MUNICIPALITY OF ANCHORAGE

ASSEMBLY MEMORANDUM

No. AM 519-2022

Meeting Date: September 27, 2022

Municipal Clerk's Office
Approved
Date: September 27, 2022

From: **MOA Elections Team**

Subject: **Risk Limiting Audit for the April 5, 2022 Regular Municipal Election**

I. EXECUTIVE SUMMARY

The MOA Elections Team, with members of the Anchorage Election Commission, conducted a post-election audit that contained three areas of focus. The audit was public noticed and candidates and observers were invited to attend.

1. ***Hand-Count.*** A pre-determined percentage of ballots in specified contests was selected and the actual ballots were hand counted.
2. ***Machine Review.*** After the hand-counting, the votes on the actual ballots were also compared to an image of the same ballots in the tabulation system and the adjudication of the votes in those specified contests was confirmed. Cast Vote Records were produced from the tabulation system and tallied for the ballots selected.
3. ***Comparison of Hand-Count and Machine Review.*** The totals from the hand-count, detailed in paragraph 1, and the totals from the machine count, detailed in paragraph 2, were compared.

   The results of the MOA post-election Risk Limiting Audit are that the scanning, adjudication, and tabulation system performed as expected and the results reflect the will of the voters. All ballots were adjudicated and tabulated as expected. The results of the hand-count and the machine tabulation were identical.[1]

II. WHAT IS A POST-ELECTION RISK LIMITING AUDIT?

A. ***Research.*** Research defines a post-election audit is a check to confirm that the voting equipment and procedures used to count votes worked properly. Post-election

[1] For more detailed information on the results of the audit, see Item G. Comparison of the Hand-Count to the Machine Count, Results of the Risk Limiting Audit and Exhibit A - RLA Worksheet.

audits are recommended by election security experts as one method of protecting the integrity of elections.

There are many types of "post-election audits" used to validate election results or outcomes. As a term of art, it refers to checking paper ballots (or records) against the results produced by the vote tallying equipment to ensure accuracy.

Risk limiting audits (RLA) use statistically developed audit techniques that allow selection of a number of ballots to be audited that provide statistical confidence that the tabulation system performed as expected. A RLA is an incremental audit system: If the percentage of risk selected in advance of the audit failed to demonstrate the tabulation system was performing as expected, election officials would review further ballots or conduct a full manual tally of the election.

The MOA Election Team elected to conduct a "Batch-Level Comparison Audit," which is a type of RLA that most resembles a "traditional" audit. In a batch-level comparison audit, the voting system must export identifiable physical batches of ballots. In the MOA RLA, Election Officials physically selected random batches from the entire election to audit. In "Batch-Level Comparison Audits" and in the MOA RLA, Election Officials add up the selected batch-level results by hand to verify that they produce the reported contest outcomes. The votes in each selected batch are and were examined manually and hand-counted, and the audit counts are compared to the tabulation system's report and subtotals. Depending on the number and type of discrepancies the audit finds in the sample, the audit either stops or examines more batches manually.

B. Implementation of the Risk Limiting Audit at the MOA

Successful implementation of any new election process requires careful thought and a considerable amount of planning. The MOA Elections Team began looking at post-election audits in 2020. One important step in preparing for the post-election audit, was obtaining the imprinters on the ballot scanners in 2020; the imprinters put a unique number – the scanner, batch, and ballot number – on each ballot, allowing elections officials the ability to pull the actual ballot to confirm the votes. Elections Officials used this technology during two recounts in 2021 and during adjudication in 2021 and 2022 when looking at certain marks and blank ballots among other items needed review of the actual ballot.

The MOA Elections Team conducted a practice audit after the 2021 Regular Municipal election in preparation for implementation of post-election audit in 2022. The practice was worthwhile: The Elections Team determined it tested too many ballots in one race and too few in another; the Elections Team pulled individual ballots which was incredibly time consuming. To address this shortcoming, the 2022 audit tested "batches" of ballots, selecting 30 batches to test with approximately 100 ballots in each batch, which was more efficient to select and re-file rather than randomly selecting individual ballots and having to replace those.

III. PROCEDURES FOR THE RISK LIMITING AUDIT

Risk limiting audits use statistically developed audit techniques that allow selection of a number of ballots to be audited that provide statistical confidence that the tabulation system performed as expected. The MOA Election Team conducts a "Batch-Level Comparison Audit," which is a type of RLA that most resembles a "traditional" audit.

A. ***Selection of Races and Measure to be audited.*** The MOA Risk Limiting Audit Procedures requires the Elections Team to identify the races and measures to be audited by rolling a 6-sided die. There was no mayor's race in 2022, so the options were to select one Assembly race and one proposition to audit.

Assembly Race. The Elections Team first rolled the 6-sided die and the result was a 1. Since Assembly District 1 was not on the ballot in 2022, the die was rolled again, and the result was a 3, or Assembly District 3 was selected to be audited.

Measure or proposition to be audited. The Elections Team rolled the 6- sided die and rolled a 4. Since Proposition 4 was an area-wide race, Proposition 4 was selected to be audited.

B. ***Target Number of Ballots.*** The MOA procedures for the RLA targeted 3% of the ballots cast in the election, or 2,119 ballots, for the areawide measure to be audited. The procedures targeted 1.5% of the total ballots cast in district-wide measures, or 194 ballots, for the district-wide race to be audited.

The exact calculations for target number of ballots are as follows:

- Calculate 3% of ballots cast in the in the Proposition selected, regardless of the number of votes cast or spread. Round down to nearest 1,000. E.g., change 71,345 to 71,000 for ease of count:
  - In 2022, total ballots cast = 70,639 x .03 = 2,119

- Calculate 1.5% of total votes cast in the Assembly race:
  - In District 3, total votes cast (note, this is different that total ballots cast used for areawide races) = 12,957 x .015 = 194

The audit actually reviewed 2121 ballots in the areawide measure and 213 in the district-wide measure because two teams of three were working side-by-side and, when we paused to see how many ballots we had reviewed, we realized we had exceeded the target number.

C. ***Random Selection of Batches.*** To reach the 2,119 ballots targeted for review in the Municipal-wide Proposition 4 ballot measure, the MOA Elections Team estimated a minimum of 21 batches would be required to be audited since during the processing of the election, approximately 100 ballots were scanned per

batch. (2,119/100 = 21.) The Team selected 30 batches for this audit – in the event that some of the batches could contain less than the 100 ballots typically scanned per batch.

Then, staff calculated the percentage of total ballots processed in the election on ICC 1 (scanner 1) and ICC 2 (scanner 2). (ICC 3 did not scan enough ballots to be included.) The result is that 14 batches from ICC1 and 16 batches from ICC2 would be audited.

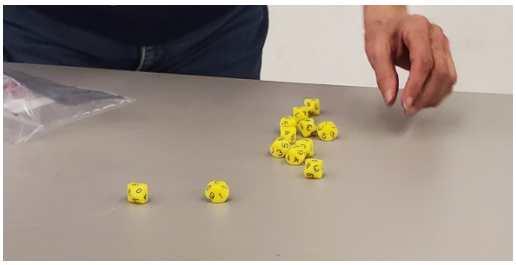
The exact calculations for the number of batches selected from each scanner are as follows:

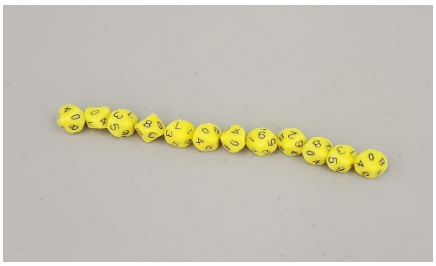
1. Determine the total number of batches scanned by each selected ICC:
   - ICC 1= 365 batches
   - ICC 2 = 407 batches
   - 772 total batches to possibly be verified

     We did not select any batches from ICC 3 because it was used rarely.
2. Determine the percentage of total batches each ICC scanned:
   - ICC 1 = 365/772 = 47%
   - ICC 2 = 407/772 = 53%
3. For each ICC selected, use the percentage of total batches each ICC scanned to determine the random number of batches needed from each ICC, and then to determine which batch numbers for each ICC to pull. Since 30 batches were selected for verification, the total number of batches for verification from each ICC is as follows:
   - ICC 1 = 47% of total batches x 30 batches for verification = 14
   - ICC 2 = 53% of total batches x 30 batches for verification = 16

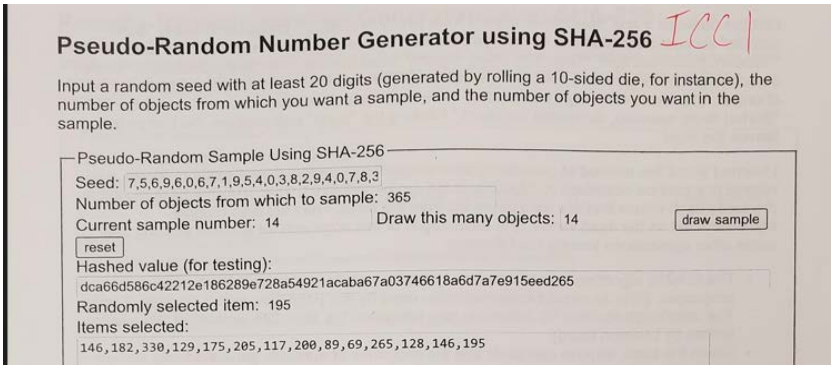
D. Use Pseudo-Random Number Generator for Random Selection of Batches. The staff then used a Pseudo-Random Number generator at https://www.stat.berkeley.edu/~stark/Java/Html/sha256Rand.htm to randomly select the batches of ballots from each ICC. Following the instructions on the Pseudo-Random Number Generator, the batches were selected as follows:

(1) Roll the ten-sided dice 1 time, and then a second time and input this information into the “Seed”.





(2) Enter the "Seed" and other information into the random number generator and press "Draw Sample." The result is the list of randomly selected items. Here, batches 146, 282, 330, 129, 175, 205, 117, 200, 89, 69, 265, 128, 146, and 195 were selected for ICC 1.



The process was repeated for ICC 2 and batches 141, 372, 260, 5, 379, 217, 11, 88, 318, 328, 245, 81, 314, 266, 67, and 6 were selected for ICC 2.

The 14 batches for ICC 1 and the 16 batches for ICC 2 were pulled and delivered to the counting teams.

E. The Hand-Count

The hand-count is the manual tallying of the preselected ballots.

Assembly District 3 – For each batch selected as detailed in section III.D. (Random Selection of Batches), the ballots were sorted for hand-counting by the top two candidates - Candidate 1 and Candidate 2 - and Other. The "Other" category included votes for other candidates, a blank race, or if the District 3 race wasn't on the selected ballot. After the ballots were sorted, the ballots were hand-counted and tallied by batch and double checked by a second election worker. The batch results were added to the RLA Worksheet. The results of the hand-count from the RLA Worksheet are as follows:

Category	**Hand-Count**
Candidate 1	145
Candidate 2	68

Total	**213**

Again, the audit actually reviewed 213 ballots in district-wide measure, which is more than the identified 194. This was because two teams of three were working side-by-side and, when we paused to see how many ballots we had reviewed, we realized we had exceeded the target number. The increase in the number of ballots reviewed gives more confidence in the audit because it increases the possibility that an incorrect outcome could be detected.

Proposition 4 – Repeating the process for Proposition 4 that was completed for Assembly District 3, for each batch selected, the ballots were sorted by Yes, No, and Other. The "Other" category included a blank measure or an overvote. After the ballots were sorted, the ballots were hand-counted and tallied by batch and double checked by a second election worker. The batch results were added to the RLA Worksheet. The results of the hand-count from the RLA Worksheet are as follows:

Category	Hand-Count
Yes	1282
No	839
Total	**2121**

Again, the audit actually reviewed 2121 ballots in the areawide measure, which is more than the identified 2119. This was because two teams of three were working side-by-side and, when we paused to see how many ballots we had reviewed, we realized we had exceeded the target number.

F. Machine Count Verification

After the batches of ballots were hand-counted, the scanning, tabulation, and adjudication system was set up and two election officials pulled up the batches of ballots that were randomly selected for review in the audit as detailed in section III.D. (Random Selection of Batches). Each ballot selected for review was checked to confirm that the scan was the same as actual ballot; the "audit mark" in the system was also checked to confirm how the actual ballot was adjudicated. During the audit, records were maintained on an "Anomaly Log" to document any ballots that were not recorded in the system the same as the votes on the actual; there were no anomalies; all ballots were recorded in the system as marked on the actual ballot.

The Cast Vote Record (CVR) was produced and tallied for the selected batches of ballots. The batch totals were transferred to the RLA Worksheet[2] and are as follows:

[2] See Exhibit A – RLA Worksheet

Assembly District 3

Category	**Machine-Count Total**
Candidate 1	145
Candidate 2	68
Total	**213**

Proposition 4

Category	Machine-Count Total
Yes	1282
No	839
Total	**2121**

G. Comparison of the Hand-Count to the Machine Count, Results of the Risk Limiting Audit

The third and final step in the post-election audit was to compare the hand-count to the machine count. The comparison is as follows:

Assembly District 3

Category	**Hand-Count**	**Machine-Count Total**
Candidate 1	145	145
Candidate 2	68	68
Total	**213**	**213**

Proposition 4

Category	**Hand-Count**	**Machine-Count Total**
Yes	1282	1282
No	839	839
Total	**2121**	**2121**

The result of the post-election audit is that of 2,121 randomly selected ballots in the areawide race and for 213 randomly selected ballots in the district-wide race, the hand count and machine count of those ballots were identical. The conclusion is that the scanning, adjudication, and tabulation system performed as expected and the results of the election demonstrated the will of the voters.

IV. LITERATURE REVIEW AND DEFINITIONS

Literature Review.

"Checking the Election: Risk-Limiting Audits," by Dylan Lynch. NCSL, LegisBrief, July 2019, Vol. 27, No. 26. https://www.ncsl.org/research/elections-and-campaigns/checking-the-election-risk-limiting-audits.aspx, Accessed May 18, 2020.

"Knowing It's Right, Part One: A Practical Guide to Risk-Limiting Audits," by Jennifer Morrell. Democracy Fund, May 2019. https://electionline.org/resources/rla-practical-guide/, Accessed May 18, 2020.

"Knowing It's Right, Part Two: Risk-Limiting Audit Implementation Workbook," by Jennifer Morrell. Democracy Fund, May 2019. https://issuu.com/democracyfund/docs/2019_df_knowingitsright_part2. Accessed May 18, 2020.

Interesting Definitions.

Outcome: outcome means the winner, not the tabulated vote totals.

Target contest: The Target contest is the contest selected for a risk limiting audit. There can be one or more target contests. (The Target contest determines the number of ballots that must be examined during the RLA and is used to determine if the audit has met the risk limit.)

Sample Size/Workload: The sample size or workload is the number of ballots required to be inspected before the audit can stop. The sample size depends on many things. Including whether the audit is a batch-level comparison audit or a ballot-polling audit. Although the calculation itself is complicated, in one example, a jurisdiction with over 2 million ballots cast, using a risk limit of 10%, sampled 49 ballots in a ballot level comparison audit and sampled 384 ballots in a ballot-polling audit. Because the calculation is so complicated and needs a statistician to explain it, the MOA Elections Team believed the State of Alaska Division of Elections audited 3% of ballots and selected that number. However, upon further review, the State of Alaska Division of Elections audits 5% of ballots returned and in future RLAs, the MOA will select 5% of ballots to be audited.

Respectfully Submitted:
MOA Elections Team:
Barbara A. Jones, Municipal Clerk
Jamie Heinz, Election Administrator

Scanner & Batch #	Handcount Column A Yes	Handcount Column B No	N= 2119	Machine Batch Level Results Prop Yes	Machine Batch Level Results Prop No	Total Proposition	Handcount Column D Candidate 1	Handcount Column E Candidate 2	N = 194	Machine Batch Level Results Candidate 1	Machine Batch Level Results Candidate 2	Total Race
1-69	53	18		53	18		4	6		4	6	
1-89	20	22		20	22		12	14		12	14	
1-117	61	8		61	8		37	13		37	13	
1-128	62	25		62	25		23	15		23	15	
1-129	72	20		72	20		16	2		16	2	
1-146	62	11		62	11							
1-175	78	31		78	31							
1.182	50	26		50	26							
1-195	55	49		55	49							
1-200	34	50		34	50							
2-5	39	5		39	5		3	0		3	0	
2-6	71	38		71	38		48	18		48	18	
2-11	68	26		68	26		0	0		0	0	
2-67	39	38		39	38		2	0		2	0	
2-81	56	68		56	68		0	0		0	0	
2-88	53	47		53	47		0	0		0	0	
2-141	57	43		57	43							
2-217	43	49		43	49							
2-245	60	39		60	39							
2-260	53	52		53	52							
2-266	54	36		54	36							
2-314	44	67		44	67							
2-318	32	31		32	31							
2-328	66	40		66	40							
	1282	839	2121	1282	839	2121	145	68	213	145	68	213

Risk Limiting Audit Instructions

Purpose: To audit Election results to confirm accurate tabulation

Frequency
Once after each election.

Performed By
Core Team, Project Manager, Four Election Workers

Required Materials

- 12 Ten-Sided Dice
- 1 Six-Sided Dice
- Random Number Generator (https://www.stat.berkeley.edu/~stark/Java/Html/sha256Rand.htm)
- Ballots
- Tally Sheets
- Anomaly Log
- Adjudication System
- Completed Batch Cover Sheets "Not in Numeric Order Due to RLA"
- White, yellow, green, and red trays
- Red pens, markers

A. Identify races or measures to audit.

1. Mayor's race. Will count top two candidates only.
2. Using six-sided dice, randomly select 1 of the Assembly races, by District number. Will count top two candidates only.
3. Using six-sided dice, randomly select 1 or more propositions each year.

B. Identify Target Number of ballots per race or measure.

1. Calculate 3% of ballots cast in the Mayor's race or in the Proposition selected, regardless of the number of votes cast or spread. Round down to nearest 1,000. E.g., change 71,345 to 71,000 for ease of count. (Use the same methodology for a School Board race.)

2. Because the potential number of ballots cast in an Assembly race is approximately 1/6 of the total ballots cast in Municipal wide races or measures, calculate 1.5% of total votes cast in the Assembly race. Round down to nearest 100 for ease of count.

C. Randomly select batches in the race.

1. Based on the number of ballots to be reviewed per race or measure as determined in paragraph B., if approximately 100 ballots were scanned per batch, select the total number of batches to comfortably reach the 3% or 1.5% of ballots to be targeted.

FOR EXAMPLE: Assume 75,000 ballots cast in mayor's race X 3% = 2,250 ballots to verify. Assuming approximately 100 ballots were scanned per batch, randomly select approximately 25 batches, (25x100 = 2,500) which would exceed the 2,250 ballots to be selected for verification.

2. Determine the total number of batches scanned by each ICC.

3. Determine the percentage of total batches each ICC scanned.

4. For each ICC selected, use the percentage of total batches each ICC scanned to determine the random number of batches needed from each ICC, and then to determine which batch numbers for each ICC to pull.

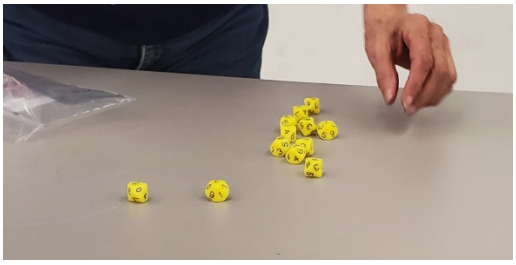
   a. Use the percentage of total batches scanned at each ICC in Step C.3., x the number of batches targeted to be verified in Step C.1.

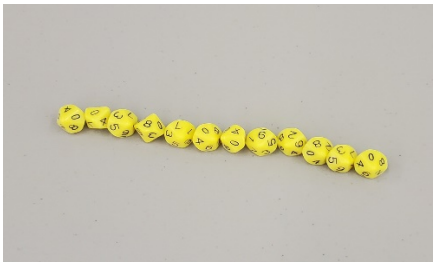
   - <u>For example, ICC 1 = 60% of total batches x 25 batches = 15 batches would be selected for verification from ICC 1.</u>

   b. Use Pseudo-Random Number Generator (site link above) and follow instructions to generate a random selection of batches to pull for the audit. (See attached example copy of Pseudo-Random Number Generator page.)

   c. Conduct the following steps for each ICC:

      (1) Roll the ten-sided dice 10 times, and then 10 times again and input this information into the "Seed".





(2) Enter the "Number of objects from which to sample" = the number for this ICC from Step C.2, the total number of batches scanned by the ICC.

(3) Enter the "Draw this many objects" = the number for this ICC from Step C.4.a, the number of batches needed from this ICC. (The computer generates the "Current sample number." Do not enter the number in that field.)

(4) Hit the "draw sample" button.

(a) Count to confirm the number of "Items Selected" if the same as the "Draw this many objects." If not, hit the "reset" button and run again.

(b) Print a copy of the results of the Pseudo-Random Number Generator report for each ICC. (You cannot save it.)

d. Repeat the process for other ICCs.

e. Tips for the Pseudo-Random Number Generator:

- Number of Objects to Sample = C.2. above, the total number of batches scanned by the ICC.
- Draw this Many Objects = C.4.a., above, the random number of batches needed from each ICC.
- Count to make sure the number of batches equals C.4.a. and is the same as the request to "Draw this Many Objects." If not, press reset and reenter the numbers and redo process.
- Reset the Pseudo-Random Number Generator and repeat the steps to generate batches to be pulled for the other ICCs.

D. Pull the batch boxes from the vault.

Pull the boxes from the vault containing the batch numbers determined from the results of the Pseudo-Random Number Generator as detailed in the printed report in Step C.4.c.(4)(b). Keep the boxes from each ICC separated on their own carts.

E. Tally Each Batch by Hand Count, then Sort

1. Working in teams of two, starting with the lowest batch number, pull the first randomly selected batch by removing the batch cover sheet and all of the batch's ballots from the batch box.

2. Complete the RLA Batch Cover Sheet (see attached sample). (Keep the original Batch Cover Sheet and both the new RLA Batch Cover Sheet and the original will be retained.)

3. For each batch, there will be two counts for each batch, (1) the "yes" votes and "no" votes for the proposition, and (2) the votes for the top two candidates.

A. Start with the proposition. Sort the ballots from the batch into three trays categories – “Yes," "No," and other (undervote, overvote).

Once sorted, count the "Yes," "No," and “Other” and write the numbers on the Risk Limiting Audit Tally Sheet.

April 5, 2022 Regular Municipal Election — **Risk Limiting Audit Tally Sheet** — May 17, 2022

		Count 1			Count 2		
		Column A	Column B	Column C	Column D	Column E	Column F
Tray # 1							
ICC #	Batch #	Prop - Yes	Prop - No	Prop - Blank	Candidate 1	Candidate 2	Everything Else
2	5	39	5	Ø	3	Ø	41
Tray # 2							

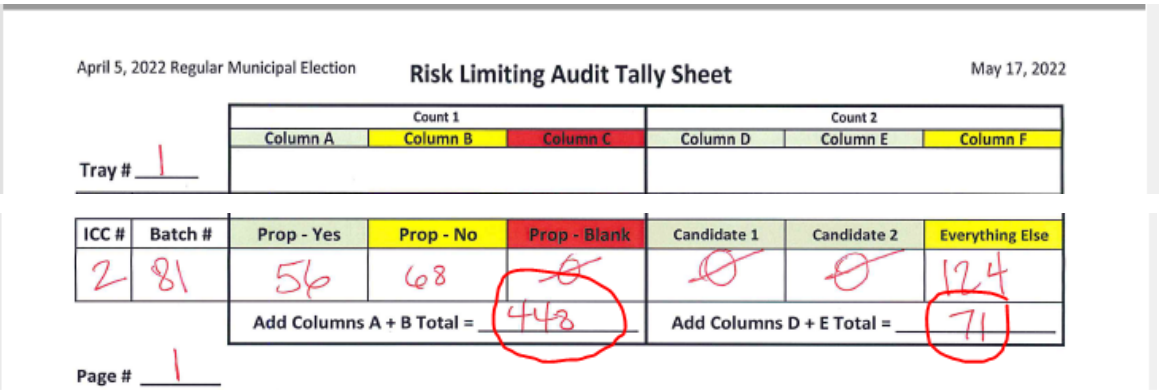
B. Continue with the candidate race. Sort the ballots in the batch into three trays appropriately – Candidate 1, Candidate 2, and other (undervote, overvote, other candidate).

Once sorted, count the Candidate 1 votes, the Candidate 2 votes, and the “other” votes, and write these numbers on the RLA tally sheet. Write the totals on the RLA Batch Cover Sheet.

4. Transfer to Adjudication. Place the Candidate 1 and Candidate 2 ballots crosswise in 1 tray; place the other category crosswise or in another tray. Document the batch number on the top of the ballots. Place the RLA batch cover sheet on top of the tray(s). Place on the ICC rack in the cage for the adjudication verification.

F. Repeat step E for each randomly selected batch. Continue to tally up the ballot count from each randomly selected batch on the RLA Tally Sheet until 3% or 1.5% of ballots is reached. (See Risk Limiting Audit Tally Sheet and Risk Limiting Audit Tally Adding Sheet.)

1. Calculate the page total votes marked YES, NO, and Other and add these numbers together at the bottom of the page on the Risk Limiting Audit Tally Sheet. Repeat and calculate the page total votes marked for Candidate 1, Candidate 2, and Other, and add to the tally sheet.



April 5, 2022 Regular Municipal Election — **Risk Limiting Audit Tally Sheet** — May 17, 2022

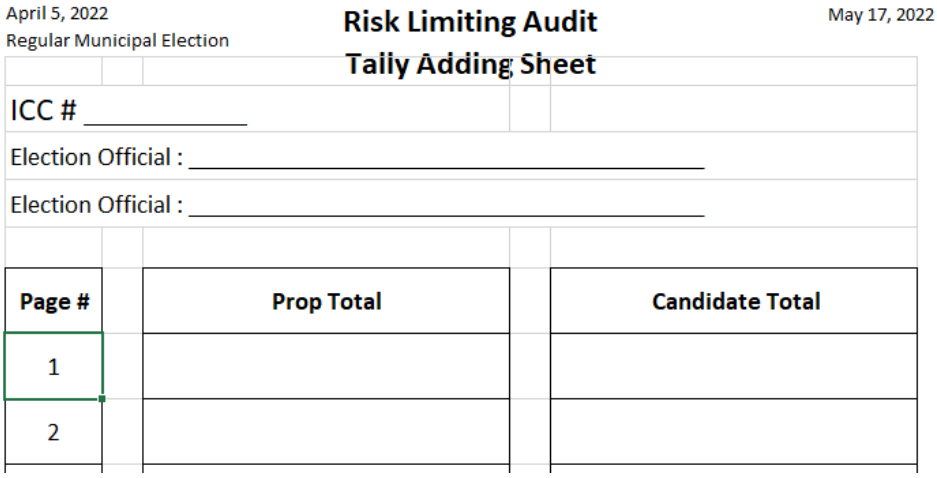
Tray # 1

		Count 1			Count 2		
		Column A	Column B	Column C	Column D	Column E	Column F

ICC #	Batch #	Prop - Yes	Prop - No	Prop - Blank	Candidate 1	Candidate 2	Everything Else
2	81	56	68	~~0~~	~~0~~	~~0~~	124
		Add Columns A + B Total = 443			Add Columns D + E Total = 71		

Page # 1

2. Transfer the page totals from the Risk Limiting Audit Tally Sheet to the Risk Limiting Audit Tally Adding Sheet, for each ICCs.



April 5, 2022
Regular Municipal Election

Risk Limiting Audit Tally Adding Sheet

May 17, 2022

ICC # _____

Election Official : _____

Election Official : _____

Page #	Prop Total	Candidate Total
1		
2		

3. When the totals for both ICCs equal the total number of ballots sought to be verified in the election, stop tallying.

a. If, after all batches and ballots selected, the totals for both ICCs does NOT equal the total number of ballots sought to verified in the election, randomly select additional batches if needed.

G. Adjudication Verification

With one person at each adjudication computer reviewing separate batches, open the batch and compare the votes marked on the actual ballot to the candidate and propositions marked in the image of the ballot on the screen and then verify the ballot was adjudicated correctly on the audit mark page of the image. For any anomalies, first confirm you are reviewing the correct ballot number. If you are, and still see an anomaly, communicate with the Supervising Election Official\Deputy Municipal Clerk –

Elections for further review and resolution. Note any unresolved anomalies on the Anomaly Log.

When finished with each batch, wrap that batch in the RLA cover sheet over the original batch cover sheet indicating that the batch is out of numeric order due to the Risk Limiting Audit and place the batch back in the box it originated from.

REFERENCE: Number of total estimated ballots.

Assuming an audit of one proposition each year, and one major race (either mayor or a randomly selected assembly seat), the smallest number of total ballots to hand count each year could look like this, based on a sampling of 2020 and 2021 election results:

Year 1

Assembly seat: 20,000 x .015 = 300
Prop: 71,000 x .03 = 2,130*
Total ballots audited = greater of 300 or 2,130: 2,130

Year 2

Mayor: 75,000 x .03 = 2,250
Prop: 71,000 x .03 = 2,130*
Total ballots audited = greater of 2,250 or 2,130: 2,250

*This is based on the assumption that the proposition selected will be an areawide question. Since current practice is to place areawide questions ahead of other questions on the ballot, the random selection of a question using a 6-sided dice should provide this outcome more often than not.

Pseudo-Random Number Generator using SHA-256 ICC1

Input a random seed with at least 20 digits (generated by rolling a 10-sided die, for instance), the number of objects from which you want a sample, and the number of objects you want in the sample.

Pseudo-Random Sample Using SHA-256

Seed: 7,5,6,9,6,0,6,7,1,9,5,4,0,3,8,2,9,4,0,7,8,3

Number of objects from which to sample: 365

Current sample number: 14 Draw this many objects: 14 draw sample

reset

Hashed value (for testing):

dca66d586c42212e186289e728a54921acaba67a03746618a6d7a7e915eed265

Randomly selected item: 195

Items selected:

146,182,330,129,175,205,117,200,89,69,265,128,146,195

Sorted items selected:

69,89,117,128,129,146,146,175,182,195,200,205,265,330

Sorted items selected, duplicates removed:

69,89,117,128,129,146,175,182,195,200,205,265,330

What this does

The "seed," concatenated with a comma and the "Sample number," is passed through the SHA-256 hash function. The result is displayed as "Hashed value (for testing)". The hashed value, interpreted as a hexadecimal number, is divided by "Number of objects from which to sample." One is added to the remainder of that division to get "Randomly selected item," which will be a

number between 1 and "Number of objects from which to sample," inclusive. Clicking "draw sample" successively adds one to "Sample number" and recomputes "Hashed value" and "Randomly selected item" "Draw this many objects" times. Selected items accumulate in "Items selected" (and "Sorted items selected"), which reset if the seed or the number of objects changes. The same item might be selected more than once. Duplicates are removed in the "Sorted items selected, duplicates removed." Clicking the "reset" button clears the history but leaves the seed.

I learned about this method of generating pseudo-random numbers from Ronald L. Rivest. It is related to a method described in https://tools.ietf.org/html/rfc3797. The SHA-256 hash algorithm produces hash values that are hard to predict from the input. They are also roughly equidistributed as the input varies. The advantages of this approach for election auditing and some other applications include the following:

- The SH-256 algorithm is public and many implementations are available in many languages. (The Javascript implementation used by this page was written by Brian Turek; The JavaScript routines for arithmetic long integers—the SHA-256 hashed values—were written by Leemon Baird).
- Given the seed, anyone can verify that the sequence of numbers generated was correct—that it indeed comes from applying SHA-256.
- Unless the seed is known, the sequence of values generated is unpredictable (so the result is hard to "game"). It is very hard to distinguish the output from independent, uniformly distributed samples.

For comparison, a reference implementation of this approach in Python written by Ronald L. Rivest is available at https://people.csail.mit.edu/rivest/sampler.py.

P.B. Stark, statistics.berkeley.edu/~stark. https://statistics.berkeley.edu/~stark/Java/Html/auditRand.htm Last modified 11 January 2017.

Pseudo-Random Number Generator using SHA-256

ICC 2

Input a random seed with at least 20 digits (generated by rolling a 10-sided die, for instance), the number of objects from which you want a sample, and the number of objects you want in the sample.

Pseudo-Random Sample Using SHA-256

Seed: 7,5,6,9,6,0,6,7,1,9,5,4,0,3,8,2,9,4,0,7,8,3

Number of objects from which to sample: 407

Current sample number: 16 Draw this many objects: 16 draw sample

reset

Hashed value (for testing):

d1163f714cabb3e49b9359723d8d9096fb517e4c77e2b9c959e742057af30641

Randomly selected item: 6

Items selected:

141,372,260,5,379,217,11,88,318,328,245,81,314,266,67,6

Sorted items selected:

5,6,11,67,81,88,141,217,245,260,266,314,318,328,372,379

Sorted items selected, duplicates removed:

5,6,11,67,81,88,141,217,245,260,266,314,318,328,372,379

What this does

The "seed," concatenated with a comma and the "Sample number," is passed through the SHA-256 hash function. The result is displayed as "Hashed value (for testing)". The hashed value, interpreted as a hexadecimal number, is divided by "Number of objects from which to sample." One is added to the remainder of that division to get "Randomly selected item," which will be a

number between 1 and "Number of objects from which to sample," inclusive. Clicking "draw sample" successively adds one to "Sample number" and recomputes "Hashed value" and "Randomly selected item" "Draw this many objects" times. Selected items accumulate in "Items selected" (and "Sorted items selected"), which reset if the seed or the number of objects changes. The same item might be selected more than once. Duplicates are removed in the "Sorted items selected, duplicates removed." Clicking the "reset" button clears the history but leaves the seed.

I learned about this method of generating pseudo-random numbers from Ronald L. Rivest. It is related to a method described in https://tools.ietf.org/html/rfc3797. The SHA-256 hash algorithm produces hash values that are hard to predict from the input. They are also roughly equidistributed as the input varies. The advantages of this approach for election auditing and some other applications include the following:

- The SH-256 algorithm is public and many implementations are available in many languages. (The Javascript implementation used by this page was written by Brian Turek; The JavaScript routines for arithmetic long integers—the SHA-256 hashed values—were written by Leemon Baird).
- Given the seed, anyone can verify that the sequence of numbers generated was correct—that it indeed comes from applying SHA-256.
- Unless the seed is known, the sequence of values generated is unpredictable (so the result is hard to "game"). It is very hard to distinguish the output from independent, uniformly distributed samples.

For comparison, a reference implementation of this approach in Python written by Ronald L. Rivest is available at https://people.csail.mit.edu/rivest/sampler.py.

P.B. Stark, statistics.berkeley.edu/~stark. https://statistics.berkeley.edu/~stark/Java/Html/auditRand.htm Last modified 11 January 2017.