Voting Methods Working Group (VMWG) , NIST Interoperability Working Group Modeling Subgroup for Multi-round Voting Tabulation Methods

Authors: H. Deutsch, G. Gilbert, D. Cary, C. Hughes, L. Lochridge (Author & Editor)

09/08/2017   DRAFT

Overview

This specification section describes models of the tabulation process for multi-round voting methods. Specifically, this section describes the variety of multi-round voting method where there is a set of candidates in competition to fill one or more seats in a given elected office and a full tabulation of votes may require more than one iteration of tallying to determine the outcome of the election contest.

This section of this standard specification provides a descriptive, rather than prescriptive, model of a tabulation process flow for multi-round voting methods in use in the United States where the voter ranks candidates in order of the voter's preference.

In this document, Instant Runoff Voting (IRV) is a term that is used to describe a single-seat RCV contest variety. Single Transferable Vote (STV) is a term that is also used when the voting method is a multi-seat RCV contest. STV is generally considered in the literature, to be equivalent to IRV in a single seat case.

A variety of RCV variants are currently used in some jurisdictions of the United States.

This RCV tabulation process model provides for common variants of single and multi-seat contests.

An RCV method process definition may set a threshold vote count that a candidate must reach or exceed in order to be elected. The tabulation process occurs in one or more rounds with one tally of votes per round. In each round, typically, one or more candidates with the fewest votes are eliminated.

Typically, the votes for eliminated candidates are transferred to the next highest ranked non-eliminated candidate until a candidate exceeds the threshold or until the number of candidates that have not been eliminated matches the number of seats to be elected. If there are no more choices available, the ballot is considered exhausted and is no longer processed.

A round in a multi-seat contest may elect one or more candidates and transfer surplus votes in excess of the threshold to other candidates. The number of rounds required will be impacted by the number of seats to be filled, the number of candidates competing for those seats, the votes accumulated for those candidates and configuration options for the particular instantiation of an election contest.

This RCV process specification is designed to be general and flexible enough to be used for the most common variants of RCV in use in the United States.

There are several jurisdictions in the United States where RCV is in use, including the cities of San Francisco, Berkeley, Oakland and San Leandro in California; Telluride, Colorado; Portland, Maine; Takoma Park, Maryland; the cities of St. Paul and Minneapolis in Minnesota; Cambridge, Massachusetts.

Voting Methods Working Group (VMWG) , NIST Interoperability Working Group Modeling Subgroup for Multi-round Voting Tabulation Methods

Authors: H. Deutsch, G. Gilbert, D. Cary, C. Hughes, L. Lochridge (Author & Editor)
09/08/2017   DRAFT

The state of Maine has adopted RCV and is planning an implementation for 2018. San Francisco is implementing a new EMS for existing use of RCV since 2005.

At the time of writing two jurisdictions in the United States that use RCV for multi-seat contests are Minneapolis, Minnesota for some city boards and Cambridge, Massachusetts for both City Council and School Committee.

RCV Tabulation Scenarios

This RCV tabulation process model is general and flexible enough to be applied to tabulation scenarios for a single or centralized RCV tabulation, and for aggregations. This process model provides specification of the basic or most common procedural steps, in practice, for an RCV tabulation.

This standard specification is aligned with the Elections Business Process Models specification and the expected level of generality and flexibility that is common to the other standards and specifications in the ecosystem.

This standard specification and this process model are not intended to provide detailed, fine grain, systems engineering level specification of all possible configuration scenarios, implementation variants, or edge case scenarios.

Common election contest features are intended to be configured, consistent with the common contest configuration business rules elements of the parent "Tally" process of the Tabulation element of the Election Business Process Model framework[]. Specialized modules may be provided by the adopter of this specification, consistent with that framework.

Examples of configurable local business rules or modules include low level implementation of optical ballot mark adjudication sub-processes or procedure, or specialized encryption modules.

The standard provides for the scenario in which summaries are input as a Cast Vote Record Report (CVRR) of the CVR CDF and output as a CVRR CDF or as an Election Results Reporting CDF.

Distributed tabulations, involving aggregation of polling place tabulators and centralized tabulators are supported.

This process model assumes an automated, or computerized implementation and is not intended to provide for human actors implementing hand counts, or other human actor implemented subprocesses. It can be adapted to and contains the essential processing needed for a manual tabulation of paper

ballots or paper CVRs. However it does not attempt to describe those adaptations or specialized procedures.

Resources

The following resources must be available as a precondition or as input to this process model to complete a tabulation of an RCV contest:

- A vote selection data set of record in the form of a Cast Vote Record Report (CVRR) or set of CVRs Common Data Format (CDF).

- A set of tabulation configuration parameters that indicate information about the contest, the candidates, and which specific options should be used during the tabulation to conform to a particular jurisdictions variant of RCV.

- A set of optional configuration parameters that specify user supplied, implementation specific business rules. For example, adjudication of ballot marks, encryption or other special security policies, specialized tie breaking rules, or other special locally required modules.

The reference implementation of this standard specification adopts the NIST Cast Vote Record (CVR) specification (https://github.com/usnistgov/CastVoteRecords) as the vote selection data set Common Data Format (CDF), for the purpose of Data Interoperability.
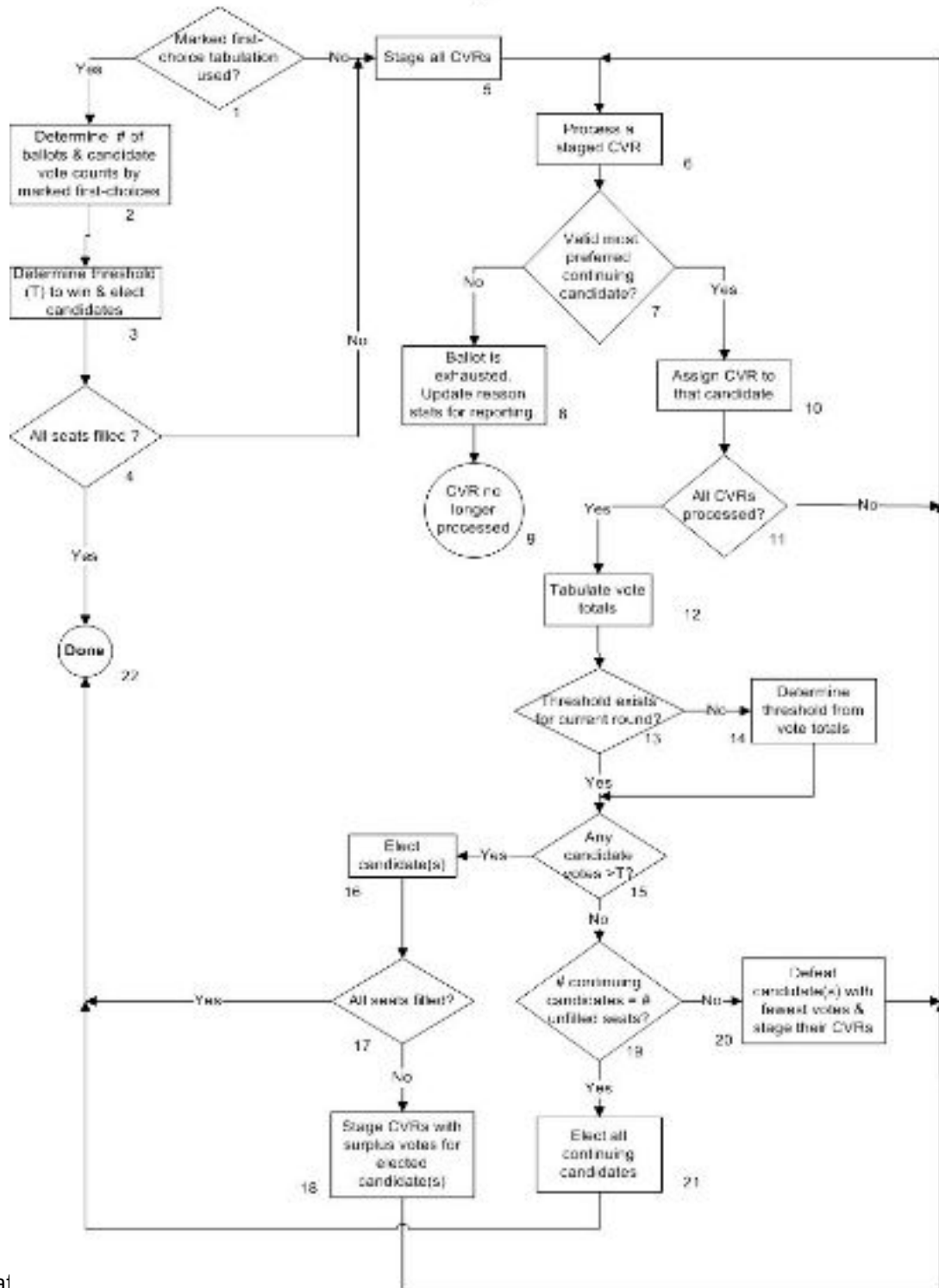
For the purpose of this standard, one or more Cast Vote Record Reports or sets of Cast Vote Records are assumed to be the vote selection data sets of record, irrespective of what medium or voting equipment was provided to voters to produce their marked ballots.

Nothing about this RCV process flow model is intended to restrict the adopter of this standard as to the internal equipment format of a system that may produce vote selection data sets. As a Common Data Format, the NIST Cast Vote Record specification (CVR CDF) is general and flexible enough to apply to a the wide variety of voting methods, election instantiations, and use cases. The CVR specification provides XML and JSON formats to support translation into the CVR CDF.

This RCV model specification supports variants of RCV implementation or any multi-round voting method instance where it is necessary to store the state of the CVRR or CVR set, per round. In that instance, the output from this RCV model is composed of a CVRR representing the outcome of the tabulation process and also the corresponding sets of CVRs in sequential order for each round that was executed.

Voting Methods Working Group (VMWG) , NIST Interoperability Working Group
Modeling Subgroup for Multi-round Voting Tabulation Methods
Authors: H. Deutsch, G. Gilbert, D. Cary, C. Hughes, L. Lochridge (Author & Editor)
09/08/2017   DRAFT

The remainder of this section illustrates the descriptive process flow and provides details of the generic RCV process and its common variants that are in use in one or more jurisdictions in the United States.

RCV Flow Single & Multi-Seat Contests

[this draf

Voting Methods Working Group (VMWG) , NIST Interoperability Working Group
Modeling Subgroup for Multi-round Voting Tabulation Methods
Authors: H. Deutsch, G. Gilbert, D. Cary, C. Hughes, L. Lochridge (Author & Editor)
09/08/2017   DRAFT

1. **Marked first-choice tabulation used?**

This decision step determines whether to perform a tabulation of marked first choices which might fill all seats and avoid a tabulation that considers marked second and later choices.

Flows. This step is the entry point for the RCV tabulation process; there are no other processing control flows into this step. The Yes path to Step 2 (Determine # of ballots & candidate vote counts by marked first-choices) is taken if an initial tabulation of marked first choices is used. The No path to Step 5 (Stage all CVRs) is taken if the CVRs are used to initially determine the effective first-choice candidate votes.

See also. The description for Step 6 (Process a staged CVR) has additional detail about how effective first choices are determined.

Resources. VM.RCV.Tabulation.Configure.MarkedFirstChoices.DoTabulation, an indication of whether a tabulation of marked first choices should be done to possibly determine which candidates are elected..

 Explanation. In some instances of an election contest, the tally of marked first choices determines the outcome of the contest. Whether to decide who is elected based on a tally of marked first choices can be a matter of law or election official choice. A tally of marked first choices may be simpler to accomplish than a tally of effective first choices, since determining an effective first choice may require examining second and later marked choices.

Variants. Some RCV variants may require a marked first-choice tabulation subprocess, allow it as an option, or perform it as a matter of practice.

If a ballot has a validly marked first choice, that is also the ballot's effective first choice. However for some variants, a ballot can have an effective first choice but not a marked first choice if the first choice is left blank or is not valid but there is a second or subsequent choice that is validly marked. See the description for Step 6 (Process a staged CVR) for additional details about how effective first choices are determined.

2. **Determine # of ballots & candidate vote counts by marked first-choices**

- This process step is the first step in the Marked First-choice Tabulation subprocess, it determines two count values: BallotCount and VoteCount.
- This process step is initiated when the jurisdiction's RCV variant implementation is configured for the Marked First-choice Tabulation subprocess .

- This process step is exited if one the following conditions are met:
  - VM.RCV.Tabulation.BallotCount and VM.RCV.Tabulation.VoteCount have been determined
  - Adjudication Event Detected
  - Error

- Associated resource/data/etc. required
  - Data input required for this process step is the Cast Vote Record (CVR) data set for this particular tabulation. Within the CVR for this tabulation, the following CVR Objects, Fields and Attributes groupings are required:
    - CVR.BallotSelectionVote
    - **...**
  - Results reports from each initial tabulation source (precinct, central, other)

- Pointer, if necessary for an understanding of this element, to another associated part of the diagram or functional information in this description.
- Further explanation as necessary to understand and use this flow diagram / model corresponding to this RCV process flow step
-

This process step is the first step of the Marked First-choice Tabulation subprocess; it tabulates votes by considering only marked first-choice ballot selections as if the contest were a vote-for-one contest; it also counts the number of ballots that comprise the threshold base.

Flows. Processing control flows in from the Yes branch from Step 1 (Marked first-choice tabulation used?) and out to Step 3 (Determine threshold (T) to win & elect candidates).

See also. None.

Resources.

- The CVRs with ballot selections to be tabulated.
- VM.RCV.Tabulation.Configure.MarkedFirstChoices.ThresholdBaseType, an indication of which ballots are counted as part of the threshold base for a tabulation of marked first choices.

Explanation. For a each CVR, if the first choice is not marked or is overvoted, the CVR will not count for any candidate, but might be counted as an undervote or overvote. If the CVR records a

validly marked first choice selection for a single candidate, the CVR contributes a single vote for that candidate.

The number of ballots used as the base for the threshold calculation in Step 3 (Determine threshold (T) and elect candidates) is also determined.

Variants. The number of ballots counted in the threshold base can be:

- the total number of cast ballots.
- the total number of ballots with validly marked first choices that count for candidates.
- the number of ballots cast except for those with a marked overvote for first choice.

3. **Determine threshold (T) to win and elect candidates**

This process step is the second step in the Marked First-choice Tabulation subprocess; it determines the threshold used to identify how many votes a candidate must have in order to be elected by the tabulation of marked first choices and then elects any candidates that have at least that many votes.

Flows. Processing control flows in from Step 2 (Determine # of ballots & candidate vote counts by marked first-choices) and out to Step 4 (All seats filled?).

See also. Compare to related calculations for a full RCV tabulation in block 14 (Determine threshold from vote totals) and block 15 (Any candidate votes > T?). Those calculations cover more variations and the possibility of fractional votes.

Resources.
- VM.RCV.NumberSeatsToFill, the number of seats to be filled by this contest;
- VM.RCV.Tabulation.MarkedFirstChoices.ThresholdBase, the number of ballots comprising the threshold base;
- VM.RCV.Tabulation.MarkedFirstChoices.VoteCount, the number of marked first-choice votes counting for each candidate;

Explanation. Although there exist possible notational variants in legislation, local rules or guidelines, the threshold formula for any number of candidates to be elected is:
$$T = 1 + B \ / \ (N+1)$$

rounded down to the nearest whole number, where T is the threshold, B is the number of ballots in the threshold base, and N is the number of candidates to be elected. The threshold criterion requires that a candidate must have at least T votes to be elected.

For example, in a contest where only one candidate is to be elected and the threshold base is 607 votes, the threshold is 304 votes [floor(1 + 607/2)], and a candidate must have at least that many marked first-choice votes to be elected.

In a contest with 4 candidates to be elected where the threshold base is 607 votes, the threshold would be 122 votes [floor(1 + B/5)], and a candidate must have at least that many marked first-choice votes to be elected, i.e. slightly more than 20%.

Variants. Equivalent to the above formula and criterion, Oakland defines its threshold, applicable only to single-seat contests, in terms of a majority of the threshold base, i.e. more than 50% of the threshold base.

4. **All seats filled?**
This decision step is the third and last step in the Marked First-choice Tabulation subprocess; it will determine whether all seats have been filled by that subprocess.

Flows. Processing control flows in from Step 3 (Determine threshold (T) to win & elect candidates) and out along one of two paths. The Yes path to Step 22 (Done) is taken if all seats are filled or if there are not more eligible candidates than seats. The No path to Step 5 (Stage all CVRs) is taken if one or more seats have not been filled due to insufficient candidates obtaining the required number of marked first-choice votes and there are more eligible candidates than seats. In a single-seat contest, electing one candidate is sufficient for taking the Yes path..

See also. A similar decision is made in Step 17 (All seats filled?).

Resources.
- The number of candidates elected by the Marked First-choice Tabulation subprocess.
- The number of candidates eligible to be elected.
- VM.RCV.NumberSeatsToFill, the number of seats to be filled.

Explanation. If the No path is taken, a tabulation is initiated and the first round of that tabulation is conducted irrespective of results of the tabulation of marked first choices (See Step 12 Variants). In particular, candidates elected by marked first choices are considered unelected continuing candidates at the beginning of the tabulation. If the Yes path is taken, the second tabulation is not required.

Variants. None.

5. **Stage all CVRs**
   This process step stages all CVRs for the initial assignment to the candidate, if any, that each CVR will count for in the first round.

   Flows. Process control flows in from the No path from Step 1 (Marked first-choice tabulation used?) and the No branch from Step 4 (All seats filled?) and out to Step 6(Process a staged CVR).

   See also. CVRs are also staged for reassignment (a.k.a. transfer) in Step 18 (Stage CVRs with surplus votes for elected candidate(s)) and Step 20 (Defeat candidate(s) with fewest votes and stage their CVRs).

   Resources. All CVRs to be tabulated.

   Explanation. Staging a CVR means to identify the CVR as needing assignment to a candidate by the Assignment subprocess of Step 6 (Process a staged CVR) through Step 11 (All CVRs processed). In this step, all CVRs are staged for assignment in preparation for the first-round vote tallies.

   Starting at this step, the tabulation proceeds without any reliance on any results of a preceding tabulation of marked first choices. In particular, all candidates are now treated as continuing candidates, neither elected nor eliminated, without any votes.

   Variants. Minneapolis is the only variant for multi-seat contests that requires a tabulation of marked first choices. For that variant, It is possible that a tabulation of marked first choices will elect at least one candidate, but will not elect enough candidates to be decisive for the whole contest. In that case, as the tabulation proceeds from this step, the any elected status from the tabulation of marked first choices is ignored. However, the Minneapolis rules are structured so that any candidate elected by the tabulation of marked first choices, will also reach the threshold in the first round and will still be elected by the remainder of the tabulation.

   For the Cambridge multi-seat variant, CVRs start with a recorded ordering. CVR's are initially processed in that order and the subsequent processing order depends in part on the initial ordering. The same initial ordering is used across different tabulations, including recounts. CVRs are initially ordered based on a random ordering of precincts, and within precincts they are given a fixed, but unspecified ordering.

   For other variants, the tabulation outcome, which candidates are elected, does not mathematically depend on the order in which CVRs are processed. However practical

implementations of some variants may produce numerical results that are dependent on the order in which CVRs are processed due to rounding of fractional values.

6. **Process a staged CVR**

This process step removes a single CVR from the collection of staged CVRs and identifies the candidate for which the CVR will next count, i.e. the CVR's highest ranked continuing candidate, if such a candidate exists.

Flows. CVRs processed in this step have been staged for assignment / reassignment from one of three sources: 1) from Step 5 (Stage all CVRs), where all CVRs are staged for the first round, 2) from Step 20 (Defeat candidate(s) with fewest votes and stage their CVRs) after a candidate has been defeated, and 3) from Step 18 (Stage CVRs with surplus votes for elected candidate(s)), after a candidate has been elected with surplus votes in a multi-seat contest. In addition, process control flows in from Step 11 (All CVRs processed?). Process control flows out of this step to decision Step 7 (Valid most preferred continuing candidate?).

See also. The description of Step 18 (Stage CVRs with surplus votes for elected candidate(s)) has further details about the two methods used to redistribute surplus votes:
   ● as a fractional vote per CVR, but for all CVRs that counted for the elected candidate.
   ● as a whole vote per CVR but usually only for a subset of the candidate's CVRs.

Step 2 (Determine threshold (T) to win and elect candidates) provides the corresponding determination of which, if any, candidate a CVR counts for during the Marked First-choice Tabulation subprocess.

Resources.
   ● The collection of CVRs staged but not yet processed for assignment.
   ● The tabulation configuration parameters that specify how the most preferred continuing candidate is determined:
      ○ VM.RCV.Tabulation.Configure.AllowSkippedRankings, an indication of whether or under what circumstances a ranking level may be left blank without causing less preferred ranking levels to be ignored.
      ○ VM.RCV.Tabulation.Configure.AllowOvervotedRankings, an indication of whether or under what circumstances a ranking level may rank more than one candidate without causing that ranking level or less preferred ranking levels to be ignored.
      ○ VM.RCV.Tabulation.Configure.AllowRepeatRankings, an indication of whether or under what circumstances a ranking level may rank a candidate who has also been ranked at a more preferred ranking level without causing the repeating ranking level or its less preferred ranking levels to be ignored.

- The set of continuing candidates; alternatively, VM.RCV.Tabulation.CandidateStatus for every candidate, the status of whether a candidate is a defeated, elected, or continuing candidate.

Explanation. A continuing candidate is a candidate that is not yet defeated nor elected. A CVR's most preferred continuing candidate is also known as the highest ranked continuing candidate. When determined to decide the first-round vote totals, the most preferred continuing candidate is also known as the effective first-choice candidate.

Determining the most preferred continuing candidate can be fairly straight forward if the voter has simply ranked candidates in a strict order of preference. If any continuing candidates have been ranked, the most preferred continuing candidate is the candidate out of all ranked continuing candidates with the lowest numerical ranking number. If no continuing candidate has been ranked on the CVR, the CVR does not have a most preferred continuing candidate. However there are several irregular ranking situations that a voter might mark and which are treated according to variant-specific rules. Those situations include:
- No candidate is ranked at a ranking level, but one is ranked at a less preferred ranking level.
- More than one candidate is ranked at a ranking level.
- A candidate is ranked at more than one ranking level.

Variants. The following describes how irregular ranking situations are treated by variants:

- Unvoted choice (no candidate ranked at a ranking level, but some at subsequent levels)
  - CVR considered exhausted
  - CVR considered exhausted if there are two unvoted choices in succession
  - Skipped and subsequent choice processed, if any
- Overvoted choice (more than one candidate ranked at a ranking level)
  - CVR considered exhausted
  - Not considered overvoted if doesn't contain more than one continuing candidate
  - Skipped and subsequent choice processed, if any
- Repeated ranking of a previously ranked candidate (who was elected or eliminated)
  - CVR considered exhausted
  - Skipped and subsequent choice processed, if any

All variants detect these irregular ranking situations incrementally, as if examining ballot selections in order of increasing numerical ranking level, and stopping once a most preferred continuing candidate is found.

If a CVR is considered exhausted, that also means that the CVR is considered to not have a most preferred continuing candidate.

[this draft is based upon : hd-7/26/2017]   12

7. **Valid most preferred continuing candidate?**
This decision step branches depending on whether the previous Step 6 (Process a staged CVR) identified a most preferred continuing candidate for the CVR.

Flows. Process control flows in from Step 6 (Process a staged CVR). The Yes path to Step 10 (Assign CVR to that candidate) is taken if the CVR has a valid selection for a most preferred continuing candidate. The No path to Step 8 (Ballot is exhausted. Update reason stats for reporting.) is taken if the CVR does not have a valid selection for a most preferred continuing candidate.

See also. None.

Resources. Whether or not a most preferred continuing candidate was found in the previous Step 6 (Process a staged CVR).

Explanation. No additional explanation.

Variants. None.

8. **Ballot is exhausted. Update reason stats for reporting**
This processing step records, for the purposes of reporting round results, the reason for the ballot being considered exhausted, based on the circumstances of not finding a most preferred continuing candidate in Step 6 (Process a staged CVR).

Flows. Process control flows in from Step 7 (Valid most preferred continuing candidate?) and out to Step 9 (CVR no longer processed).

See also. Step 6 (Process a staged CVR) describes situations in which a most preferred continuing candidate would not be found. Step 18 (Stage CVRs with surplus votes for elected candidate(s)) describes additional details for how the Cambridge variant handles the transfer of surplus from an elected candidate.

Resources. Depending on the variant:
- The tabulation configuration parameters that indicate which categories are supported and the rules for categorizing an exhausted ballot.
- Data that may be used to categorize the exhausted CVR, possibly including:
  - The situation encountered in Step 6 (Process a staged CVR) that prevented a most preferred continuing candidate from being identified.
  - The CVR that was exhausted.
  - VM.RCV.Tabulation.ReportingRound, the number of the round for which the exhausted ballot will be first reported.

○ VM.RCV.Tabulation.Configure.MaxNumberRankings, the maximum number of candidates that the voter was allowed to rank.

Explanation. For each round, the cumulative number of votes associated with exhausted ballots may be reported in one or more categories, which are variant or implementation specific. This is effectively an assignment of the CVR to one of the reporting categories.

Variants. The cumulative number of exhausted votes might not be reported at all or might be reported in a single, undifferentiated category. The categories of exhausted ballots can include one or more of the following: undervotes (first round only), overvotes, abstentions, and a category for exhausted votes not otherwise categorized. Some implementations use the term "exhausted" to refer only to ballots and votes that are categorized in that last category.

The Cambridge variant does not treat ballots as exhausted when they have been staged from Step 18 (Stage CVRs with surplus votes for elected candidate(s)). Instead, a CVR that does not have most preferred continuing candidate returned back to count for the elected candidate from which it was staged. The Cambridge variant will extend its staging and reassignment of CVRs, one CVR at a time, until the full surplus has been transferred to other candidates or until a reassignment to other candidates has been attempted for all CVRs assigned to the elected candidate.

9. **CVR no longer processed**
   This processing step explicitly marks the end of tabulation processing for an exhausted ballot.

   Flows. Process control flows in from Step 8 (Ballot is exhausted. Update reason stats for reporting.) and out to Step 11 (All CVRs processed?) [the outbound flow is not currently shown in the diagram], even though there is no further processing of this CVR.

   See also. None.

   Resources. None.

   Explanation. This step reflects that once a ballot becomes exhausted, it can not be revived and later be assigned to count for a candidate.

   Variants. None.

10. **Assign CVR to that candidate**
    This process step assigns the CVR to count for its most preferred (highest ranked) continuing candidate, as identified in Step 6 (Process a staged CVR).

Flows. Process control flows in from the Yes path from Step 7 (Valid most preferred continuing candidate?) and out to Step 11 (All CVRs processed?).

See also. None.

Resources.
- The CVR that will be assigned.
- The CVR's most preferred continuing candidate.
- The collection of CVRs already assigned to the CVR's most preferred continuing candidate.

Explanation. The CVR is added to the collection of CVRs that are assigned to the most preferred candidate.

Variants. For the Cambridge variant, where the order processing CVRs is important, the CVR is appended to a sequence of the CVRs that are already assigned to the most preferred candidate.

The CVR will contribute one full vote or a fraction of a vote to that candidate's vote total. The CVR will contribute only a fraction of a vote only for certain variants for multi-seat contests, in which case, the fraction of a vote represents the CVR's share of surplus transferred from an elected candidate.

For the Cambridge variant, a candidate is allowed to accumulate a surplus of votes only for the first round tally of candidate vote totals. After that, the assignment of CVRs in this step keeps a running total of votes counting for each continuing candidate, a total which is updated after each assignment of a CVR to a candidate. If the candidate reaches the threshold as a result of an assignment, the candidate is immediately considered elected, is not eligible to be assigned additional CVRs, and so never accumulates surplus. Instead, any subsequent ballots that would have otherwise contributed to a surplus are assigned to other candidates or are exhausted.

11. **All CVRs processed?**
This decision step checks whether all CVRs staged for assignment to a candidate have been processed.

Flows. Process control flows in from Step 10 (Assign CVR to that candidate).   The Yes path to Step 12 (Tabulate vote totals) is taken if all CVRs staged for assignment to a candidate have been processed. The No path to Step 6 (Process a staged CVR) is taken if some staged CVRs still remain to be processed.

See also. Step 10 (Assign CVR to that candidate) and Step 18 (Stage CVRs with surplus votes for elected candidate(s)) have additional descriptions of how surplus transfers are processed for the Cambridge variant.

Resources. The number of CVRs that have been staged for assignment but which have not yet been processed for assignment; alternatively, the collection of such CVRs.

Explanation. Reaching this step marks the end of processing for a specific CVR regarding its assignment within the current round or at termination of all processing of the CVR set, as the case may be.

Variants. The Cambridge variant at least sometimes stages and processes for assignment CVRs one at a time when they represent surplus transfers. In that case there may not be any staged but unprocessed CVRs, but this decision point will stage and process for reassignment an additional CVR if the elected candidate's surplus has not yet been transferred to other candidates, provided there are still CVRs for which have not yet been staged for attempted reassignment.

12. **Tabulate vote totals**
    This processing step tallies the round's vote totals for each candidate and for any other reporting categories.

    Flows. Processing control flows in from the Yes branch from Step 11 (All CVRs processed?) and out to Step 13 (Threshold exists for current round?).

    See also. Step 10 (Assign CVR to that candidate) and Step 18 (Stage CVRs with surplus votes for elected candidate(s)).

    Resources.
    - VM.RCV.Tabulation.VoteCount, the most recent vote totals from Step 12 (Tabulate vote totals) or zeroed vote totals if this is for the first round.
    - Sets of CVRs that were assigned to candidates or to a reporting category for exhausted CVRs since the previous round's vote totals or, if this is for the first round, since the staging of all CVRs in Step 5 (Stage all CVRs).
    - New vote totals for any candidates from which surplus votes have been transferred since the previous round's vote totals.

    Explanation. Vote totals for the round are produced by summing the votes by assigned candidate for all CVRs or by adding the vote contributions of newly reassigned CVRs to the vote totals produced by the previous invocation of this step. Also, the CVRs' vote contributions (for defeated candidates) or partial vote contributions (for elected candidates with surplus) are subtracted from the previously assigned candidate, if any.

    Equivalently, these vote totals may be updated incrementally, performing subtractions as CVR's assigned to a candidate are staged for reassignment in Step 18 (Stage CVRs with surplus votes

for elected candidate(s)) and Step 20 () and performing additions as CVR's are assigned to candidates in Step 10 (Defeat candidate(s) with fewest votes & stage their CVRs) or to other reporting categories in Step 8 (Ballot is exhausted. Update reason stats for reporting.).

Several ordinances specify that all of the continuing ballots for all continuing candidates shall be counted again in a new round, or the equivalent.

Variants. If there was a tabulation of marked first choices in block 2, the candidate vote totals for the first round can be higher, but never lower, than the candidate totals of marked first choices. A candidate vote total can be higher if there are one or more ballots with no marked first choice, or in some jurisdictions an invalidly marked first choice, but there is a valid candidate selection for a subsequent choice.

For variants that do not use fractional votes, these vote totals for candidates and categories of exhausted votes can also be determined by totalling the current vote contribution of every CVR, not just those recently assigned or reassigned, to the vote total for the candidate or reporting category to which the CVR is currently assigned.

For multi-seat variants using fractional surplus votes, candidates that were elected, either without surplus or that have had surplus transferred to other candidates, such candidates must also be assigned a vote total equal to the threshold. For some of these variants, this may only produce an approximation of their recorded vote totals, due to rounding in surplus transfer calculations. For other variants, that rounding may be accounted for in a reporting category of residual surplus votes.

13. **Threshold exists for current round?**
    This decision step determines whether a previously calculated threshold value will continue to be used or whether instead a new threshold value will be calculated.

    Flows. Process controls flows in from Step 12 (Tabulate vote totals).   The Yes path to Step 15 (Any candidate votes > T?) is taken if there is a current threshold value that will continue to be used. The No path to Step 14 (Determine threshold from vote totals) will be taken if a new threshold value will be calculated.

    See also. None.

    Resources.
    ● VM.RCV.Tabulation.Configure.RecalculateThreshold, an indication of whether the threshold will be calculated only in the first round and reused rather than recalculated for subsequent rounds, or whether the threshold will be calculated for every round.
    ● VM.RCV.Tabulation.ReportingRound, the number of the round for which candidate vote totals have most recently been calculated.

Explanation. The No path will be taken to calculate a new threshold value if and only if the tabulation configuration parameter indicates that the threshold will be calculated every round or if the number of the round is 1. Typically, the tabulation configuration parameter indicates that the threshold will be calculated for every round for single-seat contests but only for the first round for multi-seat contests. In that case, the Yes path is only taken for the second and subsequent rounds of a multi-seat contest, which reuses the first-round threshold.

Variants. The St. Paul variant, which is only for single-seat contests, is an exception that uses for all rounds the threshold calculated for the first round.

14. **Determine threshold from vote totals**
    This processing step determines the threshold base and the threshold used to form a sufficient criterion for electing a candidate based on that candidate's vote total.

    Flows. Process control flows in from the No path from Step 13 (Threshold exists for current round) and out to Step 15 (Any candidate votes > T?).

    See also. Step 3 (Determine threshold (T) to win & elect candidates) performs a corresponding calculation of threshold for a tabulation of marked first choices.

    Resources.
    - One of:
        - VM.RCV.Tabulation.VoteCount for each candidate, the most recent vote totals for candidates from Step 12 (Tabulate vote totals).
        - VM.RCV.Tabulation.NumberTabulatedCvrs, the total number of CVRs being tabulated.
    - Tabulation configuration parameters that indicate:
        - VM.RCV.Tabulation.Configure.ThresholdBaseType, an indication of which threshold base to use: the total votes for candidates or the total number of CVRs being tabulated.
        - VM.RCV.Tabulation.Configure.ThresholdRoundingPrecision, the number of decimal places to which the threshold should be rounded.
        - VM.RCV.Tabulation.Configure.ThresholdRoundingDirection, the direction that the threshold value should be rounded, up or down.
        - VM.RCV.Tabulation.Configure.ThresholdComparisonType, the type of threshold criterion comparison to use: must exceed the threshold (>) or must reach or exceed the threshold (≥).
        - VM.RCV.NumberSeatsToFill, the number of seats this contest may fill.

    Explanation. The threshold is a number of votes, that determines, together with the threshold comparison type, how many votes are sufficient for a candidate to be elected. It is also the

point at which additional votes for a candidate are considered to be surplus votes for the candidate.

If the threshold base is the total number of votes counting for candidates, that total must first be calculated from the vote totals for individual candidates. Otherwise, the total number of tabulated CVRs is used as the threshold base.

A formula for the threshold can be expressed as:

$$T = B / (S + 1) + X$$

where T is the threshold, B is the threshold base, S is the number of seats to be filled, and X is some small extra amount between zero and one.

There are two approaches for matching the threshold comparison to the allowed values of X in order to ensure that it is mathematically impossible to elect too many candidates:
- X must be greater than zero, but reaching the threshold (≥) is sufficient to be elected
- X can be zero, but the threshold must be exceeded (>) in order to be elected

Variants. The more traditional and common method of determining the threshold is to use a threshold base of votes counting for candidates, a ≥ threshold comparison, and the specialized formula:

$$T = floor(1 + B / (S + 1))$$

Using this formula, X is always greater than zero and less than or equal to one.

This approach works for most single-seat thresholds, even though their threshold criterion is typically defined in terms of a majority of votes counting for continuing candidates, i.e. > 50%, to the extent that single seat votes are only counted as whole votes.

St. Paul, which only uses single-seat contests, uses the total number of tabulated CVRs as its threshold base.

Some recent variants for multi-seat contests use the second approach above and require a > threshold comparison. For example, SB 1288 uses a threshold base of the the total number of votes for candidates, but a threshold formula of:

$$T = round\_up(B / (S + 1), 5)$$

which rounds the ratio B / (S + 1) up to five decimal places, the precision for calculating fractional votes. Using this formula, X can be zero but is always less than 0.00001.

In the diagram, the threshold criterion is expressed in terms of the second approach with the understanding that the greater-than-or-equal-to criterion of the first approach can be substituted as appropriate.

15. **Any candidate votes >T?**
   This decision step determines whether candidates are elected as a result of having enough votes to satisfy the threshold criterion and there not being any special conditions that require that first defeating candidates takes priority over electing candidates.

   Flows. Process control flows in from the Yes path from Step 13 (Threshold exists for current round?) and from Step 14 (Determine threshold from vote totals). The outbound Yes path to Step 16 (Elect candidates) will be taken if no special defeat criteria are satisfied and either the number of continuing candidates is not greater than the number of unfilled seats or one or more continuing candidates satisfy the threshold criterion. The outbound No path to Step 19 () is taken if special defeat criteria are satisfied or both the number of continuing candidates is greater than the number of unfilled seats and there are no candidates that satisfy the threshold criterion.

   See also. Step 3 (Determine threshold (T) to win & elect candidates) performs a corresponding determination of whether there are candidates that can be elected for a tabulation of marked first choices.

   Resources.
   - VM.RCV.Tabulation.VoteCount for all candidates, the most recent vote totals from Step 12 (Tabulate vote totals).
   - VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.
   - VM.RCV.NumberSeatsToFill, the number of seats to be filled.
   - VM.RCV.Tabulation.Threshold, the threshold value.
   - VM.RCV.Tabulation.Configure.ThresholdComparisonType, the type of threshold criterion comparison to use: must exceed the threshold (>) or must reach or exceed the threshold ($\geq$).
   - Other variant-specific information needed to determine whether defeating candidates should take priority in this round, such as:
     - VM.RCV.Tabulation.IsCandidateUndeclaredWriteIn for all candidates, an indication of whether a candidate is an undeclared write-in candidate.
     - VM.RCV.Tabulation.TotalBallotSelections for each candidate, the total number of (valid, i.e. not overvote?) ballot selections across all ranking levels for each candidate.

   Explanation. The core decision is whether any continuing candidate has a vote total that satisfies the threshold criterion, that is, the vote total > T or vote total $\geq$ T, depending on the type of threshold comparison.

However some variants may make defeating candidates a priority, even if there are continuing candidates that satisfy the threshold criterion.

The test for whether there are more continuing candidates than unfilled seats should only need to be checked when checking first-round vote totals.

Variants. Minneapolis has a rule for its multi-seat contests that requires, subject to defined conditions, that the No path to be taken in order to eliminate one or more candidates, even if one or more continuing candidates has enough votes to satisfy the threshold criterion.

In particular, if there are not enough candidates satisfying the threshold criterion (votes ≥ the threshold) to fill all of the seats, and if there are any undeclared write-in candidates or any candidates that are "mathematically impossible to be elected", as defined, then defeating those candidates takes priority over electing any continuing candidates that do satisfy the threshold criterion.

CA SB 1288 has a default provision and HR 3057 requires that for single-seat contests, the No path be taken for single-seat contests as long as there are three or more continuing candidates. This can extend the tabulation to show a one-on-one comparison between the two finalists without changing which candidate is elected. San Francisco has adopted this option in practice.

16. **Elect candidate(s)**
This processing step elects one or more of the continuing candidates with a vote total that satisfies the threshold criterion or elects all continuing candidates if their number does not exceed the number of unfilled seats.

Flows. Processing control flows in from Step 15 (Any candidate votes > 15) and out to Step 17 (All seats filled?).

See also. Step 3 (Determine threshold (T) to win & elect candidates) performs a corresponding function of electing candidates for the Marked First-choice Tabulation subprocess.

Resources.
- VM.RCV.Tabulation.VoteCount for all candidates, the most recent vote totals from Step 12 (Tabulate vote totals).
- VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.
- VM.RCV.Tabulation.Configure.RestrictNumberElected, an indication of whether at most one candidate can be elected if all seats can not be immediately filled.
- VM.RCV.Tabulation.Configure.ResolveSurplusTies, an indication of how ties for having the most surplus are resolved.

Explanation. At least one continuing candidate that satisfies the threshold criterion is elected. If the number of continuing candidates is less than or equal to the number of unfilled seats, all of them are elected.

Variants. For multi-seat contests, there can be more than one continuing candidate that satisfies the threshold criterion. If the number of such candidates is not enough to fill all remaining seats, a variant may only allow electing one of those candidates. Otherwise, all continuing candidates that satisfy the threshold criterion are elected. If the restriction of electing only one candidate applies, the candidate with the largest surplus is chosen to be elected. Variant-specific rules for resolving a tie for having the largest surplus may apply. A candidate that satisfies the threshold criterion but is not elected remains a continuing candidate and is still eligible to receive transferred votes from other candidates.

17. **All seats filled?**
This decision step checks whether all seats have been filled by elected candidates.

Flows. Process control flows in from Step 16 (Elect candidate(s)). The Yes path to Step 22 (Done) is taken if enough candidates have been elected to fill all seats. The No path to Step 18 (Stage CVRs with surplus votes for winning candidates) is taken if some seats remain unfilled, which can only happen for a multi-seat contest.

See also. Step 4 (All seats filled?) checks the corresponding condition for a Marked First-choice Tabulation subprocess.

Resources. VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.

Explanation. The Yes branch will always be taken for a single-seat election, since before reaching this step, at least one candidate has been elected.

Variants. None.

18. **Stage CVRs with surplus votes for winning candidates**
This process step stages for reassignment to other candidates selected CVRs that are associated with surplus votes for any candidate that was elected since the most recent tabulation of vote totals in Step 12 (Tabulate vote totals)).

Flows. Process control flows in from the No path from Step 17 (All seats filled?) and flows out to Step 6 (Process a staged CVR).

See also. Step 5 (Stage all CVRs) and Step 20 (Defeat candidates with fewest votes and stage their CVRs) also stage CVRs for assignment to a candidate. For the Cambridge variant,

processing between this step and Steps 6 - 11 dealing with reassignment of CVRs are more interactive in their flow of control than is depicted in the diagram.

Resources.
- For each candidate that was elected since the most recent tabulation of vote total in Step 12:
    - The collection of CVRs currently assigned to that candidate.
    - VM.RCV.Tabulation.VoteCount, the vote total for that candidate.
- VM.RCV.Tabulation.Threshold, the threshold for being elected.
- VM.RCV.Tabulation.SurplusTransferMethod, an indication of whether surplus votes may be transferred as a fraction of a vote per CVR and if so, the details of how to calculate fractional surplus votes, but if not, the details of how CVRs are selected and transferred each with a whole surplus vote.
- If fractional surplus votes are used, for each CVR that will be staged:
    - VM.RCV.Tabulation.CvrVoteValue, the amount of a vote that the CVR currently contributes to the vote total for candidate to which the CVR is assigned.

Explanation. .This step will only occur in a multi-seat contest before all seats have been filled by elected candidates. For each qualifying candidate, some or all of the CVRs assigned to the candidate are staged for reassignment.

Variants. There are two general methods currently used for handling surplus votes (votes for an elected candidate that are in excess of the threshold):

Transfer fractional surplus votes. All CVRs for a candidate are staged and each CVR is given a transfer vote value that is a fraction of a whole vote and which corresponds to the CVR's share of the elected candidate's surplus. Typically, a surplus fraction is first calculated for the elected candidate that is equal to the ratio of the elected candidate's surplus votes divided by the elected candidate's total votes. The new transfer vote value for a CVR is calculated by multiplying the surplus fraction by the CVR's previous transfer vote value. CVRs start the tabulation process with a transfer vote value of one whole vote. Variant-specific rules may specify the precision and any rounding (typically rounding down) that is associated with these arithmetic operations.

Transfer whole surplus votes. This is the method that Cambridge uses. Each selected CVR is staged counting as a whole vote, but initially only every Nth CVR is staged. The value for N is calculated as the total votes for the elected candidate divided by the number of surplus votes.This method depends on processing the CVRs assigned to a candidate in the same order they were assigned to the candidate.

In practice, Cambridge processes CVR's one at a time to be staged, find the most preferred candidate, and assign the CVR to that most preferred continuing candidate, if one exists, all

before similar processing is started for the next CVR. If a staged CVR would otherwise become exhausted, the CVR is returned to count for the elected candidate. This incremental processing continues until the number of CVRs successfully transferred to other candidates is equal to the elected candidate's surplus, or until all of the elected candidate's CVRs have been subjected to an attempted transfer. If a pass through all of a candidate's CVR's while attempting to stage every Nth ballot is not sufficient to transfer the candidate's surplus, another pass that attempts to stage every Nth + 1 ballot is made, then another pass for every Nth + 2 CVR, and so on until the surplus has been transferred or until all CVR's have been subjected to an attempted transfer.

[This description of the Cambridge variant requires further verification.]

19. **# continuing candidates = # unfilled seats**
This decision step checks if the number of continuing candidates is equal to the number of unfilled seats, in which case they can all be elected to finish the tabulation without additional rounds.

Flows. Processing control flows in from the No path from Step 15 (Any candidate votes > T?). The Yes path to Step 21 (Elect all continuing candidates) is taken if the number of continuing candidates is equal to the current number of unfilled seats. The No path to Step 20 (Defeat candidates with fewest votes and stage their CVRs) is taken if there are more continuing candidates than the number of unfilled seats.

See also. None.

Resources.
- VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.
- VM.RCV.NumberSeatsToFill, the number of seats to be filled.

Explanation. No additional explanation.

Variants. None.

20. **Defeat candidates with fewest votes and stage their CVRs**
This processing step defeats one or more candidates with the fewest votes, or any other candidates specially identified by a variant, and stages their CVRs for reassignment to other continuing candidates.

Flows. Processing control flows in from the No path from Step 19 (# continuing candidates = # unfilled seats?) and out to Step 6 (Process a staged CVR).

See also. Step 15 (Candidate votes > T?) also describes special conditions for defeating candidates.

Resources.
- VM.RCV.Tabulation.VoteCount for all candidates, the most recent vote totals from Step 12 (Tabulate vote totals).
- VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.
- VM.RCV.Tabulation.Configure.MultipleDefeats, an indication of whether or under what circumstances more than one candidate is to be defeated in a round.
- VM.RCV.Tabulation.Configure.PriorityDefeats, an indication of whether or under what circumstances one or more candidates are to be defeated regardless of their current vote totals or even there is a continuing candidate that satisfies the threshold criterion or an elected candidate with undistributed surplus.
- VM.RCV.Tabulation.Configure.ResolveTiesForFewestVotes, an indication of how to resolve ties for being the candidate with the fewest votes.
- Other variant-specific information that may be needed to determine whether special defeat conditions apply, such as:
  - VM.RCV.NumberSeatsToFill, the number of seats to be filled.
  - VM.RCV.Tabulation.IsCandidateUndeclaredWriteIn for all candidates, an indication of whether a candidate is an undeclared write-in candidate.
  - VM.RCV.Tabulation.TotalBallotSelections for each candidate, the total number of (valid, i.e. not overvote?) ballot selections across all ranking levels for each candidate.
  - VM.RCV.Tabulation.Threshold, the threshold value.
  - VM.RCV.Tabulation.ThresholdComparisonType, the type of threshold criterion comparison to use: must exceed the threshold (>) or must reach or exceed the threshold (≥).

Explanation. The basic approach is to defeat the candidate with the fewest votes and then transfer that candidate's votes to other candidates. There can be variant-specific rules for how to resolve a tie between candidates with the fewest votes.

Variants. Some variants use some sort of lottery to resolve a tie for having the fewest votes to select which of the tied candidates will be defeated, while others look at previous round results and defeat the candidate with the lowest votes in the most recent previous round that is not tied; using a lottery only if there is still a tie after looking at all previous rounds.

It is also common for single-seat contests to allow or require defeating in a single round a group of continuing candidates who individually have fewer votes than any other continuing candidates (a.k.a. batch elimination) if their combined vote total is less than the candidate with the next higher vote total. If a candidate is in the group, any candidate with the same number or

fewer votes must also be in the group. Use of this option will not change who is elected, i.e. its use is outcome invariant. San Francisco and Oakland rules require use of this option but Oakland does not use it and San Francisco has stopped using it in favor of other tabulation options its voting system supports. For multi-seat elections, the criteria for defeating two or more candidates in a round or defeating a candidate while there are other candidates that satisfy the threshold criterion are more complicated if outcome invariance is to be maintained.

Some variants have exceptional rules that prioritize defeating one or more candidates. Minneapolis allows votes to be counted for undeclared write-in candidates in the first round (but doesn't allow them to be elected) and then requires that any such candidates be defeated before tabulating votes for the second round, regardless of how many votes those candidates have. Minneapolis also requires a candidate to be defeated if the total (valid?) CVR selections across all ranking levels is small enough to ensure that the candidate can not be elected. For multi-seat contests, Minneapolis requires those special defeats and batch eliminations based on single-seat conditions even if there is a continuing candidate that satisfies the threshold criterion. HR 3057 requires certain candidate eliminations, including batch eliminations, even if there are continuing candidates that satisfy the threshold criterion. Cambridge requires defeat, after all surplus is transferred from candidates elected in the first round, of every candidate with fewer than 50 votes. The 50-vote minimum is derived the requirement to have 50 signatures on a candidate's nominating petition. None of these exceptional elimination rules is guaranteed to be outcome invariant compared to single elimination only when there is no surplus to be transferred.

.

21. **Elect all continuing candidates**
    This processing step elects all continuing candidates in order to fill the remaining unfilled seats.

    Flows. Process control flows in from the Yes path from Step 19 (# continuing candidates = # unfilled seats?) and out to Step 22 (Done).

    See also. None.

    Resources. VM.RCV.Tabulation.CandidateStatus for all candidates, the status of whether a candidate is a defeated, elected, or continuing candidate.

    Explanation. This allows a candidate to be elected without satisfying the threshold criterion.

    Variants. None.

22. **Done**

This process termination step indicates that all seats have been filled or all eligible candidates have been elected and that the tabulation process is complete.

Flows. Processing control flows in from:

- Step 4 (All seats filled?) of the Marked First-choice Tabulation subprocess.
- Step 17 (All seats filled?).
- Step 21 (Elect all continuing candidates).

See also. None.

Resources. None.

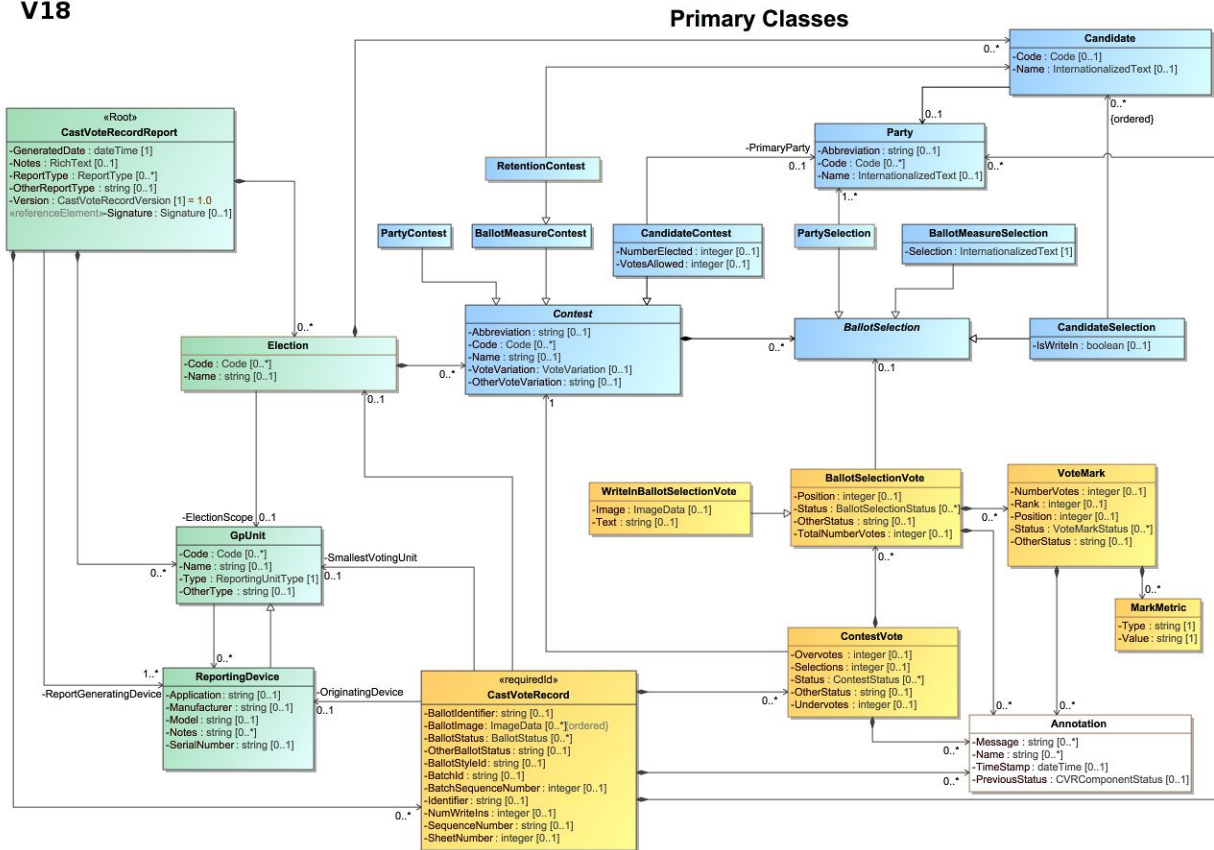**Explanation.** No additional explanation.

Variants. None.

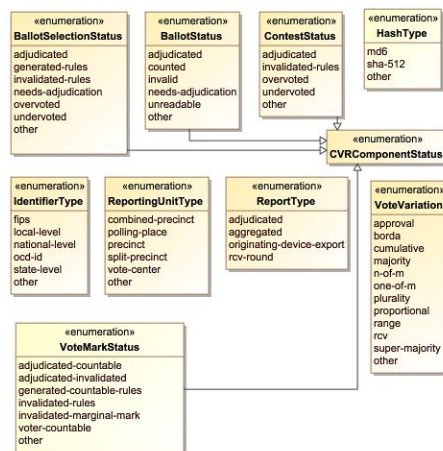Figure X: **Cast Vote Record (CVR) Common Data Format (CDF)**

V18

Primary Classes

Candidate
-Code : Code [0..1]
-Name : InternationalizedText [0..1]

«Root» CastVoteRecordReport
-GeneratedDate : dateTime [1]
-Notes : RichText [0..1]
-ReportType : ReportType [0..*]
-OtherReportType : string [0..1]
-Version : CastVoteRecordVersion [1] = 1.0
«referenceElement»-Signature : Signature [0..1]

Party
-Abbreviation : string [0..1]
-Code : Code [0..*]
-Name : InternationalizedText [0..1]

-PrimaryParty

RetentionContest

PartyContest

BallotMeasureContest

CandidateContest
-NumberElected : integer [0..1]
-VotesAllowed : integer [0..1]

PartySelection

BallotMeasureSelection
-Selection : InternationalizedText [1]

Contest
-Abbreviation : string [0..1]
-Code : Code [0..*]
-Name : string [0..1]
-VoteVariation : VoteVariation [0..1]
-OtherVoteVariation : string [0..1]

Election
-Code : Code [0..*]
-Name : string [0..1]

BallotSelection

CandidateSelection
-IsWriteIn : boolean [0..1]

-ElectionScope [0..1]

GpUnit
-Code : Code [0..*]
-Name : string [0..1]
-Type : ReportingUnitType [1]
-OtherType : string [0..1]

-SmallestVotingUnit

WriteInBallotSelectionVote
-Image : ImageData [0..1]
-Text : string [0..1]

BallotSelectionVote
-Position : integer [0..1]
-Status : BallotSelectionStatus [0..*]
-OtherStatus : string [0..1]
-TotalNumberVotes : integer [0..1]

VoteMark
-NumberVotes : integer [0..1]
-Rank : integer [0..1]
-Position : integer [0..1]
-Status : VoteMarkStatus [0..*]
-OtherStatus : string [0..1]

MarkMetric
-Type : string [1]
-Value : string [1]

ContestVote
-Overvotes : integer [0..1]
-Selections : integer [0..1]
-Status : ContestStatus [0..*]
-OtherStatus : string [0..1]
-Undervotes : integer [0..1]

ReportingDevice
-Application : string [0..1]
-Manufacturer : string [0..1]
-Model : string [0..1]
-Notes : string [0..*]
-SerialNumber : string [0..1]

-ReportGeneratingDevice
-OriginatingDevice

«requiredId» CastVoteRecord
-BallotIdentifier : string [0..1]
-BallotImage : ImageData [0..*][ordered]
-BallotStatus : BallotStatus [0..*]
-OtherBallotStatus : string [0..*]
-BallotStyleId : string [0..1]
-BatchId : string [0..1]
-BatchSequenceNumber : integer [0..1]
-Identifier : string [0..1]
-NumWriteIns : integer [0..1]
-SequenceNumber : string [0..1]
-SheetNumber : integer [0..1]

Annotation
-Message : string [0..*]
-Name : string [0..1]
-TimeStamp : dateTime [0..1]
-PreviousStatus : CVRComponentStatus [0..1]

Enumerations

«enumeration» BallotSelectionStatus
adjudicated
generated-rules
invalidated-rules
needs-adjudication
overvoted
undervoted
other

«enumeration» BallotStatus
adjudicated
counted
invalid
needs-adjudication
unreadable
other

«enumeration» ContestStatus
adjudicated
invalidated-rules
overvoted
undervoted
other

«enumeration» HashType
md6
sha-512
other

«enumeration» CVRComponentStatus

«enumeration» IdentifierType
fips
local-level
national-level
ocd-id
state-level
other

«enumeration» ReportingUnitType
combined-precinct
polling-place
precinct
split-precinct
vote-center
other

«enumeration» ReportType
adjudicated
aggregated
originating-device-export
rcv-round

«enumeration» VoteVariation
approval
borda
cumulative
majority
n-of-m
one-of-m
plurality
proportional
range
rcv
super-majority
other

«enumeration» VoteMarkStatus
adjudicated-countable
adjudicated-invalidated
generated-countable-rules
invalidated-rules
invalidated-marginal-mark
voter-countable
other

Secondary Classes

Code
-label : string [0..1]
-Value : string [1]
-Type : IdentifierType [1]
-OtherType : string [0..1]

InternationalizedText
-label : string [0..1]

File
attributes
«simpleContent»-Data : base64Binary [1]
-fileName : string [0..1]
-mimeType : string [0..1]

LanguageString
«simpleContent»-Content : string [1]
-language : language [1]

Image

«enumeration» CastVoteRecordVersion
1.0

ImageData
-Location : anyURI [0..1]
«referenceElement»-Signature : Signature [0..1]

Hash
-Type : HashType [1]
-OtherType : string [0..1]
-Value : string [1]