

# README - Nikiroot 3.5

James Smallcombe 22/12/2017 - james.smallcombe@outlook.com

---

Ensure root is sourced before running "make"

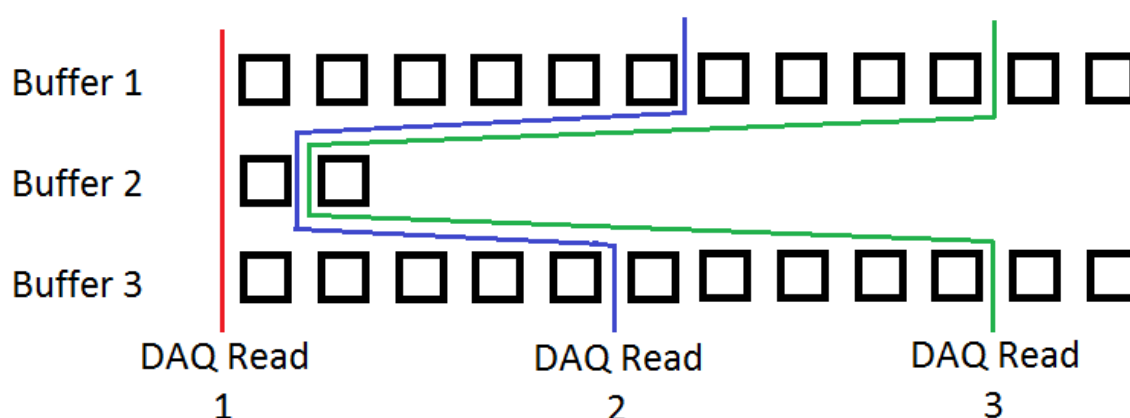
---

## Buffers and Time correction changes

With the current version some simplifications have been made and a new approach to time errors and buffer reads has been taken.

Within the .dat files the clear structure of the hardware DAQ reads can be seen, with event blocks corresponding to ascending card number. As each DAQ read corresponds approximately to a physical chunk of time during good operational running, these blocks can be used to separate out when reads from disks are needed and to keep data in individuals correlated without relying on the often faulty time stamp.

Between each daq read block a flag (incrementing negative number) is inserted into the data buffers so the relative position in absolute time can be tracked.



This has 2 major benefits:

Firstly when a low statistics channel is empty of data during event construction (while high statistics channels still contain a lot of data) additional data will not be read immediately. Data will only be read to buffers when there is no data in any trigger channel OR when a trigger channel has been established and an empty channel is within a user specified number of DAQ reads.

Secondly when a timestamp error is found, rather than discarding data in a chaotic manor, data is discarded before the corresponding DAQ read and an additional user specified number of reads.

The "safe" number of DAQ reads can be set very low (1 or 2) but one should first check the real world time one daq read corresponds to in a given experiment. With a sufficiently high data rate in a few channels a single physical event could conceivably be split over several daq reads. The scan.C root marco is provided to perform such a check.

As a result of this change the buffer size is now set in DAQ reads rather than absolute size and should have negligible effect on event construction, tune to your system, too low will be slow due to disk reads, too high will fill your memory.

## Event Construction

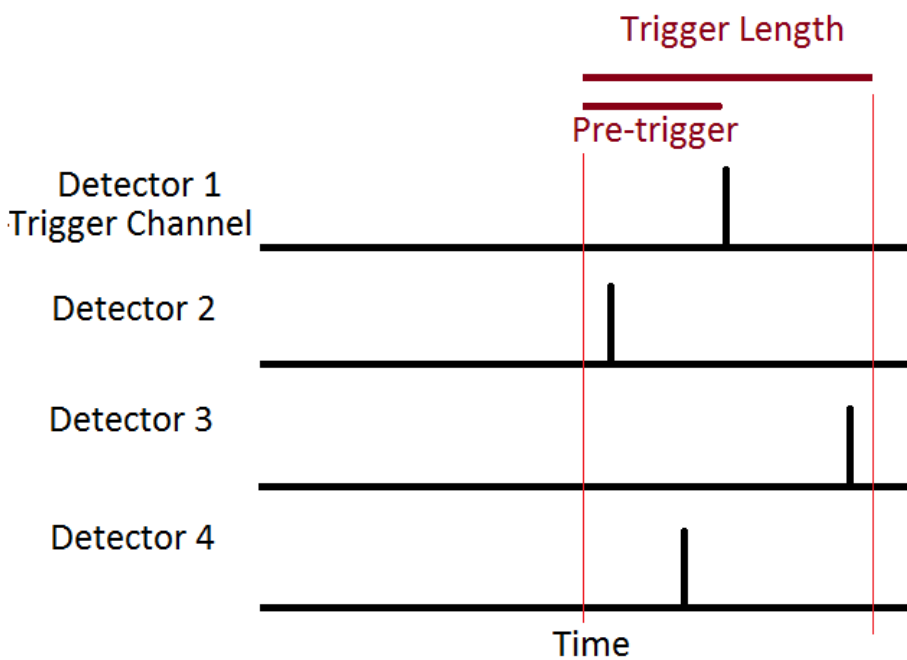
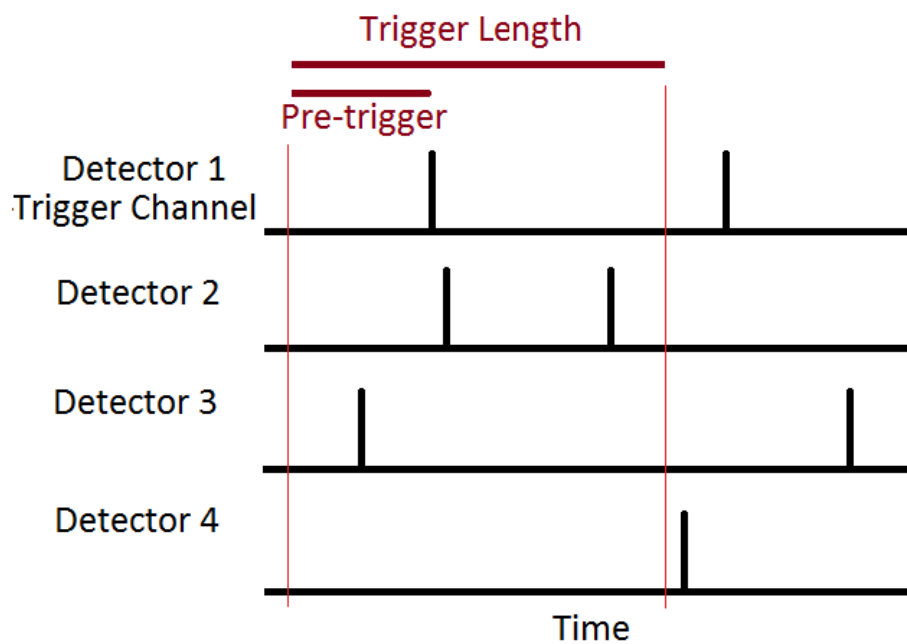
Event construction has had some minor changes made but works mostly the same. The major change is in the handling of pile up.

The earliest occurring trigger channel defines the "trigger time" for each single event construction.

The trigger time minus the "Pre Trigger" defines the event start.

The first hit of each channel occurring within "Trigger Length" elapsed ticks following the event start are collected into the current event.

Any ADDITIONAL hits occurring in a channel within the Trigger Length REMAIN to be triggers and data in subsequent events, these are no longer lost and "pile up" is no longer counted.



## Offset Time

Every channel can be given an offset time (in daq ticks) so that channels with large delays can be more easily correlated with their counterpart events.

This offset is applied when the time is read from the buffer for triggering or event construction.

If the delay is large or the rate in channels high these delayed channels from the same event could be on separate daq reads, so be sure to increase "Buffer DAQ reads safe" with the -B# flag.

## Commands

nikiwrapped takes the following command line options, all are optional. Example files are provided.

--FILE Input file

FILE should be an ascii file (with relative or absolute path).

File should contain a whitespace delimited list of input arguments.

These will be loaded sequentially as if given on the command line.

e.g. "nikiwrapped --LIST\_OF\_COMMANDS.txt"

-r# Report Level

# should be a number 0-4

0=Silent running

1=Basic file reporting DEFAULT

2=Buffer reporting

3=Buffer details

e.g. "nikiwrapped -r3"

## Trigger

-t# Trigger Length

# should be an integer

This is the event size in DAQ time stamps, default is 200.

e.g. "nikiwrapped -t100"

-p# Pre Trigger (trigger delay)

# should be an integer

This is the delay between the start of an event and the trigger channel in DAQ time stamps, default is 0.

e.g. "nikiwrapped -p50"

-D#,# DAQ triggers

# should be an integer

Input the DAQ channels that should be used as triggers.

DAQ channel numbers are given by  $N = \text{module\#} * 16 + \text{channel\#}$

-D# can be used multiple times or a comma delimited list or a combination

e.g. "nikiwrapped -D3 -D56 -D12,14,15"

## Data

-b# Buffer size

# should be an integer

Size of the read in buffer, default 100.

Larger values may result in memory problems.

If set too small in many channel systems data for very low frequency channels may be lost.

Use option -r3 to monitor buffers.

e.g. "nikiwrapped -b100"

-B#      Buffer DAQ reads safe

# should be an integer

The number of DAQ reads considered "safe", default is 5.

The outer extremes of event construction or the resetting of time breaks.

Set low for minimum data waste, high for maximum safety.

e.g. "nikiwrapped -B5"

-T      Timing Data

Turns on timing branches of trees, redundant in current trigger mode, default is off

e.g. "nikiwrapped -T"

-cFILE   Channel file

FILE should be an ascii file with relative or absolute path.

Gives names of active channels, should be a file containing a

whitespace delimited list of channel numbers followed by a name and a time offset in ticks. e.g. 66 MWPC1\_y 0

DAQ channel numbers are given by  $N = \text{module\#} * 16 + \text{channel\#}$

All channels must have an offset time, enter 0 if not desired

NOTE If left blank channellist.txt will be generated (and overwritten) in the current directory using the data.

e.g. "nikiwrapped -cLIST.txt"

## Files

-IPATH   Load path or list

PATH should contain either the path to the .dat files to be proceed

OR a .txt file containing whitespace delimited list of files to process.

Files should be names should begin in the local convention RUN###\_

e.g. "nikiwrapped -l/data/experiments/z90z90/"

e.g. "nikiwrapped -lFILES.txt"

-sPATH   Save path

PATH should contain the save path.

If the folder does not exist it will be created.

e.g. "nikiwrapped -s/data/experiments/z90z90\_presorted/"

-o      Overwrite

This must be included to overwrite output files.

If omitted one can point to a folder of data of which some runs have been sorted and these runs may be skipped.

---

Alternatively code may be run natively in root, a macro is provided to load the subroutines.