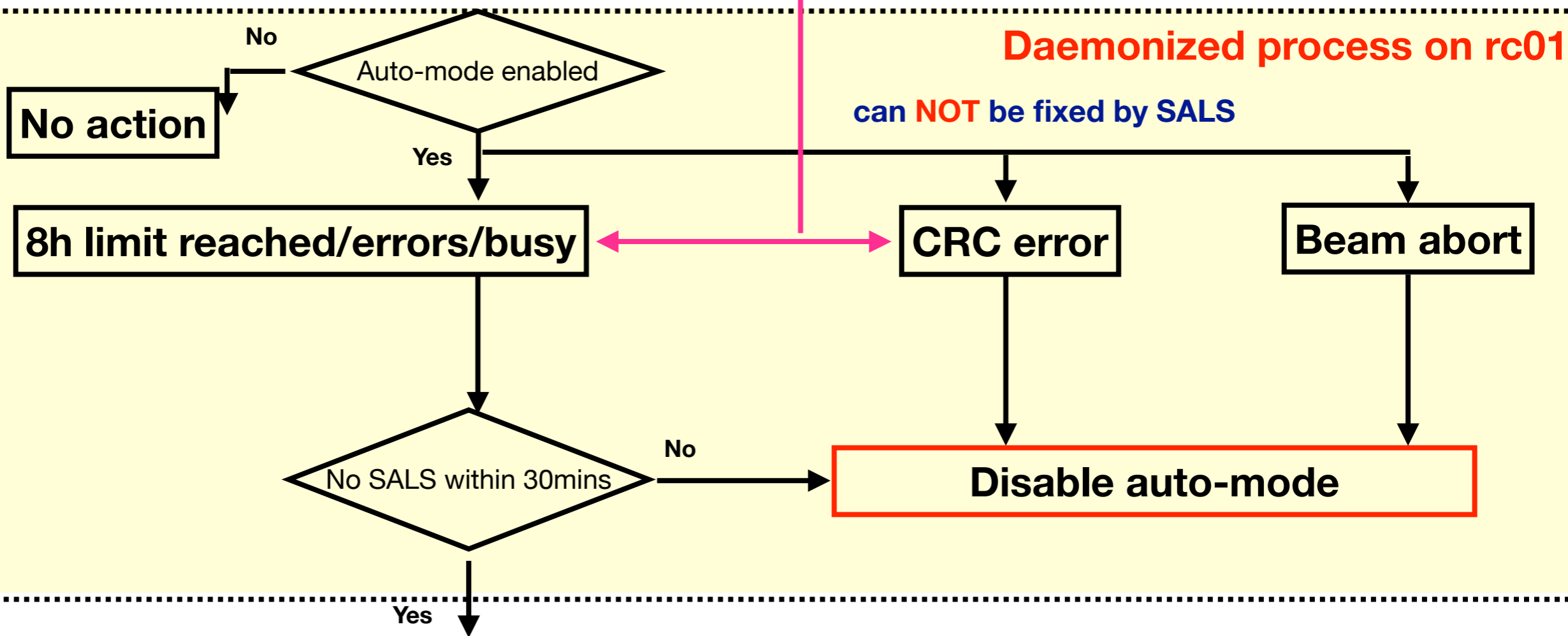
# Implementation/Flowchart

## Needed info

- Log messages
- PC metrics
- EPICS PVs
- nsm2 values
- Beam abort log

**Checker**

- Monitoring processes
- e.g Elasticsearch + Elastalert



**Automatic SALS/ALS/SS**

# From the CR shifters...

When auto mode is enabled, STOP/ABORT/LOAD/START buttons are un-clickable

The screenshot displays the control interface for the CR shifters. Key elements include:

- RC Command:** STOP, ABORT, and **AUTO MODE OFF** (highlighted with a red box and arrow).
- Run status:** Exp #: 18, Run #: 2211, RUNNING (highlighted).
- Run control:** RUNNING (highlighted).
- TTD Status:** RUNNING (highlighted).
- Data flow:** RUNNING (highlighted).
- Detector states (ABORT before you check or uncheck a subsystem):**
  - PXD: OFF
  - SVD: OFF
  - CDC: RUNNING
  - TOP: OFF
  - ARICH: RUNNING
  - ECL: RUNNING
  - KLM: RUNNING
  - TRG: RUNNING
  - HLT: RUNNING
- Run setting:** Run type: null, Trigger type: poisson, Dummy rate: 30000, HLT script: passthrough.
- Trigger / Data status:**
  - Trigger input: 308953401, Rate: 30.45 kHz
  - Trigger output: 278094863, Rate: 27.41 kHz
  - Run start: 2021-06-10 06:51
- HLT Data Table:**

	HLT01	HLT02	HLT03	HLT04	HLT05	HLT06	HLT07	HLT08	HLT09	HLT10
# events :	27797849	27800836	27800762	27797965	27800748	27800748	27798024	27801027	27801074	27798248
Rate :	2.8 kHz	2.9 kHz	2.8 kHz	2.9 kHz	2.7 kHz	2.7 kHz	2.9 kHz	2.9 kHz	2.9 kHz	3.0 kHz
Flow :	137 MB/s	138 MB/s	141 MB/s	139 MB/s	133 MB/s	131 MB/s	146 MB/s	139 MB/s	142 MB/s	153 MB/s
- Subsystem Status Grid:**
  - PXD:** Run # : 2210. Status: NOTREADY.
  - SVD:** Run # : 2210. Status: NOTREADY.
  - CDC:** Run # : 2211. Status: RUNNING.
  - TOP:** Run # : 2210. Status: NOTREADY.
  - ARICH:** Run # : 2211. Status: RUNNING.
  - KLM:** Run # : 2211. Status: RUNNING.
  - TRG:** Run # : 2211. Status: RUNNING.
  - ECL:** Run # : 2211. Status: RUNNING.
  - HLT:** Run # : 2211. Status: RUNNING.

B2\_alarm:AUTORESTART:enable

## Auto mode is enabled

- ▶ No sound alert
- ▶ RocketChat automatic message for notification + “Stop reason” in the elog entries filled (automatically) for logging

# SALS and SS functions

- SALS and SS functions defined
- confirmed that these functions work fine

```
1 #!/usr/bin/env python3
2 import time
3 import datetime
4 import epics
5 import nsm2
6
7 def SALS(timeout_start, timeout_load, timeout_abort, timeout_stop):
8     nsm2.init('AUTO_RESTARTS', 9020)
9
10    nsm2.send('RUNCONTROL', 'RC_STOP')
11    time_send = datetime.datetime.now()
12    while True:
13        state = nsm2.vget('RUNCONTROL', 'rcstate')
14        time.sleep(1) # remove?
15        duration = (datetime.datetime.now() - time_send).total_seconds()
16        if duration > timeout_stop:
17            raise Exception('stuck at STOPPING')
18        if state == 'READY':
19            break
20        elif state == "ERROR" or state == "FATAL":
21            raise Exception('RC state is ERROR or FATAL')
22
23    nsm2.send('RUNCONTROL', 'RC_ABORT')
24    time_send = datetime.datetime.now()
25    while True:
26        state = nsm2.vget('RUNCONTROL', 'rcstate')
27        time.sleep(1) # remove?
28        duration = (datetime.datetime.now() - time_send).total_seconds()
29        if duration > timeout_abort:
30            raise Exception('stuck at ABORTING')
31        if state == 'NOTREADY':
32            break
33        elif state == "ERROR" or state == "FATAL":
34            raise Exception('RC state is ERROR or FATAL')
```

# Unrecoverable errors

```

14 Timeout:
15     START: 300
16     LOAD: 300
17     ABORT: 300
18     STOP: 300
19 UnrecoverableErrors:
20     - '/home/group/b2daq/tkunigo/elk-input-scripts-temp/elastalert/rules/DAQ/crc_errors.yaml'

```

**This list need to be updated  
At least, I'll add the errors covered by the recover GUI**

```

# If error or busy
elif (rc_state=="ERROR") or (rc_state=="FATAL") or (error>0.0) or (busy>0.0):
    list_SALS = check_sliding_window()
    n_SALS    = len(list_SALS)

```

# Unrecoverable errors

```

UnrecoverableErrors = config_data['UnrecoverableErrors']
unrecoverable = False
for this_alert_rule in UnrecoverableErrors:
    if elasticsearch_query.query(this_alert_rule):
        unrecoverable = True
        break
if unrecoverable:
    # disable the auto-mode when unrecoverable errors detected
    epics.caput(config_data['AutoRestartFlag'], 0.0)

```

```

elif n_SALS<1:
    try:
        SALS(timeout_stop=config_data['Timeout']['STOP'], timeout_abort=config_data['Timeout']
    except Exception as e:

```

# For future

## Recoverable errors will be fixed by auto-SALS

## “Unrecoverable errors” can be solved by the recovery GUI ➔ switch to automatic recovery

The screenshot shows the RCControlMain GUI with the following sections:

- RC Command:** STOP, ABORT, AUTO MODE ON
- Run status:** Exp #: 18, Run #: 2231, RUNNING (RUNNING, RUNNING, RUNNING)
- Detector states (ABORT before you check or uncheck a subsystem):**
  - PXD: RUNNING
  - SVD: RUNNING
  - CDC: RUNNING
  - TOP: RUNNING
  - ARICH: RUNNING
  - ECL: RUNNING
  - KLM: RUNNING
  - TRG: RUNNING
  - HLT: RUNNING
- Run setting:** Run type: physics, Trigger type: gdl, Dummy rate: 30000, HLT script: beam\_reco\_filter
- Trigger / Data status:**

Trigger input	# events	Rate	Trigger output	# events	Rate	Run start:
18351251	18351251	2.190 kHz	18340846	18340846	2.188 kHz	2021-06-11 08:34

	HLT01	HLT02	HLT03	HLT04	HLT05	HLT06	HLT07	HLT08	HLT09	HLT10
# events:	1833923	1833694	1833954	1833718	1833984	1833949	1833954	1833745	1833754	1833948
Rate:	227.0 Hz	225.0 Hz	248.0 Hz	252.0 Hz	231.0 Hz	201.0 Hz	215.0 Hz	243.0 Hz	233.0 Hz	198.0 Hz
Flow:	7.6 MB/s	6.6 MB/s	8.8 MB/s	7.0 MB/s	6.8 MB/s	5.6 MB/s	7.7 MB/s	6.9 MB/s	7.9 MB/s	5.8 MB/s
- Subsystem Controls:** PXD, SVD, CDC, TOP, ARICH, KLM, TRG, ECL, each with individual status indicators and control buttons.



The screenshot shows the CDC Alarms GUI with the following details:

- CDC Alarms:** RC state: RUNNING
- FEE reprogram:** Status: ACTIVATED, FEE ID: 22-23, FEE bitfile: /bdaq/group/b2cdc/bit/trg\_recbe\_v61b.bit
- Buttons:** Reprogram FEE

## TOP and ARICH are joining



## (Pseudo) automatic data-taking

## This is not “Pie in the sky”



# Backup

# Demonstration

- **Git repo, [link](#)**  
- **config.yaml**

Controlled by supervisor b2daq@rc01

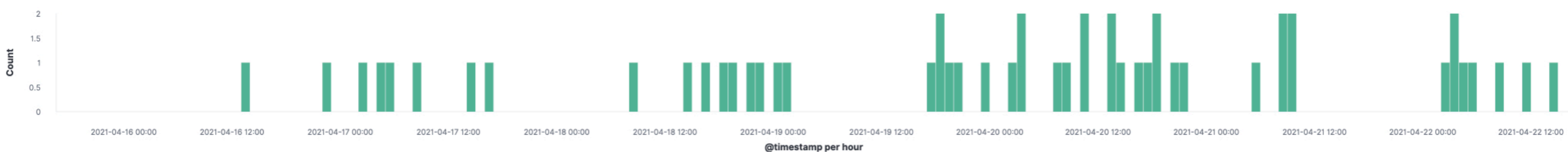
\$ ~/.local/bin/supervisorctl [start/stop/restart] auto\_restartd

```

1 RCState: 'DAQ:RC:State'
2 RunType: 'DAQ:RC:RunType'
3 ExpNumber: 'DAQ:RC:ExpNumber'
4 RunNumber: 'DAQ:RC:RunNumber'
5 HVState: 'B2_nsm:get:HVMaster:hvstate'
6 HVPermission: 'B2_MDI:HV_PERMISSION'
7 AutoRestartFlag: 'test:B2_alarm:AUTORESTART:enable'
8 SKB:
9   Abort: 'B2_alarm:beam_abort'
10 TTD:
11   tStart: 'DAQ:TTD:tStart'
12   ERROR: 'B2_alarm:TTD:ftstate:ERROR'
13   BUSY: 'B2_alarm:TTD:ftstate:BUSY'
14   NotRecoverableErrors:
15     CRC errors: 'B2_alarm:DAQ:CRCErrors'
```

**To be updated**

**Properly detected the events at which we should try automatic run restarts**



Time	ExpNumber	RunNumber
> Apr 22, 2021 @ 14:09:19.111	18	570
> Apr 22, 2021 @ 11:14:45.081	18	561
> Apr 22, 2021 @ 08:59:08.426	18	556
> Apr 22, 2021 @ 05:36:26	18	554
> Apr 22, 2021 @ 04:36:49.515	18	553
> Apr 22, 2021 @ 03:48:10.917	18	550
> Apr 22, 2021 @ 03:18:09.821	18	550
> Apr 22, 2021 @ 03:18:09.821	18	550