**Bin Tracking**

**Objective:**

A tool for service or engineering personnel to evaluate bin tracking performance using log files.

**Features**:

The application allows user to select various ways to open log file: Menu bar selections (open file/files,

open folder)

There are four section on main screen: Summary, Unlinked Details, Charts, File lists

* Summary contains:
  + Entrance events:
  + Entrance misreads:
  + Read last RFID (found previous bag):
  + Total Bins:
  + Total Linked Bins:
  + Total Unlinked Bins:
  + Attenuation events:
  + Total Linked Internal Bags:
  + Total Unlinked Internal Bags:
  + Exit events:
  + Exit misreads:
  + Lost Tracking Crusty Bags:
  + Bag already sent to next zone:
  + Loose Bags: (Exit misreads – Lost Track + Bag already sent)
  + Total Exit Bins: (Exit events – loose bags)
  + Percent of Unlinked Bins:
  + Percent of Unlinked Internal Bags:
* Unlinked details:
  + Slipping Bins: , Percentage (unlinked): %
  + Pushed Bins: , Percentage (unlinked): %
  + Unlinked Bin Read at Exit:
  + Unlinked Bin Lost Track before Exit:
* Charts:
  + - Linked Bins Histogram (see Figure 1,2)
      * X axis labeled as “Attenuation Window”
      * Y axis labeled as “Frequency”
    - Unlinked Bins Donut Chart (see Figure 3,4)
      * Pushed by operator (-1 to -50)
      * *Linking bins (0 to 150) not show in this histogram*
      * Slipping on belt (151 to 200)
      * Others
    - Summary Pie Chart (see Figure 5,6)
      * Linked Bins /Unlinked Bins show in pie chart
* File lists
  + List of file/files/folders that is currently execute

*\*Error handling when data is not ready (1. Show empty chart; 2. Show “data not ready” message.)*

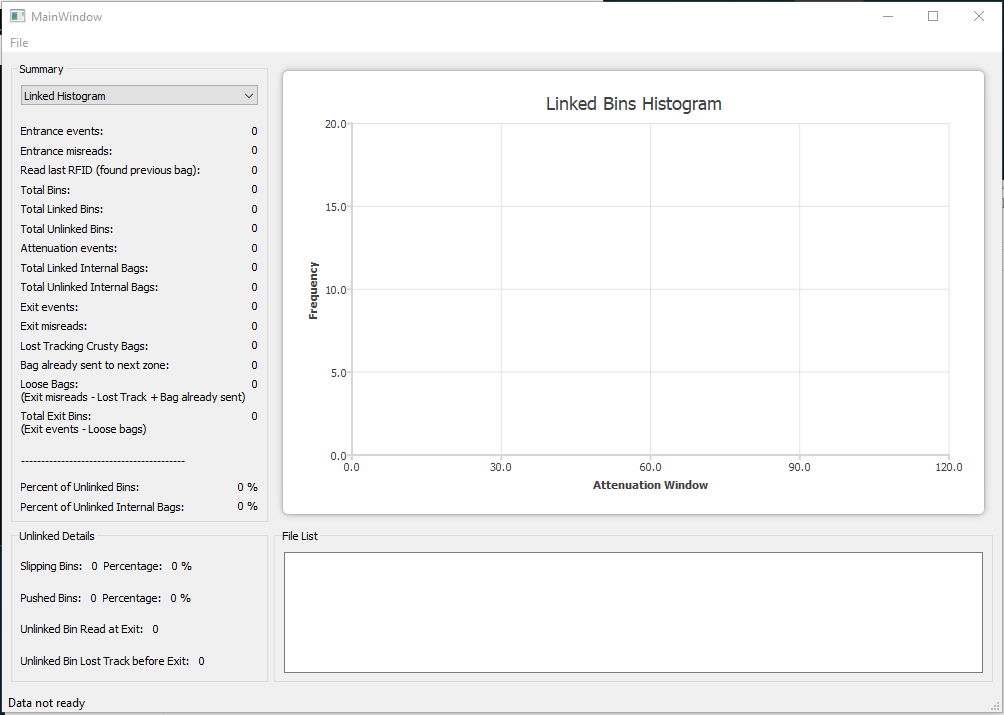


Figure 1. Linked Bins Histogram without data

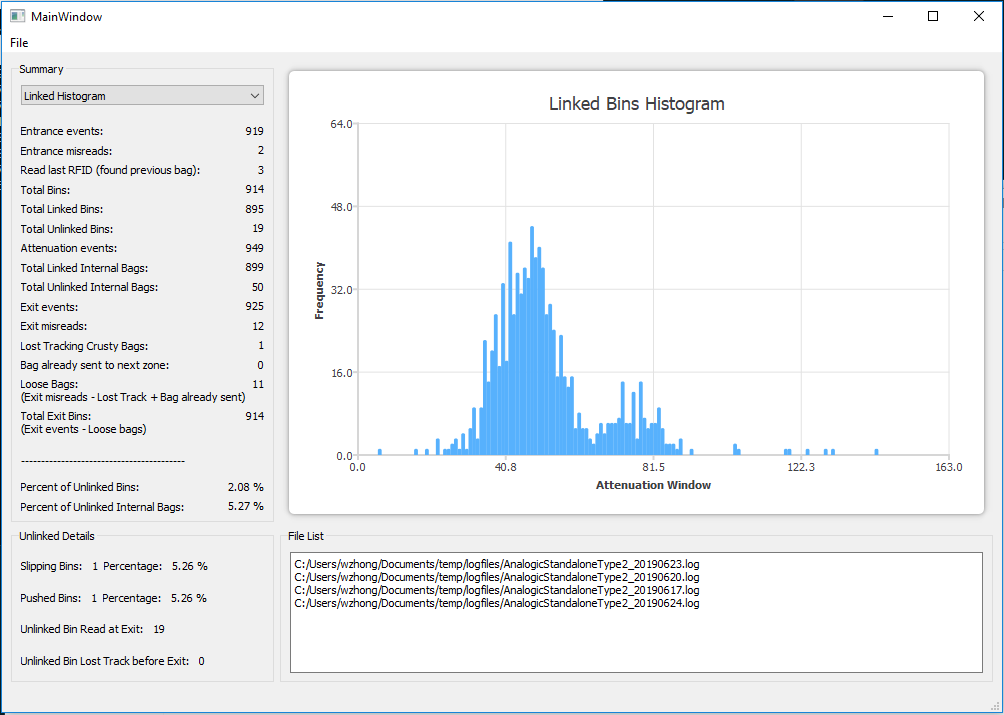


Figure 2. Linked Bins Histogram with data

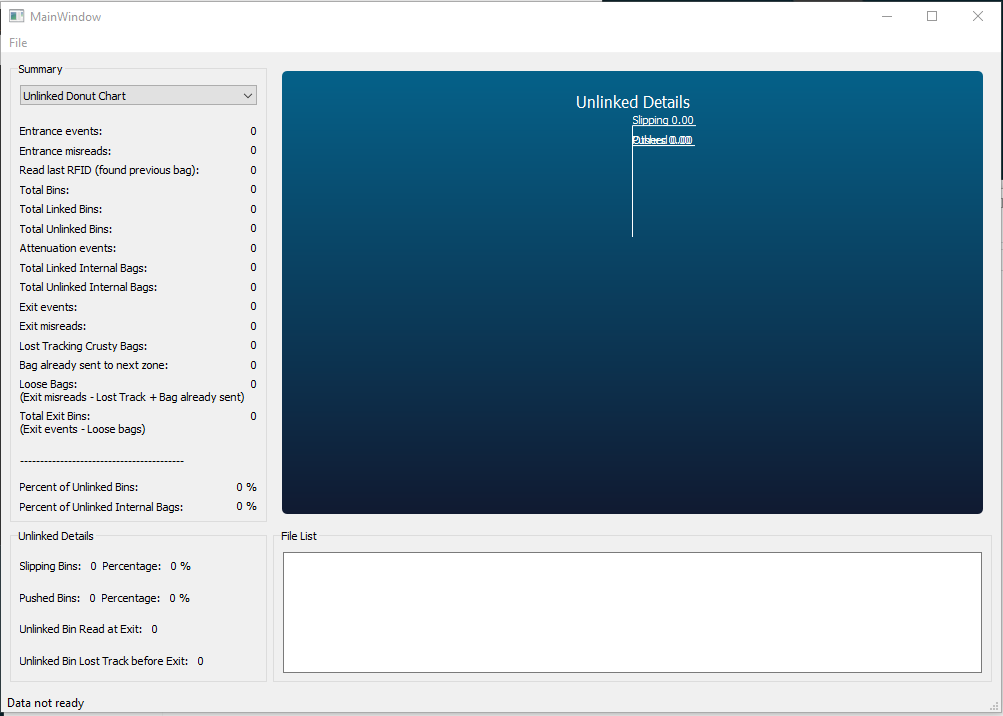


Figure 3. Unlinked Bins Detail Donut Chart without data

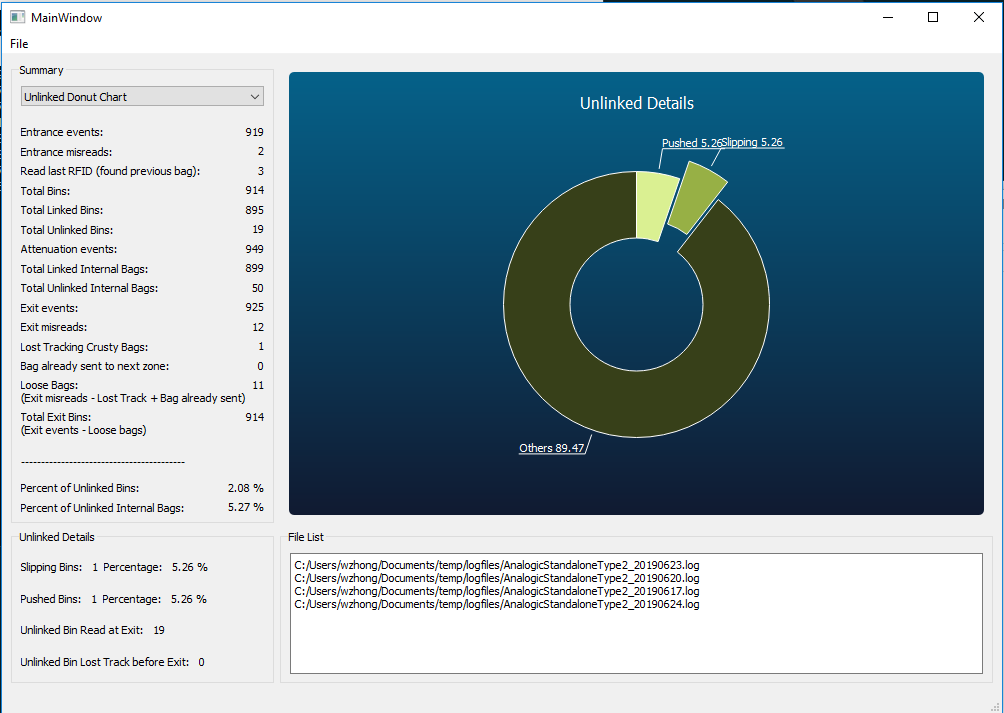


Figure 4. Unlinked Bins Donut Chart with data

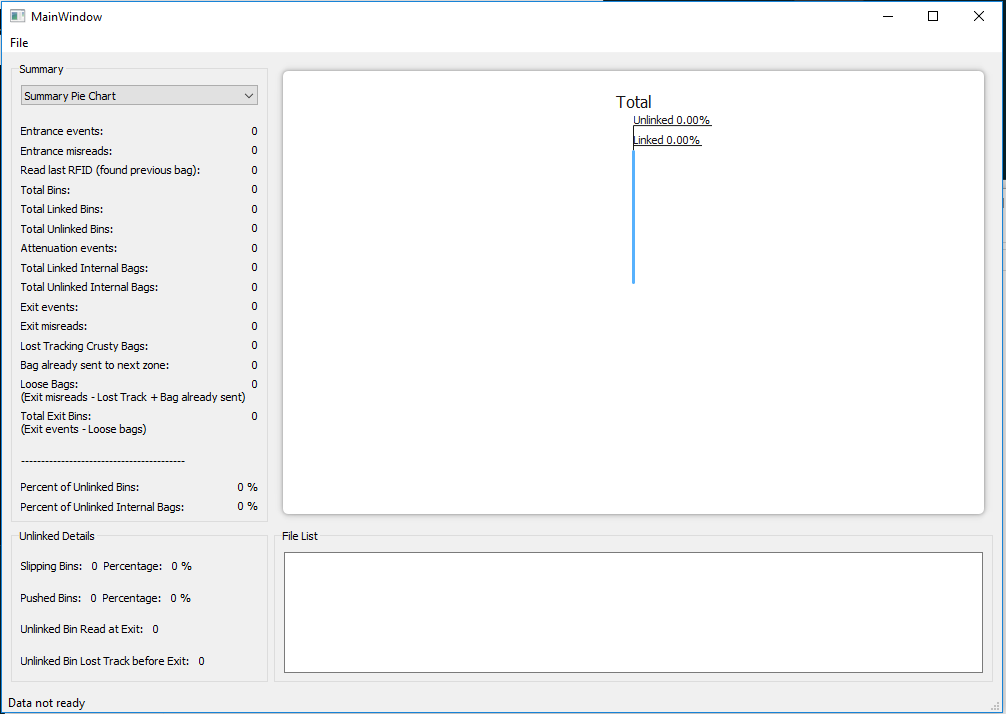


Figure 5. Summary Pie Chart without data

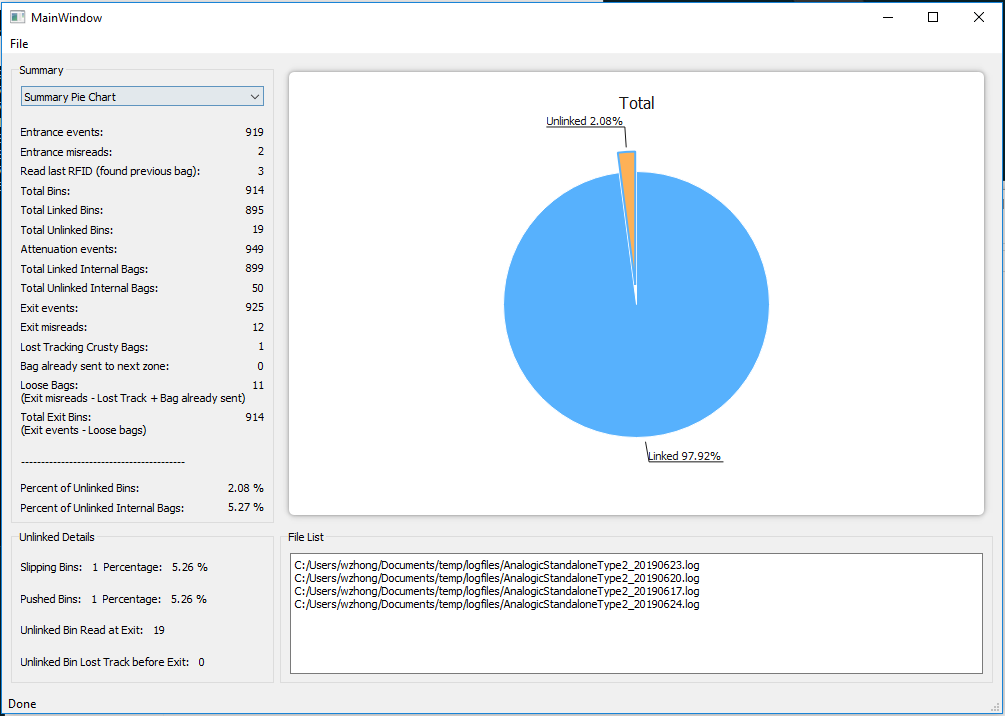


Figure 6. Summary pie chart with data

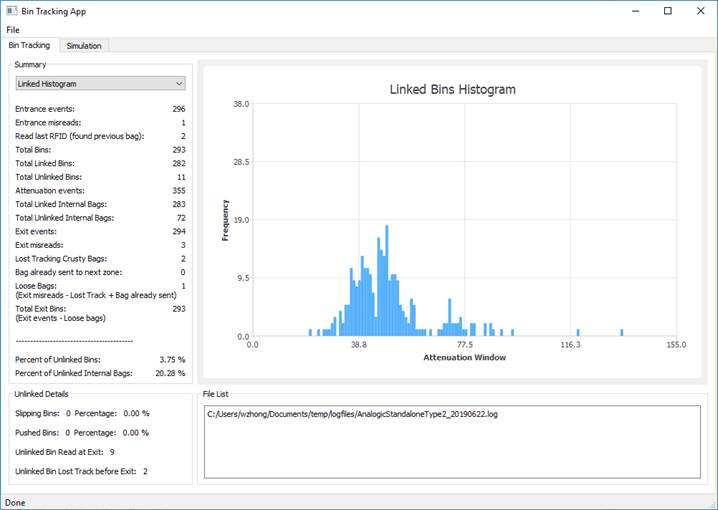


Figure 7. Latest App Layout

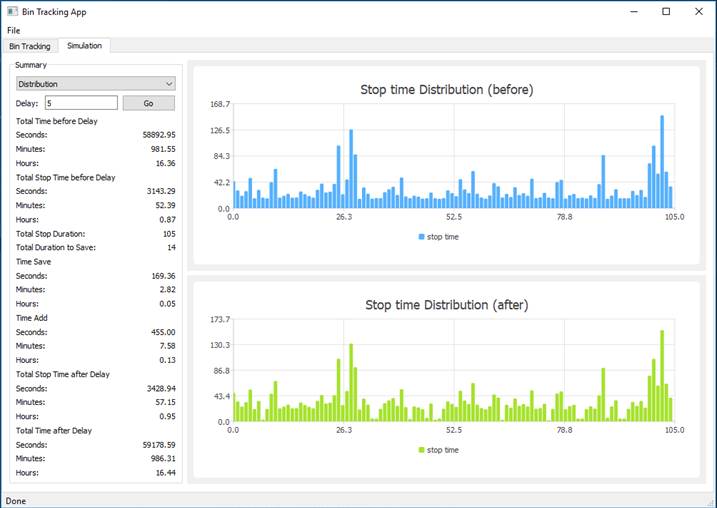


Figure 8. Latest App Simulation Layout

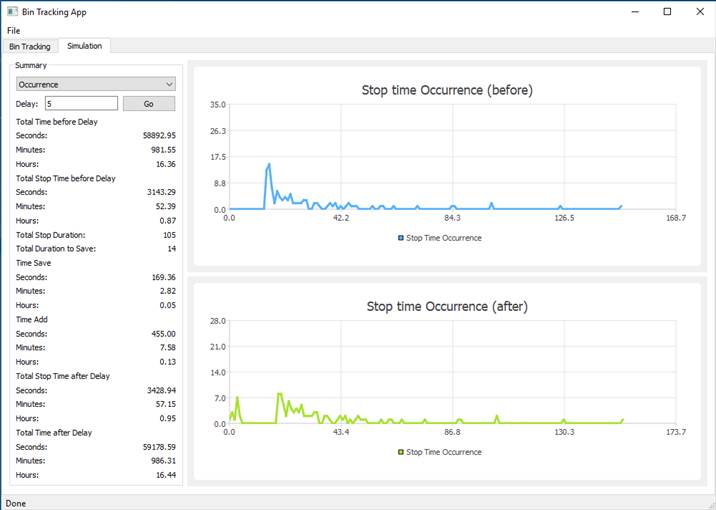


Figure 9. Latest App Simulation Layout

**Log File Analysis**

Unlinked bins:

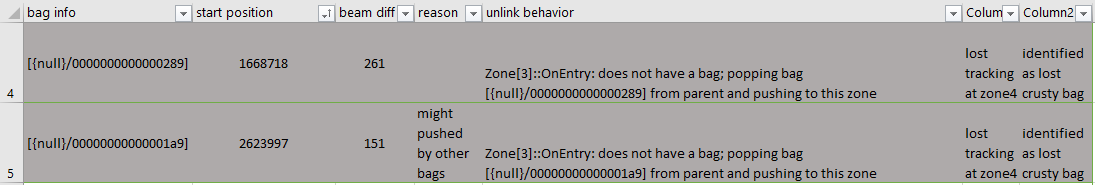
**What is the issue:**

The unlinked indicated at zone 4 might underestimated the actual unlinked bins. Consider the situation below:

(AnalogicStandaloneType2\_20190623.log)

Zone[3]::OnEntry: does not have a bag; popping bag [{null}/0000000000000289] from parent and pushing to this zone

Zone[3]::OnEntry: does not have a bag; popping bag [{null}/00000000000001a9] from parent and pushing to this zone



Those two bins are never linked at the attenuation window. They lost tracking before they are made decision at zone 4. (might take by operator to search, no evidence to show that the bag was been rescanned)

**How did I aware of this:**

“should have reached attenuation” at the attenuation window and “is unlinked” event at zone 4 have the different set of RFIDs in it. Some RFID is in former statement but not in latter statement (bag might lose tracking); some are in zone 4 but not in attenuation window (bag might flush out or in X-ray Min etc., scanner doesn’t know).

**Ways to solve:**

1. Use linked bin counts at the attenuation window, subtract by total bin entered. (How to find total bin? See next section.)
2. Keep track of RFID bin created at the entrance and corresponding linked bins, for those of remain not touched, are the unlinked bins.

How to ensure correctness:

Compare result from solution 1 and solution 2, should have the same number.

**The implementation:**

1. Total bins: (will talk about it in next section)

Total linked bins: (search for “Linked internal bag\_id”, get total count)

Total unlinked bins: (Total bins – Total linked bins)

1. Make a queue to store events: first push “CREATED new bag” except “\_\_I\_\_” (only care about real RFID) into queue, then compare the RFID at the queue head with “Linked internal bag\_id” except “\_\_I\_\_” event’s RFID is the same, pop out the RFID, we are done for this RFID bin; If is not the same RFID, pop head until find the same RFID, those bins are popped out are considered as unlinked. Finally, if there are bins remain on the queue, also consider as unlinked.

Total unlinked bins = total unlinked calculated + queue remain

Program output:

Total bins: 1403

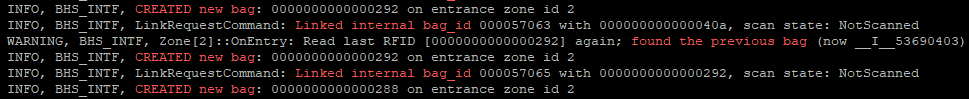
Total unlinked by queue calculation: 8

Total linked: 1395

Total unlinked by subtract from total bins - linked count: 8

Special condition that need to consider (solution 2):

“found the previous bag”



Avoid pushing another “CREATED new bag” into queue, this RFID already in queue (might pulled by operator, gets the read again)

RFID tracking:

**What is the issue:**

If solely rely on “CTEATED new bag” to get total RFID bins entered to the scanner, it is not stable. Some cases might happen such as “Read last RFID” (read repeat RFID, see above figure). The “RFID tag read at the entrance” might be overestimated.

**How did I aware of this:**

The RFID counts at the entrance is inconsistence with RFID counts at the exit. So, we might need to do some calculations to get the actual RFID bins enter to the scanner.

**Ways to solve:**

“Read last RFID” has two difference conditions:

1. “found the previous bag” (read RFID twice, the one before current read is marked as artificial RFID id)
2. “Did not find the bag in the next zone” (allow this read)

The condition 1 would affect the counts for RFID, so it need to subtract from the total events.

**How to ensure correctness:**

Assume the number of bins enter to scanner are the same as the bin exit the scanner. Compare the entrance RFID bin and exit RFID bin has same number.

**The implementation:**

1. RFID at the entrance: get counts for “CREATED new bag” events except “\_\_I\_\_” (this should be RFID only counts); take care of “found the previous bag” condition, if any, subtract the number from total RFID bin reads

Actual RFID bins at entrance = RFID reads at entrance – previous bag num

1. RFID at the exit: there are two conditions we need to consider “We already sent that bag to the next zone; moving it back” and “Removing crusty normal bag from front”

Actual RFID bins at exit = RFID reads at exit + crusty bag – move back bag

Program output:

Entrance events: 1410

Entrance RFID misread: 6

read last RFID: (found previous bag): 1,

(Did not find bag in next zone, allow this read as is): 0

RFID at entrance: 1403

Exit events: 1417

Exit RFID misreads: 16

RFID read but bag already sent to next zone: 1

Removing crusty normal bag from front: 3

RFID at exit: 1401

Applying the equations to the number:

Actual RFID bins at entrance = RFID reads at entrance – previous bag num

= (1410 – 6) – 1

= 1403

Actual RFID bins at exit = RFID reads at exit + crusty bag – move back bag

= (1417 – 16) + 3 – 1

= 1401 + 3 – 1

= 1404 – 1

= 1403

1403 == 1403

Actual RFID bins at entrance == Actual RFID bins at exit

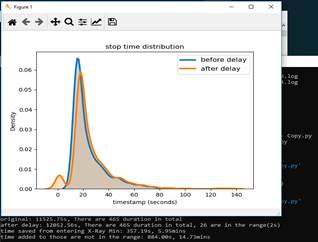
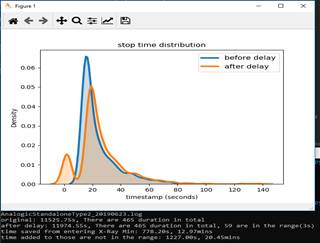
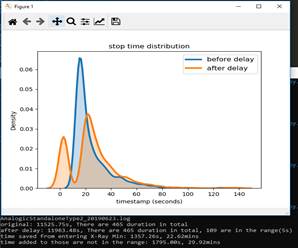
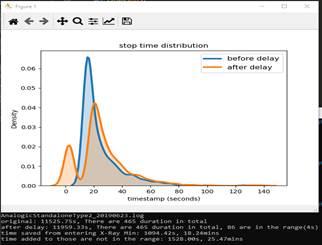
So, the total bins that enter to the scanner can be found by either equations given above.

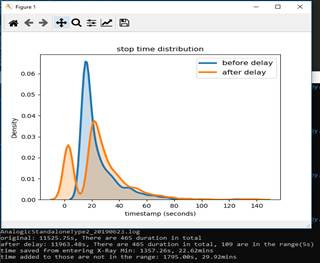
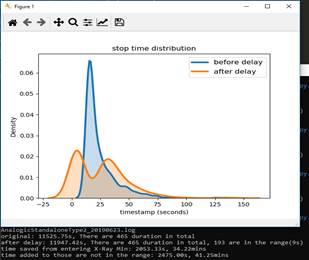
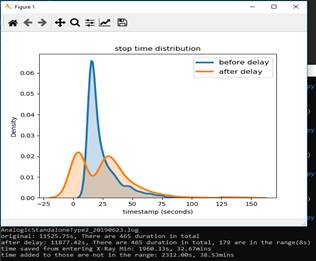
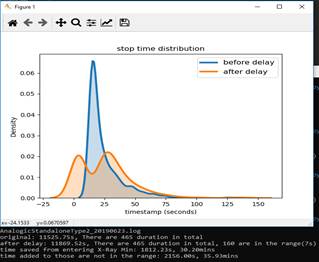
**Simulation delay**

* Graph plot total stop time before/after delay

File name: AnalogicStandaloneType2\_20190623.log

The blue shed is the original stop time and the orange shed is stop time after delay. These graph shows from range 2s delay to 9s delay (left to right, top to down accordingly).

A few comments:

* I show three screens available with the current CentOS version in Accurev: Linked Histogram, Unlinked Donut Chart, Summary Pie Chart
* The specification has more screens: Unlinked Bins Histogram with Data, Stop Time Distribution Before/After Delay, Stop Time Occurrence Before/After Delay, etc.
  + Are these being developed?
    - Add it back anytime only if it’s needed
* The test log files are from a particular baggage handling system (BHS): AnalogicStandaloneType2, AnalogicStandaloneType5
  + Does it work for other types of BHS?
    - Virtual BHS
    - MacH BHS
    - Vanderlande BHS
    - Herbert BHS
  + The log files for the other BHS may share some keyworks but are not the same.
    - Do you need example log files?
      * Yes, add config file to each model

Output file save into one separate folder