NIST Special Publication 1500-107

Voting Methods and Tabulation Models

A standard for precise specification of elections administration voting methods and tabulation models, vote variations, and operations on vote selection datasets.

NIST Interoperability Public Working Group
Voting Methods and Tabulation Subgroup
Voting Methods Working Group (VMWG)

Version 1.0
December, 2024

John Wack, Lauren Lochridge, Philip Stark, Chris Hughes, Editors

This publication is available free of charge from:

http://dx.doi.org/URL-of-this-doc

NIST Special Publication 1500-107

Voting Methods and Tabulation Models

A standard for precise specification of elections administration voting methods, counting, tabulation, and mathematical evaluation modules

Version 1.0

John Wack, Lauren Lochridge, Philip Stark, Chris Hughes Editors

Authors

John Wack,
Lauren Lochridge,
Herb Deutsch,
George Gilbert,
Chris Hughes,
Celeste Landry,
Phillip Stark

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.1500-107

January, 2025

U. S. Department of Commerce
Gina Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, Director

National Institute of Standards and Technology (NIST) Special Publication 1500-107

NIST Special Publication series 1500 is intended to capture external perspectives related to NIST standards, measurement, and testing-related efforts. These external perspectives can come from industry, academia, government, and others. These reports are intended to document external perspectives and do not represent official NIST positions.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in

accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST publications are available at http://www.nist.gov/publication-portal.cfm.

Comments on this publication may be submitted to John P. Wack

National Institute of Standards and Technology
Attn: John Wack, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8970) Gaithersburg, MD 20899-8970
Email: john.wack@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. IT's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. This document reports on ITLs research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

This publication describes a standard for precise specification of elections administration voting methods, tabulation methods, and common operations on vote selection datasets. This specification's primary purpose is to specify a set of voting methods variants where each voting method model variant definition is sufficiently complete as a unit and has a unique identifier and description. Each variant specification is a precise definition that can be referenced without ambiguity, reused, and understood by a variety of stakeholders including legislators, elections officials, analysts, systems and software manufacturers, and those performing testing and certification. This specification may optionally be adopted as a standard that complies with Data Interoperability as defined in the NIST Common Data Format. This specification includes a reference implementation of select voting methods variants and the tabulation methods that voting methods use to produce election contest outcomes. In this version of this standard, a reference implementation of this standard has been produced. Specifically, an open source Ranked Choice Voting (RCV) Tabulator that implements the subset of Voting and Tabulation methods that are RCV Voting Methods.

Keywords

Common data interoperability format; contest; district; election results; jurisdiction; overvote; political office; political party; precinct; undervote.

Acknowledgments

The authors wish to thank their colleagues of the National Institute of Standards and Technology VVSG-Interoperability Public Working Group, who reviewed drafts of this document and contributed to its technical content. The authors gratefully acknowledge and appreciate the following contributors for their keen and insightful assistance with developing this specification, affiliations where given are for the purpose of identification only:

Authors:
John Wack, NIST

Lauren Lochridge, VMWG Chair Independent Researcher & Principal Technologist, Google.

Phillip Stark, UC Berkeley

Herb Deutsch, Election Systems & Software

George Gilbert, Ranked Choice Voting Resource Center

Chris Hughes, FairVote/Ranked Choice Voting Resource Center,

Celeste Landry, Voting Methods Team, League of Women Voters of Boulder County (Colorado)

The editors for this document are:

John Wack, NIST

Lauren Lochridge, VMWG Chair Independent Researcher & Principal Technologist, Google.

Phillip Stark, UC Berkeley

Chris Hughes, FairVote/Ranked Choice Voting Resource Center

Contributors:
Affiliations where given are for the purpose of identification only, may reflect prominent affiliations at the time of most substantive contribution rather than currently, and do not imply the participation of the affiliated entity.

Joseph Kiniry, Galois and Free & Fair	Carsten Schurmann, IT University of Copenhagen
Kimball Brace, Election Data Services, Inc.	Kenneth Bennett, Los Angeles County, CA
David Buerger, Elections Specialist Wisconsin Government Accountability Board, WI	John Dziurlaj, Elections Consultant, Hilton Roscoe LLC Formerly Office of Secretary of State, OH.
Barbara Simons, Former President ACM, Former EAC Board of Advisors	Olivier Pereira, Catholic University of Leuven
Ron Rivest, MIT Institute Professor at the Massachusetts Institute of Technology (MIT), and a member of MIT's Department of Electrical Engineering and Computer Science, Cryptography	Don Rehill, Dir. Elections Research and Vote Tabulation, Associated Press

Peter Ryan, University of Luxembourg	Chris Jerdonek, Elections Commissioner & President of the City & County of San Francisco Elections Commission, CA
Tammy Patrick, Senior Advisor to the Democracy Project, Bipartisan Policy Center	Jared Marcotte, Open Source Advocate, Developer, and UX Strategist
Benjamin Rice, Dominion Voting	Steve Schneider, Asooc. Dean, Director, Surrey Centre for Cyber Security University of Surrey
Beth Ann Surber, West Virginia, Office of Secretary of State, WV	Chris Jackson, Ipsos Public Affairs
Sarah Whitt, Wisconsin Elections Division	David Wallick, Votem
Arthur Keller, Computer Engineering department, University of California, Baskin School of Engineering	Lynn Garland, Independent
Michael Greenman, VP of Emerging Technology & Corporate Development, SOE, a Scytl company	Neal McBurnett, ElectionAudits
Art Griesser, Prometheus Computing	Vanessa Teague, University of Melbourne, Dept. of Computing and Information Systems
John Rose, Maxim Consulting, Inc., a Scytl Company	Justin Moore, Google, Voting Information Project
Poorvi Vora, GWU	Rob Richie, FairVote
David Becker, Center for Elections Innovation & Research	Jameson A. Quinn, Harvard
Jon Moldover, Bright Spots (San Francisco), Developers of the NIST 1500-107 RCVRC Reference Implementation	Louis Eisenberg, Bright Spots (San Francisco), Developers of the NIST 1500-107 RCVRC Reference Implementation
Hylton Edingfield, Bright Spots (San Francisco), Developers of the NIST 1500-107 RCVRC Reference Implementation	David Cary, Individual Contributor

Ryan Macias, EAC Former Acting Director of Testing and Certification, U.S. Election Assistance Commission	Matthew Masterson, EAC Former Commissioner (2014-2018), Senior Cybersecurity Advisor at the Department of Homeland Security Elections Security
The approximately 50 members of the Voting Systems Working Group, (VMWG), who may have contributed.	

Affiliations where given are for the purpose of identification only, may reflect prominent affiliations at the time of most substantive contribution rather than currently, and do not imply the participation of the affiliated entity.

Executive Summary

This publication specifies voting methods and tabulation methods or modules that operate on vote selection data. The elements of this specification are precisely defined voting method models. The models enable Data Interoperability and Data Integrity of election data sets operated upon by processes composed of modules performing counting, tabulation, mathematical evaluation or common vote selection data set operations.

The current prevailing practice is that written legislation, statute, rules, or Requests For Proposals (RFP) and other specifications produced by Elections Officials (EO) as algorithmic plain language definitions specify requirements for Elections Administration (EA) systems voting, counting, and tabulation processes. Any spoken and written language is naturally less precise than a mathematical specification. No matter how well crafted, when mathematical or constraint logic requirements of EA systems are specified in plain language, unintended under-specification or unplanned gaps in the specification may occur. Under-specification or gaps may negatively impact the validity of election operations and insufficiently support data interoperability and the integrity of election data sets. The lack of a shared, precisely defined set of standard voting method models for voting system modules causes unnecessary consumption of time and resources. Under-specification can increase risk and rates of error when stakeholders have to repeat the same work for common voting methods and are required to interpret under-specified or inherently ambiguous definitions.

The voting method models in this specification can be referenced without ambiguity, reused, and understood by EA systems and a variety of stakeholders, including legislators, elections officials, analysts, systems and software manufacturers, elections service providers, and testing and certification labs.

When a mathematical model unambiguously defines each voting method or module that operates on election data, that method's specification, characterization, and properties are precisely known, supporting accurate and reliable analysis of election data. Precisely known mathematical characterization of system components supports robust and correct voting systems and facilitates their development, operation, analysis, and testing.

Examples of typical EA processes that perform counting, tabulation, or mathematical evaluation operations on elections data sets include but are not limited to counting cast ballots, determining election outcomes, tabulation, performing aggregations to determine statewide elections counts or inter-jurisdiction election

counts, data analysis, auditing, verification, acceptance testing for new systems, capacity planning, and application of information security and information privacy requirements.

This specification is intended for Election officials, administrators, voting and election equipment manufacturers and service providers, election-affiliated organizations, election analysts, and the general public.

Table of Contents

Executive Summary **7**
1 Introduction **18**
1.1 Purpose 18
1.2 Scope and Objectives 18
1.3 Motivation and Methodology 20
1.4 Document Structure 21
1.5 Future Work 21
2 Background and Overview **22**
Voting Method Universe Definitions 23
2.1 Standards and Specifications Adopted by This Standard 23
2.1.1 Elections Administration Business Process 23
2.1.2 Ballot Definition CVR, Adopted Standard 24
2.1.3 Cast Vote Record CVR, Adopted Standard 24
2.1.4 Election Results Reporting ERR/ENR, Adopted Standard 24
2.1.5 Election Logging EEL, Adopted Standard 24
2.1.6 VVSG Cybersecurity Requirements, Adopted Recommended Requirement 24
2.1.7 NIST Voting Glossary 24
2.1.8 Voluntary Voting Systems Guidelines (VVSG) Principles and Guidelines 24
2.1.9 Technical Guidelines Development Committee (TGDC) Recommendations 24
3 Voting and Tabulation Method Domain Modeling **25**
3.1 Authoritative References used in Voting and Tabulation Method Domain Modeling 25
3.1.1 Legislative References 25
3.1.2 Elections Official and Administrator References 25
3.2 Tabulation Process Models Definitions 25
3.2.1 Tabulation Process Models Introduction 27
3.2.1.1.1 Treatment of Write-ins 28
3.2.1.1.2 Treatment of Tiebreaking 29
3.2.1.1.3 Treatment of Withdrawn or Disqualified Candidate 30
3.2.2 Tabulation Process Flow, High Level, By Contest 31
3.2.2.1 Introduction 31
3.2.3 Process N-of-M, Baseline Contest 37
3.2.3.1 Introduction 37
3.2.4 Plurality, See N-of-M 40
3.2.5 Majority, See N-of-M 40
3.2.6 Super-Majority, See N-of-M 40
3.2.7 Approval, See N-of-M 40

3.2.8 Limited, See N-of-M 40
3.2.9 Process Straight-Party Contest 40
3.2.9.1 Introduction 40
3.2.10 Process Contest Linked to Straight-Party Contest 43
3.2.10.1 Introduction 43
3.2.11 Process Open Primary Party Preference Contest 46
3.2.11.1 Introduction 46
3.2.12 Process Open Primary Partisan Contest 48
3.2.12.1 Introduction 48
3.2.13 Process Cumulative Voting Contest 51
3.2.13.1 Introduction 51
3.2.14 Process Proportional Voting Contest 53
3.2.14.1 Introduction 53
3.2.15 Process Recall Controlling Contest 55
3.2.15.1 Introduction 55
3.2.16 Process Recall Controlled Contest 57
3.2.16.1 Introduction 57
3.2.17 Process Ranked Choice Voting (IRV, STV, Multi-Pass) Contest 58
3.2.17.1 Introduction 58
3.2.17.2 Ranked Choice Voting (IRV, STV) 59
3.2.17.3 Multi-Pass IRV/Bloc Preferential Voting 60
3.2.18 Process Top-2 IRV Contingent Vote 68
3.2.18.1 Introduction 68
3.2.19 Process Presidential Preference Contest 72
3.2.19.1 Introduction 72
3.2.20 Process Presidential Delegate Contest 74
3.2.20.1 Introduction 74
3.2.21 Process Massachusetts Group Vote Contest 76
3.2.21.1 Introduction 76
3.2.22 Test Conditions & Expected Outcomes 79
3.2.22.1 Introduction 79
3.2.22.2 Baseline Contest 80
3.2.22.3 Approval Voting 81
3.2.22.4 Straight Party Contest 82
3.2.22.5 Open Primary Contest 83
3.2.22.6 Cumulative Contest 85
3.2.22.7 Proportional Voting Contest 86
3.2.22.8 Controlling and Controlled Contest Pair 87

3.2.22.9 Ranked Choice Voting Contest 87
3.2.22.9.1 Unvoted choice (no candidate ranked at a ranking level) 87
3.2.22.9.1.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet0_Skipped1stChoice 87
3.2.22.9.1.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet3_Skipped Choice_Exhaust Ballot 89
3.2.22.9.1.3 Expected_Outcome_RCV Test Sets_Single-winner - TestSet4_Skipped Choice_Advance to next 91
3.2.22.9.1.4 Expected_Outcome_RCV Test Sets_Single-winner - TestSet5_Two Consec Skipped Choices_Exhaust Ballot 92
3.2.22.9.2 Overvoted choice (more than one candidate ranked at a ranking level) 93
3.2.22.9.2.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet1_Overvote_Exhaust Ballot 93
3.2.22.9.2.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet2_Overvote_Advance to next 94
3.2.22.9.2.3 Expected_Outcome_RCV Test Sets_Single-winner - TestSet2_Overvote_Advance to next 95
3.2.22.9.3 Repeated ranking of a previously ranked candidate 96
3.2.22.9.3.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet6_Repeat Ranking_Exhaust Ballot 96
3.2.22.9.3.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet7_Repeat Ranking_Advance to next 97
3.2.22.9.4 Threshold Calculation 97
3.2.22.9.4.1 Whole # Threshold 97
3.2.22.9.4.2 Fractional threshold 100
3.2.22.9.5 Top-Two IRV Contingent Vote 103
3.2.22.9.6 Presidential Preference Contests 105
3.2.22.9.7 Massachusetts Group Vote 107
4 Voting Methods Models 108
4.1.1 Terminology 108
4.2 Voting Methods Modules 116
4.2.1 Domain of discourse, notation, and dictionary 116
Domain Specific Language (DSL) Notation 116
4.2.2 Constraints that generally apply to most Voting Methods 118
4.2.3 High-Level Decision Rule Constraints Applicable to Voting Methods 122
4.2.4 Ties and Tiebreaking 124
4.2.5 Interrelated Contests or Connected Contests 126
4.2.6 N-of-M (Baseline) Contests 128
4.2.6.1 Plurality 128
4.2.6.1.1 Determine Outcome: Plurality, Candidates, Single Seat 128

4.2.6.1.2 Determine Outcome: Plurality, Candidates, Multiple Seats 130
4.2.6.2 Majority 131
4.2.6.2.1 Determine Outcome: Majority, Candidates, Single Seat 131
4.2.6.2.2 Determine Outcome: Majority, Measure 133
4.2.6.3 Super-Majority 135
4.2.6.3.1 Determine Outcome: Super-Majority, Measure 135
4.2.6.4 Approval Voting 136
4.2.6.4.1 Determine Outcome: Approval Voting, Candidates, Single Seat 136
4.2.6.4.2 Determine Outcome: Approval Voting, Candidates, Multiple Seats 139
4.2.7 Straight Party 141
4.2.7.1 Determine Party Indication, Straight Party 141
4.2.7.2 Determine Effect of Straight Party Indication on Connected Contests 143
4.2.8 Open Primary 146
4.2.8.1 Determine Open Primary 146
4.2.8.1.1 Determine Open Primary Party Preference, Controller 148
4.2.8.1.2 Determine Open Primary Partisan Contest, Controlled 150
4.2.9 Cumulative Voting 152
4.2.9.1 Determine Outcome: Cumulative Voting, Candidates, Multiple Seats 152
4.2.10 Proportional Voting 154
4.2.10.1 Determine Outcome: Proportional Voting, Candidates, Multiple Seats 154
4.2.11 Recall Contest 156
4.2.11.1 Determine Outcome: Recall Controlling Contest 156
4.2.11.2 Determine Outcome: Recall Controlled Contest 158
4.2.12 Ranked Choice Voting (RCV) 160
4.2.12.1 High-Level Requirements Applicable to Most RCV Methods 160
4.2.12.2 Determine Outcome: Colorado Local Option (IRV) 169
4.2.12.3 Determine Outcome: Statewide IRV_1 175
4.2.12.4 Determine Outcome: CA-NM-OR-PortlandME (IRV) 181
4.2.12.5 Determine Outcome: Virginia Local Option (IRV) 187
4.2.12.6 Determine Outcome: Portland, OR (IRV) 193
4.2.12.7 Determine Outcome: Utah Local Option (IRV) 199
4.2.12.8 Determine Outcome: NYC (IRV) 205
4.2.12.9 Determine Outcome: Minnesota Cities (IRV) 211
4.2.12.10 Determine Outcome: Utah Local Option (Multi-Pass IRV/Bloc Preferential Voting) 217
4.2.12.11 Determine Outcome: Colorado Local Option (STV) 223
4.2.12.12 Determine Outcome: Albany, CA (STV) 231
4.2.12.13 Determine Outcome: Virginia Local Option (STV) 238
4.2.12.14 Determine Outcome: Portland, OR (STV) 246

4.2.12.15 Determine Outcome: Minnesota Cities (STV) 254
4.2.12.16 Determine Outcome: Cambridge, MA (STV) 262
4.2.13 Top-Two IRV Contingent Vote Contest 269
4.2.13.1 Determine Outcome: Top-Two IRV Contingent Vote Contest, Candidates, Single Seat 269
4.2.14 Presidential Preference Primary 272
4.2.14.1 Presidential Preference Primary Election 272
4.2.14.1.1 Determine Outcome: Presidential Preference Contest, Controller 274
4.2.14.1.2 Determine Outcome: Presidential Delegate Contest, Controlled 276
4.2.15 Massachusetts Group Vote Contest 278
4.2.15.1 Determine Outcome: Massachusetts Group Vote Contest 278
4.3 Domain Models 280
Election Business Process Model 280
VMWG Voting Methods Universe 281
Appendix A: DSL Dictionary and Glossary 282
VMWG Domain Specific Language Dictionary 282
Domain Specific Language (DSL) Notation 297
Glossary 299
Appendix B: Reference Implementation, RCV Tabulator 309
RCTab f/k/a the Universal RCV Tabulator 309
Overview 311
Contest Info Tab 312
CVR Files Options 314
CDF 317
Clear Ballot 318
Dominion 319
ES&S 320
Hart 322
Winning Rules Options 325
Tiebreaking 328
Single-winner majority determines winner 329
Multi-winner allow only one winner per round 330
Multi-winner allow multiple winners per round 331
Bottoms-Up 332
Bottoms-up using percentage threshold 333
Multi-pass IRV 334
Tiebreak Modes 335
Random, Generate Permutation, Previous Round Counts (then random) 335
Stop counting and ask, previous round counts (then stop counting and ask), use candidate

order in the config file 335
Voter Error Rules 336
Output Tab 338
Examples of output files include: 339
1. summary.csv 339
2. audit_0.log 340
3. summary.json 341
4. cvr_cdf.json 343
Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.0.1 345
Certification of RCV Tabulator including compliance with NIST 1500-103 Cast Vote Record CDF, Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.1.0 352
Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.2.0 363
Appendix C: Related Standards, Models and Information 374
VVSG certification 374
Election Business Process Model BPMN Diagrams, Use Cases of 1500-107 in Election Business Process Model BPMN Diagrams 377
NIST 1500-103 Cast Vote Record 380
NIST 1500-100 Election Results Reporting 381
NIST 1500-101 Election Event Logging 382

List of Figures

Figure 1. Process Contests On A Ballot flow.
Figure 2. High level baseline ballot tabulation process flow
Figure 3. Process N-of-M, Baseline contest process flow.
Figure 4. Straight party contest process flow.
Figure 5. Contest linked to straight-party contest process flow
Figure 6. Party preference contest process flow
Figure 7. Open primary partisan contest process flow
Figure 8. Cumulative voting process flow
Figure 9. Proportional Voting process flow
Figure 10. Recall controlling contest process flow
Figure 11. Recall controlled contest process flow
Figure 12. Ranked choice voting contest process flow
Figure 13. Process flow for top-two IRV, also known as contingent vote contest
Figure 14. Presidential preference contest process flow
Figure 15. Presidential delegate contest process flow
Figure 16. Massachusetts Group Vote contest process flow
Figure 17. RCTab Tab 1: Contest Info.
Figure 18: Example Completed Contest Info tab
Figure 19: RCTab Tab 2: CVR Files
Figure 20: RCTab Tab 2: CVR Files | CDF
Figure 21: RCTab Tab 2: CVR Files | Clear Ballot
Figure 22: RCTab Tab 2: CVR Files | Dominion
Figure 23: RCTab Tab 2: CVR Files | ES&S
Figure 24: ES&S Example CVR File
Figure 25: RCTab Tab 2: CVR Files | Hart
Figure 26: RCTab Tab 3: Candidates
Figure 27: Example Complete Candidates tab
Figure 28: RCTab Tab 4: Winning Rules
Figure 29: RCTab Tab 4: Winning Rules | Single-winner majority determines winner screen
Figure 30: RCTab Tab 4: Winning Rules | Multi-winner allow only one winner per round screen
Figure 31: RCTab Tab 4: Winning Rules | Multi-winner allow multiple winners per round screen
Figure 32: RCTab Tab 4: Winning Rules | Bottoms-up screen
Figure 33: RCTab Tab 4: Winning Rules | Bottoms-up using percentage threshold screen
Figure 34: RCTab Tab 4: Winning Rules | Multi-pass IRV screen
Figure 35: RCTab Tab 4: Winning Rules | Random Tiebreak mode screen
Figure 36: RCTab Tab 4: Winning Rules | User input Tiebreak mode screen
Figure 37: RCTab Tab 5: Voter Error Rules
Figure 38: Example Complete Voter Error Rules tab
Figure 39: RCTab Tab 6: Output
Figure 40. RCTab Summary.csv file example.
Figure 41. RCTab Audit Log example.

Figure 42. RCTab Summary.json file example.
Figure 43. RCTab CVR CDF .json example.
Figure 44. Tally Ballots process.
Figure 45. Logic and Accuracy Testing process.
Figure 46. Report results process.
Figure 47. Verify vote count process.

List of Tables

Table 1. Baseline contest expected outcomes and sample test sets
Table 2. Approval Voting contest sample test sets and outcomes
Table 3. Straight-party expected outcomes and sample test sets
Table 4. Open primary expected outcomes and sample test sets
Table 5. Cumulative voting expected outcomes and sample test sets
Table 6. Proportional voting expected outcomes and sample test sets.
Table 7. Controlling and Controlled Contest expected outcomes and sample test sets.
Table 8. Ranked choice voting expected outcomes and sample test sets for a ballot with a skipped ranking in the first rank.
Table 9. Ranked choice voting expected outcomes and sample test sets for a ballot exhausting after a single skipped ranking.
Table 10. Ranked choice voting expected outcomes and sample test sets for advancing to the next valid ranking when encountering skipped ranking
Table 11. Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering consecutive skipped ranking
Table 12. Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering an overvoted ranking
Table 13a. Ranked choice voting expected outcomes and sample test sets for skipping to the next valid ranking when encountering an overvoted ranking
Table 13b. Ranked choice voting expected outcome and sample test sets for skipping to the next valid ranking when encountering an overvote and using batch elimination.
Table 14. Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering a repeat ranking
Table 15. Ranked choice voting expected outcomes and sample test sets for advancing to the next valid ranking on a ballot when encountering a repeat ranking
Table 16.1. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold. Round descriptions.
Table 16.2. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold, Rounds 0–2.
Table 16.3. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold, Rounds 3–6.
Table 17.1. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Round descriptions.
Table 17.2. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Rounds 0-3.
Table 17.3. Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Rounds 4-6.
Table 18. Top-Two IRV Contingent Vote expected outcomes and sample test sets.
Table 19. Presidential delegate contest expected outcomes and sample test sets.
Table 20. Massachusetts Group Vote expected outcomes and sample test sets

1 Introduction

The Voting Methods Working Group (VMWG) develops guidance and specifications of robust models and other formalizations to describe a variety of voting methods, algorithmic modules, and interoperable operations on election data sets, for counting, tabulation, or operations on election data sets as usually performed by systems that operate elections to determine counts and outcomes.

Voting equipment manufacturers and election administrators may use VMWG models to more easily implement a given voting method or to produce modules that operate on vote selection data in election administration processes.

VMWG is also working to develop a reference architecture, reference implementations, and set of tools to verify the proper implementations of different voting methods and modules that operate on vote selection data sets, and identify gaps in legislation or administrative procedures that specify methods of counting, tabulation, or numerical evaluation.

The models and use cases identified by VMWG can also be used to improve the usability and understanding of voting methods by election administrators, legislators, analysts, and the public.

As an aide to understanding how to use the models this standard provides, this document provides a reference implementation and reference tabulation processes.

1.1 Purpose

This standard supports Data Interoperability and Data Integrity (DI) in Elections Administration Systems (EAS) that operate a variety of voting schemes, counting modules, and modules performing operations on voting data sets. The standard enables modeling, configuration, validation, and testing of EAS to be valid to a high degree of confidence and accuracy, as well as fail-safe and cost-effective. Thereby, this standard supports the Elections Administrator (EA) in producing valid operations of U.S. government elections for the public.

This specification's purpose is to facilitate elections integrity through standard and precise definitions of voting methods and EAS modules that perform counting, tabulation, or mathematical evaluation operations on election data sets, specifically vote selection data sets. Thereby, this standard enables Data Interoperability and verifiable correctness of operations on elections data sets.

Adoption of this specification enables manufacturers, elections administrators (EA), elections officials (EO), elections analysts, and testing and certification labs to produce systems and modules compatible with the voting methods described within and seeks to inform the Technical Guidelines Development Committee (TGDC) of the Election Assistance Commission (EAC) and future versions of the Voluntary Voting System Guidelines (VVSG), and other stakeholders as to the practical operation of voting and tabulation methods.

1.2 Scope and Objectives

This standard provides a set of rigorously and precisely defined models of voting and tabulation methods where Elections Administration (EA) systems perform counting, tabulation, mathematical

evaluation, or common operations on vote data sets. This standard specification is structured to enable the adopter to selectively adopt parts of the standard that pertain to the jurisdiction's use cases.

The primary focus of this standard is models of voting methods, also known as voting variations, currently in use or projected to become in use in U.S. government elections over the lifecycle of this standard. However, nothing about this standard's domain model, methodology, and reference architecture is designed to prevent it from being applied to other election administration use cases.

This standard's elements are models. The domain or universe modeled is voting methods, tabulation methods, and process flows used where Elections Administration (EA) systems perform counting, tabulation, numerical evaluation, or data set operations on election vote data sets. The inputs and references for modeling the EA domain are specifications in the form of written legislation or an algorithmic plain language specification that typically may be the work product of legislators, Elections Officials (EO), or Boards or Commissions of Elections (BoE) (CoE). Sources for this specification also include RFIs or RFPs for EA systems manufacturers and other documents, such as voter-facing instructions produced by elections officials.

Examples of research topic areas presenting mathematical definitions or models of voting methods include research into social choice, vote power, voting system criteria, verifiable voting systems, and auditable voting systems. Models presented in research publications primarily focus on voting system criteria and vote power analysis, and are often metamodels, or higher-level and more general models of the in-use voting methods of the EA domain.

This specification is geared towards the following audiences:

- election officials,
- voting equipment and systems manufacturers,
- voting systems testing laboratories,
- election-affiliated or associated organizations,
- election analysts, and
- the public.

Metamodels or theoretical models are out of scope.

Definitions of new voting system criteria or voting system metrics for use specifically in comparison of, or advocating for, the use of particular voting schemes, methods, systems implementation, or modules are out of scope.

The innovation of voting methods or schemes that are not in use, in practice, or reasonably projected to become in use in U.S. government-operated elections are out of scope.

Reference artifacts are existing public domain legislative specifications (legal text) or other forms of official documentation produced by or used by legislators, elections officials, or administrators to provide voting schemes, methods, or particular modules for counting, tabulation, or mathematical evaluation within EA systems.

Nothing about this specification is intended to constrain adopters of this specification from implementing it either as a system of interoperating sub-systems, or as independent interoperating systems, or a mix of both, as the particular use-case as defined by Elections Officials (EOs) or Elections Administrators (EAs) may vary. Therefore, the VMWG models are agnostic concerning any particular elections administration system architecture.

Nothing about this specification is intended to constrain adopters from implementing a proprietary representation of data used internally in a system or subsystem component or device as an innovative extension of this specification to expand upon its features. For use cases of this specification where a proprietary vote data set may be used by, for example, a device within a systems architecture, the exchange of a vote data set is required to be implemented using NIST SP 1500-103 Cast Vote Record (CVR) specification in the choice of XML or JSON formats. This specification supports output from a component implemented following this specification in the form of NIST SP 1500-103 Cast Vote Record (CVR) and also NIST SP 1500-100 Election Results Reporting (ERR) in the choice of XML or JSON formats. However, an adopter of this standard may additionally implement and support other formats.

Nothing about this specification is intended to constrain adopters from implementing a proprietary representation of data used internally in a system or subsystem component or device as an innovative extension of this specification to expand upon its features. For use cases of this specification where a proprietary vote data set may be used, for example, by a device within a systems architecture, the exchange of a vote data set is required to be implemented using NIST SP 1500-103 Cast Vote Record (CVR) specification in the choice of XML or JSON formats. However, the adopter of this standard may additionally support other formats. This specification supports output from a component implemented in accordance with this specification in the form of NIST SP 1500-103 Cast Vote Record (CVR) and also NIST SP 1500-100 Election Results Reporting (ERR) in the choice of XML or JSON formats.

1.3 Motivation and Methodology

Defining each voting method precisely permits knowing its specification, characterization, and properties exactly. This precise knowledge supports the accurate and reliable analysis of elections vote selection data. Precise characterization of system components enables robust and correct voting systems and facilitates their development, analysis, and testing.

No complete compendium exists of precisely researched and modeled voting methods and tabulation methods that are used in practice today. In-use voting methods are those defined in currently applicable legislation or elections official's specifications, or as realized in systems that operate elections today, or that are reasonably projected to become in-use during the lifecycle of this standard.

Reference models or definitions of voting schemes or methods typically exist in various forms and contexts, including glossary definitions, RFPs by elections administration systems and software manufacturers and systems, systems documentation, and research works on social choice theory and voting theory. None of these varieties of existing definitions satisfy the need for a standard set of precise definitions of applied voting methods, as specified in the application as plain language algorithmic statute or legislation, so that the voting methods specifications are commonly understood, without ambiguity, among the full spectrum of stakeholders in the applied operation and administration of elections.

Based on information gathering and the knowledge of our stakeholders and members, this working group identifies a representative knowledge base that is sufficient to model the universe of voting methods and tabulation methods in use in the United States. Authoritative reference sources used to build the knowledge base include legislation, election law, local jurisdiction rules, documentation of practice of operating elections, and RFPs, RFIs, and contracts. In some cases, we interview election

officials for clarification about operational practice. The plain language algorithmic definitions and rules are extracted from authoritative sources to develop models of tabulation methods and voting methods.

Use Cases

In this document, Use Cases are scenarios where actors interact with a system, process, or model. The primary purpose of the use case is to help identify features or characteristics that the representation of the modeled universe must satisfy to perform some function correctly or to satisfy requirements.

Support for a use case by this specification means that to the best of our ability in developing this specification and the voting method or voting variations models that it contains, this specification can apply to the use case as a scenario. Support of a use case does not imply that this specification specifies a model for each use case.

This standard supports use cases such as auditability, verifiability, capacity planning, testing, and certification, information security, information privacy, reportability, traceability, and logging, aggregation and "roll-up," mechanized tiebreaking, encryption if required for a voting method or vote selection data set, and elections result reporting.

This standard's use cases are useful to elections officials and administrators and align with the elections administrator's view of elections administration processes.

The tabulation scenarios listed in section 3 are the supported use cases for this version of this specification.

The use cases from an election business process layer point of view are documented in Appendix C.

1.4 Document Structure

This specification document is organized so that the key sections of interest to adopters who wish to implement the specification as a standard are Section 3 Tabulation Methods Process Models and Section 4 Voting Methods Models.

Section 3 contains detailed descriptions, flow charts, and processing examples of voted ballots for the various tabulation methods currently in use in the United States.

Section 4 contains voting methods modules that are representations of the plain language algorithmic definitions extracted from local jurisdiction legislation, rules, and other documentation that define voting methods as implemented in practice and operation in the United States.

1.5 Future Work

In future versions, this standard may extend existing formulae or models for additional use cases or include additional voting and tabulation methods and new varieties of numerical and data operations as may be justified by technology, legislation, and election administrators' needs.

At this time of writing, high-priority work planned for the next revision, Version 1.1, include:

- Expanding upon our initial, high-level, UML Model of the VMWG Voting Methods Universe.
- Implementing method specification modules for auditing.
- Implementing SAT Solver and/or Model Checker proofs of the constraint logic specification we have produced in the Domain Specific Language (DSL) that the VMWG has developed in this version of 1500-107.
- Development of an online interactive version of the voting method modules in this specification.
- Maintaining this specification as a living document as the model universe of voting methods in use in the United States.

2 Background and Overview

The Voting Methods Models Public Working Group (VMWG) is a subgroup of the NIST Interoperability Working Group (WG). The VMWG participants kicked off and began deliberating toward consensus in 2015 after initially being approved as an IEEE-SA standards development project sponsored by NIST's Voting Systems Standards Committee (VSSC). Subsequently, VMWG has operated as a subgroup of the NIST Technical Guidelines Development Committee (TGDC) Interoperability public working groups.

In this section, we describe the work of the VMWG and how this standard fits in the ecosystem of Interoperability standards, specifications, and use cases that are part of the NIST Common Data Format (CDF).

The VMWG views the universe of elections and voting interoperability as a framework for the generic elections administration domain. The standards or specifications that are parts of the framework correspond to the appropriate decomposition of the elections domain into an architecture where each specification fulfills a use case of CDF or interoperability. Examples of standards and specifications this specification adopts include NIST SP 1500-100 Election Results Reporting (ERR), NIST SP 1500-101 Election Events and Logging, and NIST SP 1500-103 Cast Vote Record (CVR) specifications. Standards in the CDF data transport interoperability area are generally modeled in the Unified Modeling Language and implemented in the two ubiquitous formats, XML data structures or JSON. A CDF is used to deliver data between systems or subsystems or often to store data in data structures within systems. CDFs are crucial to data integrity and may enable data security and privacy. CDFs also implement the commonly understood syntax and semantics for consumers and producers of the data.

The data interoperability and integrity of CDF are a focus of this standard. Standards and specifications in this layer are concerned with the interoperability of the election data sets, their data integrity, data structures in the common data format, and the data contained within the standard data structure operated upon by a system. To this end, test conditions and expected outcomes of tabulation suitable for logic and accuracy testing are provided in section 3.

By specifying voting methods and tabulation models for operating on vote selection data, this standard supports data interoperability where EA systems or modules of EA systems perform counting, tabulation, mathematical operations or common data set operations on vote selection data sets. Through the adoption of this standard, and extension of this standard in the form of the NIST CDFs that this standard adopts, stakeholders are enabled to know that operations for counting, tabulation, and other calculations performed on elections data will produce predictably correct results. Predictable performance of operations for

capacity planning and for testing, auditing, verifiability, and other analysis relying on known standard operations that perform calculations is a benefit of standard vote selection data set operations.

Voting Method Universe Definitions

We define the following key elements of the voting methods universe that VMWG has recommended during the development of VVSG 2.0, which were incorporated by reference:

Voting Method
A voting method encompasses a set of partially ordered process steps, tabulation methods, and decision rules. Each voting method has at least one associated tabulation method and at least one election outcome decision rule set consisting of at least one decision rule. Executing the appropriate tabulation method and election contest decision rule set is in the scope of the voting method.

Tabulation Method
A tabulation method is a set of partially ordered process steps and operations for performing counting, accumulation, aggregation, and other data and math operations on vote selection data sets. Each tabulation method supports at least one tabulation scenario.

Tabulation Scenario
A tabulation scenario is a situation and sequence of events or activities intended by election administrators to be sufficient to achieve a tabulation goal.

2.1 Standards and Specifications Adopted by This Standard

In this section we outline how standards, specifications, guidelines or frameworks adopted by this specification are used.

2.1.1 Elections Administration Business Process

- The development of this specification was aided by the NIST Election Modeling business process model. These models are represented as using Business Process Modeling Notation (BPMN) Version 2 [https://github.com/usnistgov/ElectionModeling].
- The NIST Election Business Process Models provide the election administration business process context for voting methods for tally and tabulation for this specification.
- This specification assumes the Tally process as the primary scope of Tabulation use case for voting method and tabulation method models : https://pages.nist.gov/ElectionModeling/diagrams.html#Diagrams___18_5_1_11940316_1500578226532_496469_36542.
  However, several other high level business process layer processes also use voting methods or tabulation methods. See Appendix C. Related Standards, Models and Information, Use Cases of 1500-107 in Election Business Process BPMN Diagrams.

2.1.2 Ballot Definition CVR, Adopted Standard

See Appendix C. Related Standards, Models and Information, Nist 1500-20 Ballot Definition

2.1.3 Cast Vote Record CVR, Adopted Standard

See Appendix C. Related Standards, Models and Information, Nist 1500-103 Cast Vote Record

2.1.4 Election Results Reporting ERR/ENR, Adopted Standard

See Appendix C. Related Standards, Models and Information, Nist 1500-100 Election Results Reporting

2.1.5 Election Logging EEL, Adopted Standard

See Appendix C. Related Standards, Models and Information, Nist 1500-101 Election Logging

2.1.6 VVSG Cybersecurity Requirements, Adopted Recommended Requirement

See Appendix C. Related Standards, Models and Information, VVSG Cybersecurity Requirements

2.1.7 NIST Voting Glossary

Election Terminology Glossary

2.1.8 Voluntary Voting Systems Guidelines (VVSG) Principles and Guidelines

EAC VVSG 2.0 Feberuary 2022

2.1.9 Technical Guidelines Development Committee (TGDC) Recommendations

- EAC Technical Guidelines Development Committee

  "The Technical Guidelines Development Committee (TGDC) assists EAC in developing the Voluntary Voting System Guidelines. The chairperson of the TGDC is the director of the National Institute of Standards and Technology (NIST). The TGDC is composed of 14 other members appointed jointly by EAC and the director of NIST."
  - https://www.eac.gov/about/technical_guidelines_development_committee

- NIST Voting Systems

  https://www.nist.gov/voting-systems

- Archived NIST Voting Methods Twiki, the Voting Methods Working Group (VMWG) 2015-2020
https://www.nist.gov/itl/voting/voting-methods-working-group-vmwg

3 Voting and Tabulation Method Domain Modeling

3.1 Authoritative References used in Voting and Tabulation Method Domain Modeling

In this section, we describe the references and input sources that this standard specification uses to acquire knowledge about voting methods used in practice in the United States for domain modeling.

3.1.1 Legislative References

Legislative sources include the election laws that govern how government-operated elections are operated in the United States.

3.1.2 Elections Official and Administrator References

To determine how elections are operated in practice, to understand use cases, or for a fine-grained understanding of how voting methods or voting variations are typically implemented, this specification includes the following types of documents as authoritative sources of information:

- Legislation and elections legal code defining voting methods and tabulation methods
- Elections administrator manuals for operating a local jurisdiction's elections
- Requests for Proposal (RFP) and Requests for Information (RFI) produced by a local jurisdiction Board of Elections or Department of Elections
- Elections Administrators and Elections Officials interviewed or consulted where detailed information or clarification on practice or rules was needed.

3.2 Tabulation Process Models Definitions

The following definitions are relevant for understanding the context and purpose of the tabulation methods models presented in this section.

Practical factors drove our definitions, distinctions drawn, and related modeling regarding voting methods, tabulation methods, and tabulation scenarios.

The Voting and Tabulation Methods Working Group (VMWG) defined voting methods, tabulation methods, and tabulation scenarios with the primary purpose of solving specific problems known to occur in practice. We identified particular issues to be solved, primarily issues that are known to arise that can create costly errors due to ambiguity in RFPs and election system provider contractual documents, as well as operator manuals. The VMWG also performed extensive research into the existing legislation that EOs, EAs, vendors, and other stakeholders are governed by when implementing election operations and systems.

The separation of the voting method concept from the tabulation method is based on the variety of tabulation scenarios that are implemented in practice.

1. Tabulation methods must be defined as a separate concept from voting methods because a voting method has a broader scope of functionality than a tabulation method, given real practical tabulation scenarios that are implemented widely by jurisdictions. Further, a voting method is, in part, composed of a set of one or more tabulation methods, where a key differentiator is that a voting method incorporates the decision rule set, which is applied to determine context outcome.

    a. In the set of common in-use tabulation scenarios, numerous of the scenarios do not and are not intended to determine the overall outcome of a contest. For example, some tabulation scenario variants do not and cannot result in the total contest count. An example use case is when a subset of a contest is tabulated for a precinct. The subset of the total contest count is to be reported but also combined or aggregated at a later point in the higher-level process with the subset counts of other precincts for a county-wide count. Another illustrative example is county-wide counts aggregated at the higher-level process for arriving at a state-wide count.

2. Multiple voting methods in practice (as well as in governing legislative definitions) use the same tabulation methods even though the higher-level, broader functionality, enclosing voting methods differ in ways required by their governing legislative and local rules in practice.

    a. For example, most voting methods use the N-of-M tabulation method.
        i. The means of making vote selections for variants that use the N-of-M tabulation method may differ, e.g., straight party or party preference.
        ii. The contest outcome decision rules for variants that use the N-of-M tabulation method may differ, e.g., between plurality and majority.
        iii. However, in each of the use-case categories - the tabulation method is the same.

Tabulation scenarios must be feasible to operationalize across the full set of voting methods variants for which they are applicable and across the full set of use cases for tabulation, including auditing, ballot accounting, capacity planning, and L&A testing, for example.

Voting Method
A voting method encompasses a set of partially ordered process steps, tabulation methods, and decision rules. Each voting method has at least one associated tabulation method and at least one election outcome decision rule set consisting of at least one decision rule. Executing the appropriate tabulation method and election contest decision rule set is in the scope of the voting method.

Tabulation Method
A tabulation method is a set of partially ordered process steps and operations for performing counting, accumulation, aggregation, and other data and math operations on vote selection data sets. Each tabulation method supports at least one tabulation scenario.

Tabulation Scenario
A Tabulation Scenario is a situation and sequence of events or activities intended by election administrators to be sufficient to achieve a tabulation goal.

See Section 4.0 within this standard for a more complete definition of Tabulation Scenarios and descriptions of all tabulation scenarios as use cases.

3.2.1 Tabulation Process Models Introduction

This section defines tabulation process models in the form of process flow charts and descriptions of the process flow for tabulation methods used in the United States. Sections are divided into two major components: the tabulation process flow charts themselves and detailed step-by-step descriptions of each step depicted in the flow charts. The tabulation process flows provide a generic high-level model for each tabulation method supported by this specification.

Tabulation process models are encapsulated by voting methods described in Section 4., and implement tabulation methods. This section provides definitions of tabulation methods as a set of process flows with steps that define the execution of a tabulation method. Each tabulation method is applicable to one or more tabulation scenarios and is used by one or more voting methods.

The tabulation methods in this section are representations of the plain language algorithmic definitions extracted from local jurisdiction legislation, rules, and other documentation that define tabulation methods, both explicitly and implicitly, as implemented in practice and operation by elections officials and election systems manufacturers. The tabulation methods process flows defined and illustrated in this section are based on a descriptive model of existing legislation, local jurisdiction rules, and practice rather than prescriptive.

For purposes of this section, adjudication processes are out of scope.

This specification supports all tabulation methods needed for use by all voting methods defined in the NIST CDF elections standards, specifications, and guidelines listed in this standard Appendix C., and all voting or tabulation methods defined in the EAC testing and certification manual as depicted in this specification Appendix C, VVSG Certification. Additionally, this standard supports all of the Tabulation Scenarios as defined by this standard in Section 4.

The High-Level Baseline Ballot Tabulation Process Flow figure (below) illustrates the generic meta-level tabulation process models in the form of process flow charts and descriptions of the specific process flow for tabulation methods used in the United States.

Tabulation Process Flow, Generic Baseline Ballot

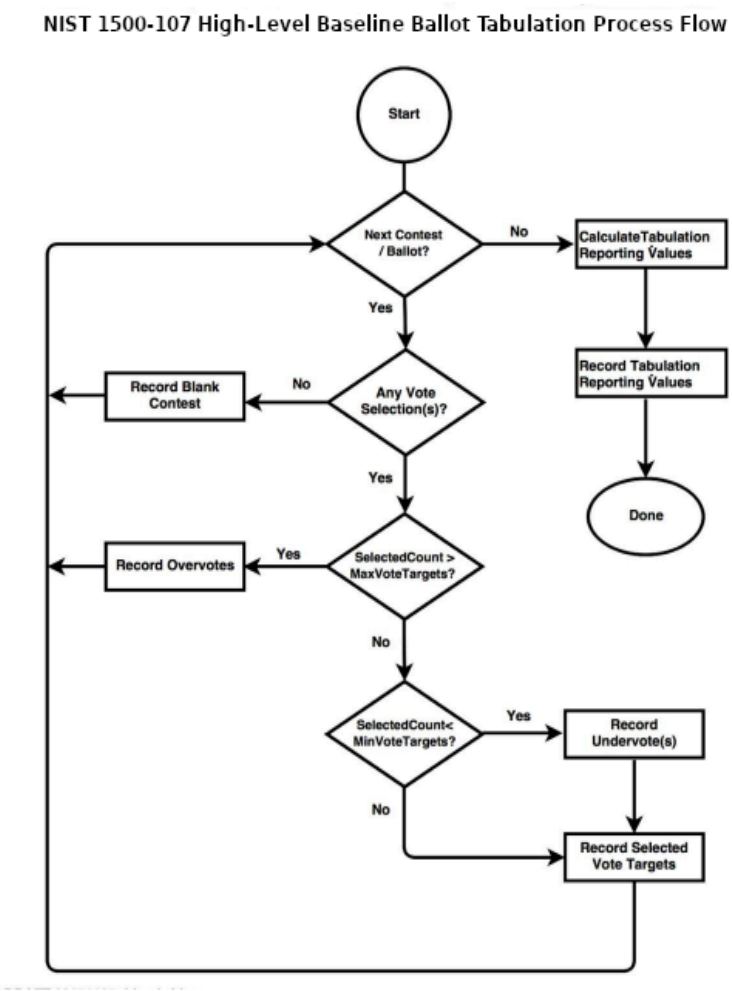


Figure 1. High level baseline ballot tabulation process flow.

Universally Applicable Rules

3.2.1.1.1 Treatment of Write-ins

Write-in is a candidate option that allows the voter to select a candidate not listed on the ballot. State law determines the contests where this option must/may be included. When the voter selects this option, the voter is then able to enter/write the name of the desired candidate in the designated candidate position of the contest on the ballot. Contests with write-in selection options are subject to the same tabulation method that is configured for the contest to which the write-in selection pertains. However, the treatment of write-in selections is differentiated from other

means of selections made by the voter. Most typically, write-in must be subject to a sub-process or separate distinct process as defined by the jurisdiction to determine validity as a selection. In some states, candidates who run as a write-in must follow a procedure to become a certified write-in. In these states, any entry of a candidate that is not certified is not counted. Write-in entries, if counted, in general, must be adjudicated to determine the candidate that the voter intended to select, as the entered name may not correspond to the candidate's official name. I.e., John Smith, John E. Smith, J. E. Smith, and Jack Smith may all be variations of John Edward Smith.

When write-in selections are made on a paper ballot, State rules determine whether a voter must both mark the write-in vote target and write in the desired name. Some states will automatically treat a write-in selection as an undervote if a name is written and the target is not also marked. Where a state allows counting a write-in without marking the target selection, some tabulation equipment can sense the presence of a written-in entry without the target being marked and will tabulate it as if the target was marked.

Write-in selections are usually initially reported in aggregate as a candidate named “Write-in” in each contest in which the write-in option appears. Once adjudicated, selections of invalid write-in candidates may be reported as undervotes in the associated contest. However, some states aggregate counts for these and report them in a scattering/miscellaneous category.

3.2.1.1.2 Treatment of Tiebreaking

In most cases, handling ties is a decision rule that comes into play post-tabulation. In some cases, ties must be handled during the tabulation process, e.g., during RCV rounds. Ties and other such decisions are subject to the applicable jurisdiction rules.

An election that results in a tie is an outcome preventing a candidate from being elected. Ties may occur irrespective of the tabulation method. A tie can happen in a single-seat contest where a plurality of votes is required, and two candidates get the same number of votes, or even if a majority of votes is required and there are only two candidates, they get the same number of votes. A tie can also happen in a multi-winner election if more candidates get the same number of votes than the number of seats remaining after seats are filled by candidates receiving more votes (if any) than the number in the tie. Typically, ballots are recounted during the canvassing process in the event of a tie to ensure that the number of votes received is indeed equal and correct.

In the event of a tie, jurisdiction rules determine how the tie is broken so a candidate or candidates can be declared a winner. In practice, tiebreaking methods usually involve a lottery-type process, such as drawing a high

card, flipping a coin, rolling a die or pair of dice, pulling numbers out of a hat, or drawing the long or short straw. The tiebreaking process is often performed by the election contest candidates themselves in a public setting and supervised by the appropriate election official.

Ties can also occur in an RCV tabulation process where a tie is encountered during candidate elimination in an RCV round. These are discussed in the RCV tabulation section below.

3.2.1.1.3 Treatment of Withdrawn or Disqualified Candidate

A candidate can be removed from the ballot for many reasons. These include:

- The candidate may decide to not run for the office and withdraw their candidacy.
- The candidate may die or otherwise become unable to hold office before the election.
- A legal challenge to the candidate's qualification for the ballot may result in a court order to remove the candidate.

How this is handled depends on the timing of the election process as well as jurisdiction- or state-specific rules. However, handling is independent of the voting method used to decide the election/contest. If removal occurs before printing or creation of ballots and before tabulator election file creation, there will be minimal impact. However, if a removal happens after ballot printing/creation and after installing files in the tabulators/voting units, there will be a significant impact. If removal occurs after printing/creating ballots but with enough time to recreate tabulator/voting unit files, the consequences are of medium impact. Note that state and local jurisdiction rules determine whether a reprint and file recreation is mandatory, depending on the timing of the election process or other state rules. If ballot reprinting does not occur or new files are not created for the tabulators/voting units, tabulation will happen as if the candidate is still active. Still, rules are applied to the disposition of the tabulated candidate's votes. Before the election, voters may be advised of this "logical" removal of the candidate choice and suggestion to choose another candidate or write-in as a substitute. Tabulated votes for the candidate may become undervotes.

There are several additional considerations regarding the impact on the operation of the election, including:

- If tabulator election definitions are to be changed, then media may need to be loaded with new files, and L&A testing may need to be re-executed.

- The appropriate course of action may depend on whether a correction is election-wide or related to a districted contest and only affects the units containing that district.
- When reprinting ballots must occur, the course of action may depend upon whether ballots that have already been sent for absentee, EarlyVoting, and overseas voters must be handled.

3.2.2 Tabulation Process Flow, High Level, By Contest

3.2.2.1 Introduction

The purpose of this process flow is to provide a generic, logical system architecture for contest-by-contest ballot processing flow that supports the case where ballots may contain multiple voting methods, each with one or more contests. Also, for voting methods or ballot layouts requiring it, ballots may include a method where the processing of a given contest is dependent on processing results from other contests. This process flow operates in conjunction with other tabulation process flows in this section, providing the detailed processing flow and descriptions for each of the contest types identified in this flow. When contests using a voting method are dependent on other contests using the voting method, contests are processed in a ballot order sequence to produce the correct outcome. A ballot may typically contain multiple contests with multiple voting methods. Therefore, for each contest on the ballot, a pass through the flow chart is performed.

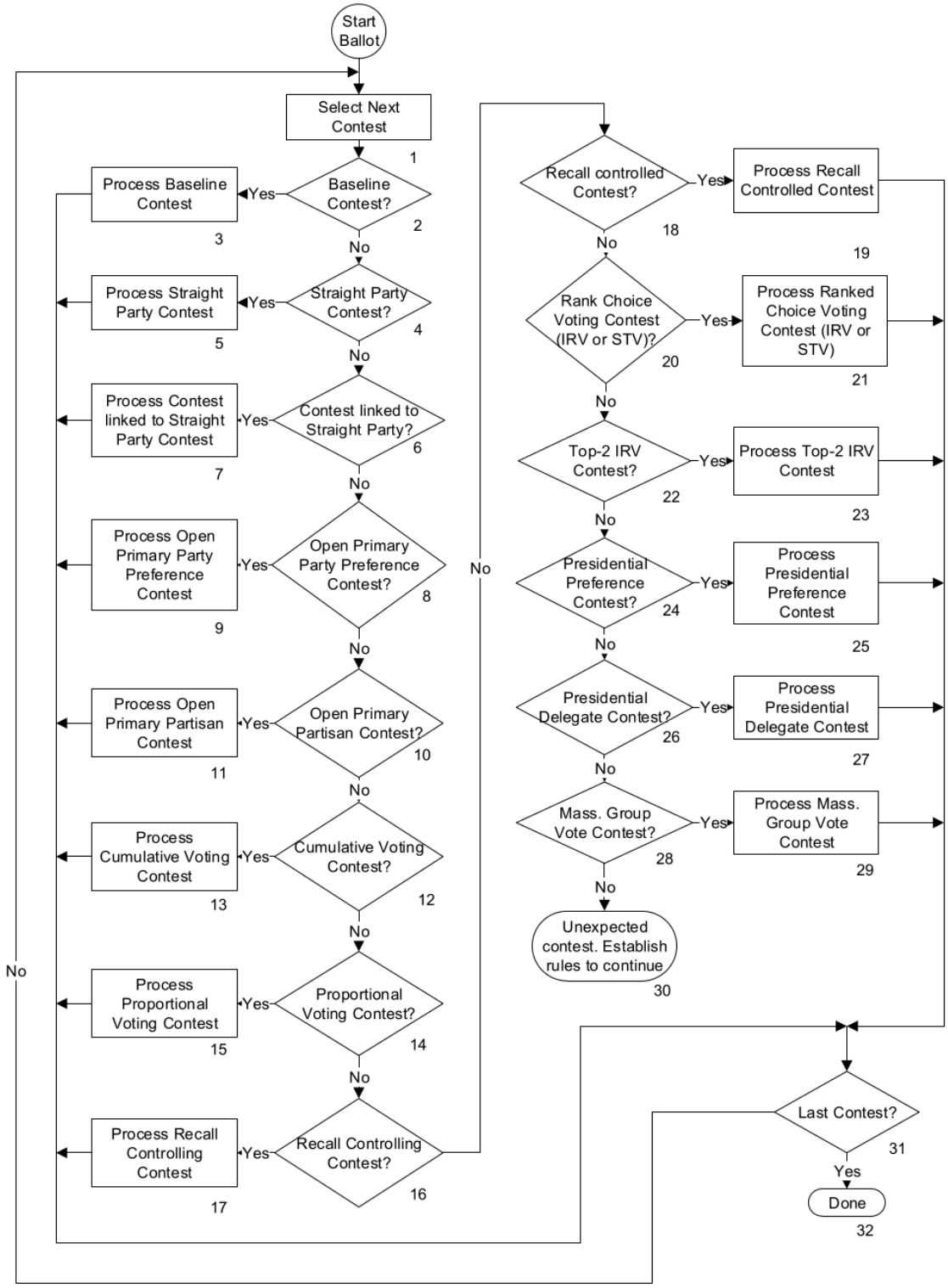


Figure 2 - Process Contests On A Ballot flow

Description

1. Select Next Contest

The next contest on the ballot is selected or the first contest is selected if processing is just beginning.

2. Baseline Contest?

A “Baseline Contest” is one whose tabulation does not interact with the tabulation of any other contest and has no special contest type identification. It would include any nonpartisan contest on a partisan or nonpartisan ballot as well as any contest on a General election ballot where straight-party voting is not applicable and any contest on a Closed Primary ballot (one that is issued based on party registration or party choice). Take the No path if the contest is not “Baseline.” Take the Yes path if it is.

3. Process Baseline Contest

Use the Baseline Contest Tabulation section for the low-level flow chart and description of processing a Baseline contest.

4. Straight Party Contest?

A straight-party contest appears on a General election ballot in states where straight-party voting is permitted. It is a contest that provides party selection. Take the No path if the contest is not a straight-party contest. Take the Yes path if it is.

5. Process Straight Party Contest

The straight-party voting method provides a voter with a single ballot selection to select all candidates on the ballot associated with a given party in all contests on the ballot. The straight-party contest contains the list of party selections. See the straight-party Voting Methods section for an overview of this method and its variants. See the Process straight-party Contest for the low-level flow chart and a description of processing a straight-party contest.

6. Contest Linked to Straight Party?

For a straight-party selection to control the selection of candidates of that party in other contests, those contests must programmatically tie to the straight-party contest. Take the No path if the contest is not tied to a straight-party contest. Take The Yes path if it is.

7. Process Contest Linked to Straight Party Contest

A contest tied to the straight-party contest allows a party selection in the straight-party contest to select all candidates associated with the selected party to be automatically selected. See the straight-party Voting Methods section for an overview of this method and its variants. See the Process Contest Tied to straight-party for the low-level flow chart and a description of how this contest is processed.

8. Open Primary Party Preference Contest

An Open Primary ballot contains partisan contests from all parties, but the voter can only choose to vote in contests from one party. If they vote in contests from more than one party, their entire

partisan ballot is spoiled. The Open Primary Party Preference contest is used in some states to allow the voter to select the party ballot they intend to vote in. If a voter makes a selection in the contest, only votes in contests of that party are subsequently considered. Take the No path if the contest is not a Party Preference contest. Take the Yes path if the contest is a Party Preference contest.

9. Process Open Primary Party Preference Contest

The Open Primary Party Preference contest is a nonpartisan contest that allows the voter to choose the party whose set of contests they plan to vote for. The contest only applies to an Open Primary ballot (one that contains the set of contests for all parties represented in the election). See the Open Primary Voting Method section for an overview of this method of voting. See the Process Open Primary Party Preference Contest section for the low-level flow chart and a description of processing this contest.

10. Open Primary Partisan Contest?

An Open Primary ballot contains partisan contests from all parties, but the voter can only choose to vote in contests from one party. If they vote in contests from more than one party, their entire partisan ballot is spoiled. Take the No path if this is not a Partisan contest on an Open Primary ballot. Take the Yes path if it is.

11. Process Open Primary Partisan Contest

An Open Primary contest is a partisan contest that appears on an Open Primary ballot, which contains a set of contests for all parties represented in the election. See the Open Primary Voting Method section for an overview of this method of voting. See the Process Open Primary Partisan Contest section for the low-level flow chart and a description of how this contest is processed.

12. Cumulative Voting Contest?

A Cumulative Voting contest is a multi-winner contest where a voter can distribute votes in whole-vote allotments to one or more candidates as long as the voter assigns no more votes than the number of candidates to be elected. Take the No path if the contest is not a Cumulative Voting contest. Take the Yes path if it is.

13. Process Cumulative Voting Contest

A Cumulative Voting Contest is one where the voter can assign the allowable votes to one or more contest options in whole-vote increments. See the Process Cumulative Voting Contest section for the low-level flow chart and a description of processing this contest.

14. Proportional Voting Contest?

A Proportional Voting contest is a multi-winner contest in which votes are proportionally allocated to selected candidates. It is also known as Equal and Even Cumulative Voting. Despite its name, Proportional Voting is considered a semi-proportional voting method.

15. Process Proportional Voting Contest

A Proportional Voting contest allocates the same number of votes to each selected candidate, equal to the votes allowed in the contest divided by the number of selected candidates, as long as the

number of selected candidates doesn't exceed the permitted number of selections. See the Process Proportional Voting Contest section for the low-level flow chart and a description of processing this contest.

16. Recall Controlling Contest?

A recall controlling contest is part of a linked contest pair. A vote selection in the controlling contest determines whether or not to tabulate the controlled contest. Take the No path if the contest is not a Recall Controlling contest. Take the Yes path if it is.

17. Process Recall Controlling Contest

The controlling contest in a linked pair of contests where a selection in this contest determines whether to tabulate the linked contest. See the Recall Contesting Control Voting Method section for an overview of this voting method. See the Process Recall Controlling Contest section for the low-level flow chart and a description of processing this contest.

18. Recall Controlled Contest?

Is the contest under the control of a specific Recall Controlling contest? Take the No path if the contest is not a Recall Controlled contest. Take the Yes path if it is.

19. Process Recall Controlled Contest

A contest whose tabulation is controlled by selections, or lack thereof, in the linked controlling contest. See the Recall Contest Controlled Voting Method section for an overview of this voting method. See the Process Recall Controlled Contest section for the low-level flow chart and a description of processing this contest.

20. Ranked Choice Voting Contest (IRV or STV, Multi-Pass IRV)?

Ranked Choice Voting (RCV) allows the voter to identify their order of preference of contest options for the designated contest. The contest can be a single-winner or multi-winner and uses a process known as Instant Runoff Voting (IRV) for single-winners and multi-winners or Single Transferable Vote (STV) for multi-winners. Take the No path if the contest is not an RCV contest or the Top 2 variation of an RCV contest. Take the Yes path if it is.

21. Process Ranked Choice Voting Contest (IRV or STV, Multi-Pass IRV)

Ranked Choice Voting (RCV) is a method of allowing the voter to identify their order of preference for candidates running for the designated office. RCV requires a candidate to have sufficient votes to exceed a threshold (typically 50% in a single-winner contest). There are several variations for processing Ranked Choice Voting (RCV) depending on jurisdiction rules. Some of these variations depend on whether the contest is a single-winner or a multi-winner contest. Some jurisdictions use tabulated 1st choice to determine whether RCV processing is required. RCV processing requires the use of a Cast Vote Record of the voted choices on each ballot cast or a ballot set, depending upon the applicable tabulation scenario. Algorithmic processing uses multiple rounds to eliminate candidates with the least votes and promote the next choice of a non-eliminated candidate to count in place of the eliminated candidate. This round-by-round process continues until the outcome criteria are met. See the Process Ranked Choice Voting Contest (IRV or STV) section for a more

detailed overview, the low-level flow chart, and a description of processing this contest with its different variations.

22. Top-2 IRV Contest?

Instant Runoff Voting (IRV) is a name used in an RCV voting method that applies to a single-winner contest, allowing the voter to identify the order of preference of candidates running for the designated office. There is a variation of IRV where only the top 2 vote-getters are eligible to win. This method is also known as the Contingent Vote. Take the Yes path if this contest uses the Top-2 variation of IRV processing. Take the No path if it is not.

23. Process Top-2 IRV Contest

Instant Runoff Voting (IRV) is a name used in an RCV voting method that applies to a single-winner contest, allowing the voter to identify the order of preference of candidates running for the designated office. There is a variation of IRV where only the top 2 vote-getters are eligible to win. This method is also known as the Contingent Vote. As with IRV, Top-2 IRV requires a candidate to have sufficient votes to exceed a threshold (typically 50% in a single-winner contest). However, the outcome of Top-2 IRV is determined in one runoff round of tabulation rather than, potentially, multiple rounds since all candidates but the top two vote-getters in first-choice selections are eliminated in the first round of tabulation. See the ProcessTop-2 IRV Contest section for an overall description of the Top2 method and the low-level flow chart that shows how the contest choices are processed, as well as a description of the blocks in the flow chart.

24. Presidential Preference Contest?

The Presidential Preference contest appears on a Presidential Preference Primary partisan ballot for a specific party. Party rules restrict the vote for delegates to their national nominating convention to candidates representing the presidential choice made by the voter. Take the No path if this is not a Presidential Preference contest. Take the Yes path if it is.

25. Process Presidential Preference Contest

The Presidential Preference contest is a partisan contest used on a Presidential Preference Primary ballot in states where a political party requires the selection of presidential candidates to restrict delegate selection to the nominating convention for those candidates representing their presidential selection. See the Presidential Delegate Voting Method section for an overview of this voting method. See the Process Presidential Preference Contest section for the low-level flow chart and a description of processing this contest.

26. Presidential Delegate Contest?

The Presidential Delegate contest is a partisan contest on a Presidential Preference Primary ballot where delegates to the party's National Nominating Convention are chosen from various categories but where the voter's choice is limited to the candidates that represent their presidential choice. Take the No path if this is not a Presidential Delegate contest. Take the Yes path if it is.

27. Process Presidential Delegate Contest

The Presidential Delegate contest appears on a specific party ballot in a Presidential Preference Primary where candidate selection is limited to candidates representing the presidential candidate selected in the Presidential Preference contest on the same ballot. See the Presidential Delegate Voting Method section for an overview of this voting method. See the Process Presidential Delegate Contest section for the low-level flow chart and a description of processing this contest.

28. Mass. Group Vote Contest
    A Massachusetts Group Vote contest is a contest for Town, Village, or Ward Committee and only appears on a Massachusetts Presidential Primary ballot. A large number of candidates are concurrently chosen, and candidates may run in groups with unique group selection positions embedded in the contest. Take the No path if this is not a Massachusetts Group Vote contest. Take the Yes path if it is.

29. Process Mass. Group Vote Contest
    The Massachusetts Group Vote contest only appears on a Massachusetts Presidential Primary ballot and is used to elect a large number of candidates (as many as 35) for a Town, Village, or Ward committee. Due to the large number of candidates and seats to be filled, candidates can run as affiliated groups (e.g., incumbents) with a group selection for each identified group to simplify voting for the group. See the Process Massachusetts Group Vote section for the low-level flow chart and a description of processing this contest.

30. Unexpected contest. Establish rules to continue
    A placeholder to cover the existence of a contest type associated with a voting method that this specification does not include. If encountered, it would cause contest processing of the ballot to cease. Rules would have to be determined for a new contest type and added to the flow diagram and explanation.

31. Last Contest?
    If the last contest on the ballot has been processed, take the Yes path. Processing is complete. If not, take the No path and select the next contest on the ballot for the next processing iteration.

32. Done. Processing of all contests on the ballot completed.

3.2.3 Process N-of-M, Baseline Contest

3.2.3.1 Introduction

This section describes the processing of a baseline N-of-M contest on a ballot, such as one that is part of "plurality," "majority," or "first past the post" voting, or voting related to a referendum that may be on a ballot. The differences relate to how the outcome is determined or whether a referendum passes at the global results level, not how to process individual ballots.

For this section, a baseline N-of-M contest is considered to be any contest that does not have special processing/tabulation rules associated with it. This category includes nonpartisan contests, partisan contests on a Closed Primary ballot, partisan contests on a General election ballot that does not contain straight-party voting, and

partisan contests on a General election ballot with straight-party voting but with no straight-party selection made.

The voting methods using the baseline N-of-M contest processing for the tabulation methods the voting methods implement are defined as follows; see section 4 below for the voting methods models.

Plurality Voting: A voting method that can apply to a single-winner or a multi-winner contest. Independent of the number of candidates running (single-winner) or winners (multi-winner), the winner is the candidate or candidates that receive the highest number of votes.

Substantial Plurality: A voting method that is a variation of Plurality in which a percentage threshold less than 50 percent is required for a winner to be declared, e.g., 40%.

First Past the Post (FPTP): A voting method that applies to a single-winner contest where the winner is the individual who receives the highest number of votes independent of the percentage of votes cast. It is a special case of Plurality Voting as it only applies to a single-winner contest.

Majority Voting: A voting method that applies to a single-winner contest where the winner must receive more than 50% of the votes cast (more than all other candidates combined) in the contest. If no candidate receives more than 50% of the votes, the two candidates receiving the most votes participate in a runoff election.

Approval Voting (single-winner) and Approval Block Voting (multi-winner): A voting method that applies to a contest where the winner(s) is the individual(s) that receives the highest number of votes. Voters in an Approval Voting contest are permitted to vote for (or "approve") as many or as few candidates as they want, regardless of the number of seats to fill. The allowed number of selections is equal to the number of candidates. For this reason, it is not possible to overvote an approval voting contest. Also, note that undervote tabulation is not relevant since a voter is not customarily expected to utilize all allowed vote selections as it indicates no preference.

Super-Majority Voting: A voting method that applies to a single-winner contest. The winner is the contest option that receives more than the super-majority of votes, defined variously to mean 60% of the vote, 2/3rds of the vote, more than 50% of the vote in each congressional district in a state, or other similar additional requirements.

Limited Voting: A voting method that applies to a multi-winner contest. Voters are limited to fewer vote selections than there are seats to be filled. Candidates receiving the most votes win the election.

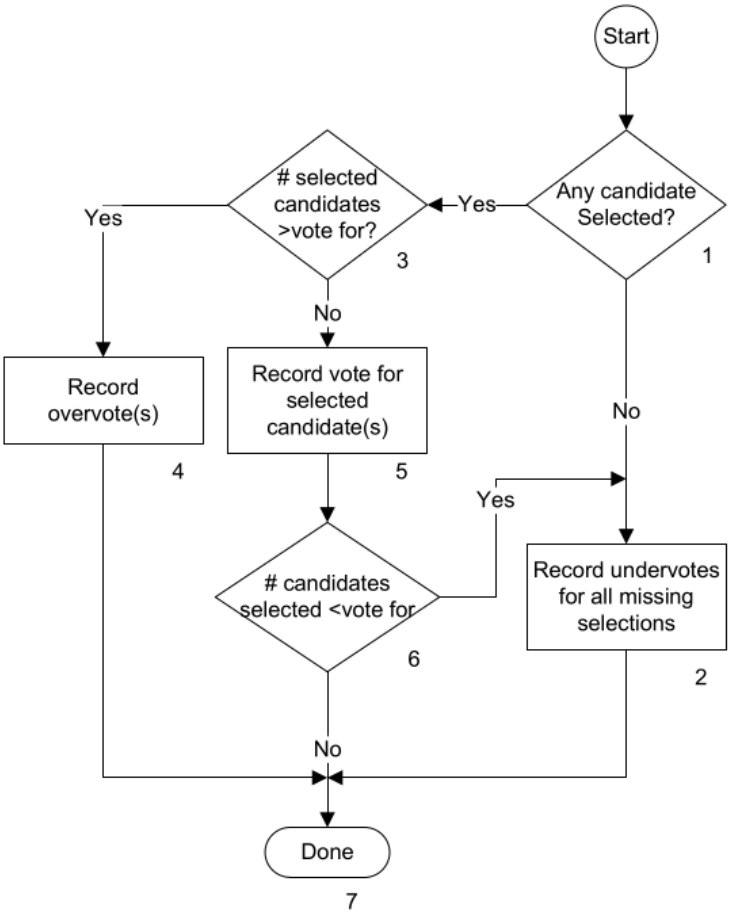


Figure 3 - Process N-of-M, Baseline contest process flow

Description

1. Any Candidate Selected?

Does the contest have any vote selections identified? If any selection is identified, then take the Yes path. Otherwise, take the No path.

2. Record Undervotes for all missing selections

Execute this process if no selections are identified in the contest or if it is a multi-winner contest, and fewer selections are made than allowed. Record an undervote for each missing selection.

3. # Candidates selected > vote for?

Compare the number of candidates selected to the number of allowed selections. Take the Yes path if the number exceeds the number allowed. Take the No path if the number is less than or equal to the number allowed. If the contest is a referendum or single-winner contest, the number allowed is 1. If it is a multi-winner contest, the number permitted usually is equal to the number of seats.

Exceptions are Limited Voting for multi-winner contests and Approval Voting for both single- and multi-winner contests.

4. Record Overvote(s)

Execute this process if the number of candidates selected in a contest exceeds the number of selections allowed. Record a number of overvotes equal to the number of selections allowed in the contest. The count indicates the number of votes lost due to the overvote. There is an exception to this rule that applies to the states of Hawaii and Illinois. In these states, the number of overvotes counted for a ballot is always one independent of the number of selections allowed. It reflects a count of the number of ballots containing an overvote for the contest. This difference only manifests itself in a multi-winner contest where votes are lost on a ballot, and ballots containing the overvote are not equal.

5. Record vote for selected candidate(s)

Execute this process if there are valid selections in the contest being processed. At least one selection is detected, and the number of selections does not exceed the allowed number of selections. Each selected candidate receives a vote.

6. # selected candidates < vote for

Compare the number of selected candidates to the number of selections allowed. If it is equal to the number permitted, take the No path. If it is less than the number allowed (which can only happen in a multi-winner context), take the Yes path.

7. Done

All processing for the contest on the current ballot completed.

3.2.4 Plurality, See N-of-M

3.2.5 Majority, See N-of-M

3.2.6 Super-Majority, See N-of-M

3.2.7 Approval, See N-of-M

3.2.8 Limited, See N-of-M

3.2.9 Process Straight-Party Contest

3.2.9.1 Introduction

This section describes the low-level processing associated with the contest on the ballot that allows the user to select a straight-party ticket. The straight-party contest may encompass several contests linked to on the ballot. Voters can indirectly select Candidates in all partisan contests through a single straight-party selection. Generally, the straight-party contest must precede the contests that are linked to it

or affected by it on the ballot. Note that straight-party processing rules are generally restricted to paper ballot processing where the voter does not additionally make direct candidate selections in each contest but may select a straight-party selection and only make exception selections within a contest. When the voter makes a straight-party selection and additionally makes direct or exception selection, then the tabulator must determine voter intent from the superposition of these selections based on jurisdiction-specific rules. DRE or BMD units generally show both direct selections and indirect selections when viewing a contest, preventing a voter from making a selection that violates the applicable rules.

There are two general rule sets for governmental elections. The differences come into play in contests where more than one choice is allowed. This specification designates those two rule sets as Inclusive and Exclusive. Inclusive rules allow any additional candidates selected by the straight-party selection to be included with directly selected candidates if the total number of candidates does not exceed the number of selections allowed. Exclusive rules exclude any straight-party selections if there is a directly selected candidate in a contest.

The flow chart below indicates the processing flow. Each block in the flow is described.

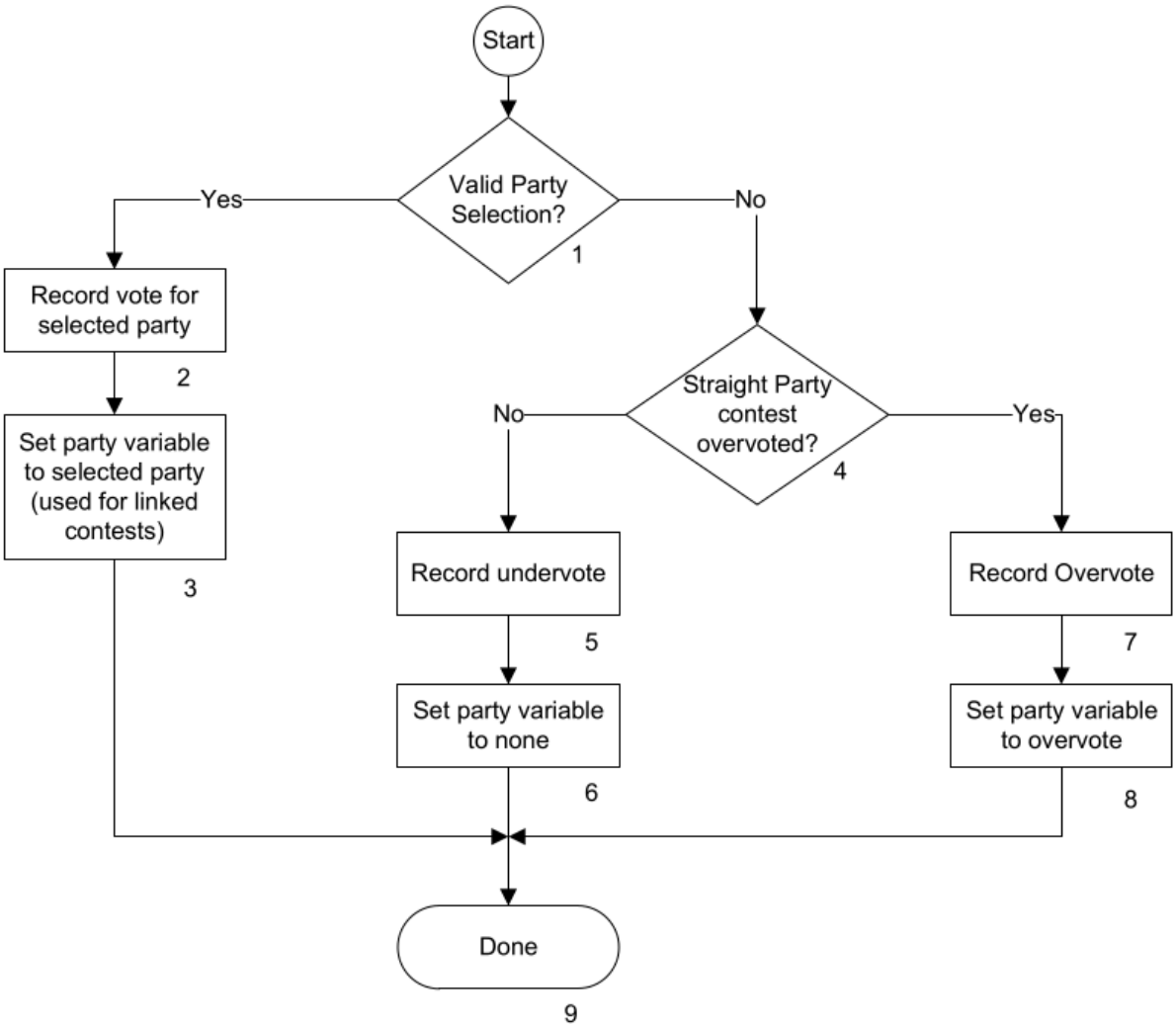


Figure 4 - Straight party contest process flow

Description

1. Valid Party Selected?

Determine whether there are any straight-party selections and whether they are valid (not more than one selection). Take the Yes path if there is, and take the No path if there is not. Note that, although no current rules use this, before eliminating straight-party voting, Illinois used to allow multiple straight-party selections and looked at the consequence for overvoting on a contest-by-contest basis. Illinois also used to allow multiple straight-party contests on a ballot in local elections, one for the County and one for the Township.

2. Record vote for selected party

Record a vote for the selected party to use it for reporting the number of ballots that selected each party. Some systems also record “true straight-party ballots” and remove the count of this selection if a voter directly selected any candidate in any contest that is not of the same party.

3. Set party variable for selected party

Set the party variable to the party selected by the voter. For flow chart purposes, the party variable term is used to remember the result of this straight-party contest processing to use it when processing selections in any contest tied to this straight-party contest.

4. Straight-Party contest overvoted?

Take the Yes path if there are more selections than allowed (typically more than one selection). Take the No path if there is no selection (if there is a valid selection, it is determined in block one, and this block is not reached).

5. Record undervote

Record an undervote for the contest for reporting and auditing purposes.

6. Set party variable to none

The party variable, used in processing contests tied to the straight-party selection, is set to none.

7. Set party variable to overvote

The party variable, used in processing contests tied to the straight-party selection, is set to overvote.

8. Done

Processing of the Straight party contest completed.

3.2.10 Process Contest Linked to Straight-Party Contest

3.2.10.1 Introduction

This section describes the low-level processing associated with a contest that is tied to/affected by a contest that allows a voter to make a straight-party selection. When voting a General Election paper ballot with the ability to vote straight-party, the only selections that usually appear on the ballot other than nonpartisan contest selections are the straight-party selection and any exception selections in partisan contests. The tabulator, using state-specific rules, must determine the candidates that the voter intends to select by looking at any straight-party selection in combination with any direct selection of candidates within contests. Section 3.2.9 describes Straight Party Voting Method in overview. Also reference Process Straight Party Contest for further information. For governmental elections, there are two general rule sets. The differences only affect contests where more than one choice is allowed. These rules are designated as Inclusive and Exclusive. Inclusive rules allow any additional candidates selected by the straight-party to be included with directly selected candidates if the total number of candidates does not exceed the number of selections allowed. Exclusive rules exclude any straight-party selections if there is a directly selected candidate in a contest. The flow chart below

along with the block by block descriptions explains the processing of these contests.

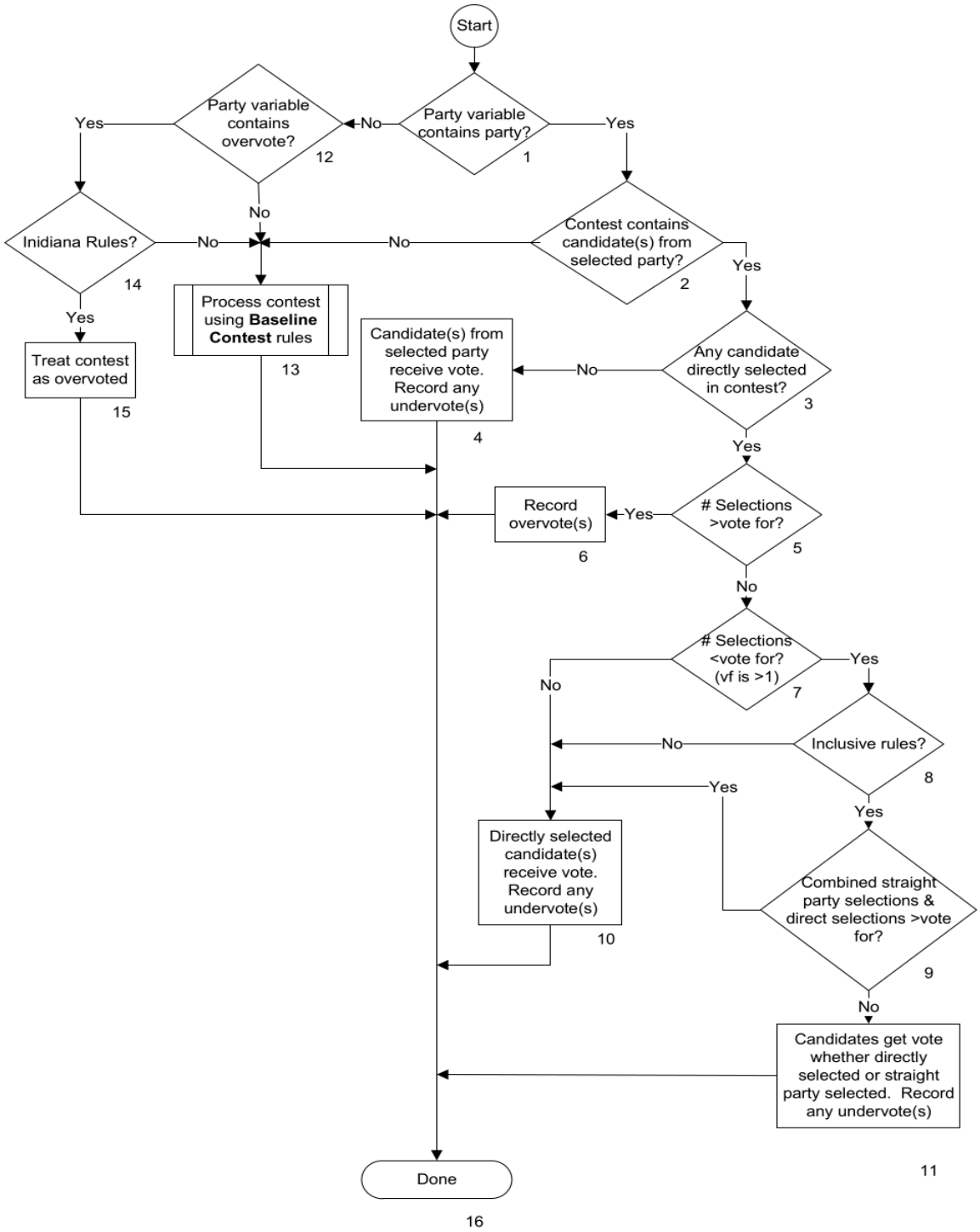


Figure 5 - Contest linked to straight-party contest process flow.

Description

1. Party variable contains party?

Processing of this contest requires knowledge of the result of processing the straight-party selection contest. For description purposes, the party variable is the term being used to capture this result. If the party variable contains a specific party, take the Yes path. If it does not, take the No path.

2. Contest contains candidates(s) of selected party?

Take the No path if the contest does not contain any candidates from the party selected in the straight-party contest as it is unaffected by the straight-party selection. Take the Yes path if the contest does contain one or more candidates from the selected party.

3. Any candidate selected in contest?

Take the No path if there are no direct candidate selections in the contest. Take the Yes path if there are one or more directly selected candidates.

4. Candidate(s) from the selected party receive vote

Since there are no candidates directly selected in the contest, any candidate associated with the straight-party selection receives a vote. In a multiple vote-for contest, if the slate for the party is not full (number of candidates is less than the number of seats), undervotes are recorded for all missing votes.

5. # Selections >vote for?

If the number of selections is greater than the vote for, the contest is overvoted independent of any straight-party selection. Take the Yes path. If not, then take the No path to determine candidate vote selections.

6. Record Overvote(s)

Record a number of overvotes equal to the count of total overvotes for the batch or contest

7. # Selections < vote for? (vf is >1)

If there are selections in the contest and the number is less than the allowed number of selections, the contest must be a multi-winner contest. Take the No path if the number of selections made is equal to the allowed number. Take the Yes path if the number is less than the allowed number.

8. Inclusive Rules?

At this point in the flow, we know that the contest is a multi-winner contest. We also know that there are one or more direct selections for candidates. However, the total count of selections amounts to less than the number allowed. Additionally, we know that there is a straight-party selection. The candidates receiving votes depends on the rules used for determining voter intent. If using Inclusive rules, take the Yes path. If using Exclusive rules, take the No path.

9. Combined Straight-Party selections & direct selections >vote for?

The candidates receiving votes using the Inclusive rules depends on whether the additional candidates selected by the straight-party selection when combined with the candidates directly

selected exceeds the number of selections allowed. If they do, take the Yes path. If the combined selections are less than or equal to the number allowed, take the No path.

10. Directly selected candidate(s) receive vote. Record any undervote(s).
Since adding straight-party selections of candidates to the directly selected candidates results in more selections than allowed, only direct selections receive votes. Straight party selections are ignored/discarded. Record undervotes for any unallocated votes.

11. Candidates get votes whether directly selected or straight-party selected. Record any undervote(s).
All selected candidates receive votes whether directly selected, selected by the straight-party selection, or selected by both. This is according to inclusive rules. Record undervotes for any unallocated votes.

12. Party variable contains overvote?
If the party variable indicates that the straight-party contest contains an overvote, take the Yes path. Take the No path if the straight-party contest does not contain an overvote.

13. Use Process Baseline Contest
Even though the contest is tied to the straight-party contest, if there is no straight-party selection, if the straight-party contest is overvoted but Indiana rules do not apply, or if a straight-party selection was made but there are no candidates in the contest from the selected party, the contest is processed as a Baseline contest. Use the Process Baseline Contest flow diagram at this point.

14. Indiana rules?
Take the Yes path if Indiana rules apply. Indiana rules state that any contest tied to the straight-party contest is overvoted if the straight-party contest is overvoted. Take the No path if they do not apply.

15. Treat contest as overvoted
As in #6, record a number of overvotes equal to the count of overvotes in the contest. According to Indiana rules, if the straight-party contest is overvoted, all partisan contests are also overvoted even if there are no partisan candidates in the contest (non-affiliated or write-in).

16. Done
Contest Processing completed.

3.2.11 Process Open Primary Party Preference Contest

3.2.11.1 Introduction

This section describes the low-level processing associated with a contest that, in some states that have Open Primaries (ballots that contain contests from all eligible parties on a single ballot), allows the voter to select the party whose ballot they intend to vote in the privacy of the voting booth. This is a nonpartisan contest that appears on the ballot before any partisan contest and, if a selection is made, protects the voter from spoiling the partisan section of the ballot. See Open

Primary Voting Method for a description of Open Primary with and without the use of the Party Preference Contest.

The flow chart below indicates the processing flow for the contest and each block in the flow is described.

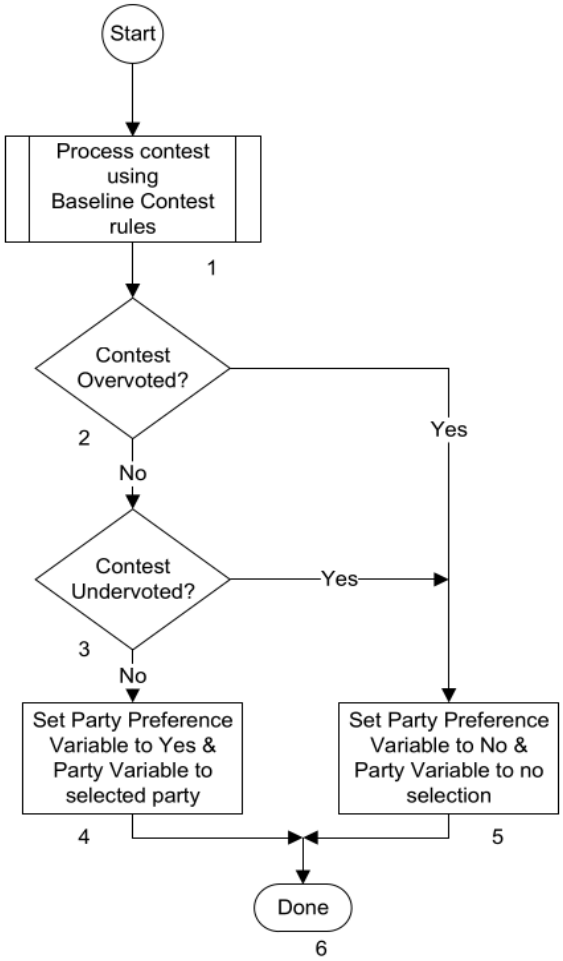


Figure 6 - Party preference contest process flow

Description

1. Process the Contest using Baseline Contest rules.

Although the result of processing this contest has special uses, the processing itself treats it like any other Baseline contest. Selections are always limited to one choice, so there are three possible processing results: a party selected, an undervote (no selection) or an overvote (too many selections).

2. Contest Overvoted?

Was the result of processing the contest an overvote? Take the Yes path if it was. Take the No path if it was not.

3. Contest Undervoted?
Was the result of processing the contest an undervote? Take the Yes path if it was. Take the No path if it was not.

4. Set Party Preference Variable to Yes & Party Variable to selected Party
Reach this block only if the voter made a valid party selection. Set the Party Preference Variable to Yes and set the Party Variable to that party. For flow chart purposes, the Party Preference Variable is the term being used to remember whether a voter made a valid selection in the Party Preference contest. Use the Party Variable term to identify whether the ballot is already associated with a particular party, which is needed when processing selections from partisan contests on the ballot.

5. Set Party Preference Variable to No and Party Variable to no selection
Reach this block if the contest was either not voted or overvoted. In either case, there is no valid party selection. Set the Party Preference Variable to No and Set the Party Variable to no selection.

6. Done
Processing of the Party Preference contest completed.

3.2.12 Process Open Primary Partisan Contest

3.2.12.1 Introduction

This section describes the low-level processing associated with a partisan contest on an Open Primary ballot. States that have Open Primaries (those where a single ballot contains contests from all eligible parties) allow the voter to select the party whose ballot they intend to vote in the privacy of the voting booth. However, the voter may not select contest options in contests associated with more than one party. If they do, the partisan portion of the ballot may be spoiled, and the ballot considered cross-voted. See Open Primary Voting Method for a description of Open Primary with and without the use of the Party Preference Contest, a contest that is included on the ballot to protect the voter from spoiling their ballot by making selections in multiple party contests.

Issues with handling Open Primary partisan contests are generally restricted to paper ballot processing where the ballot is submitted completed, and the tabulator must, therefore, determine the validity of the set of selections when performing the tabulation. DRE or BMD units do not have this issue as selections are validated before ballot acceptance and, for an Open Primary ballot, once a voter makes a partisan selection (or selects a party in a Party Preference contest), only that party's contests are available to be voted.

The flow chart below, along with the block by block descriptions, explains the processing of these contests.

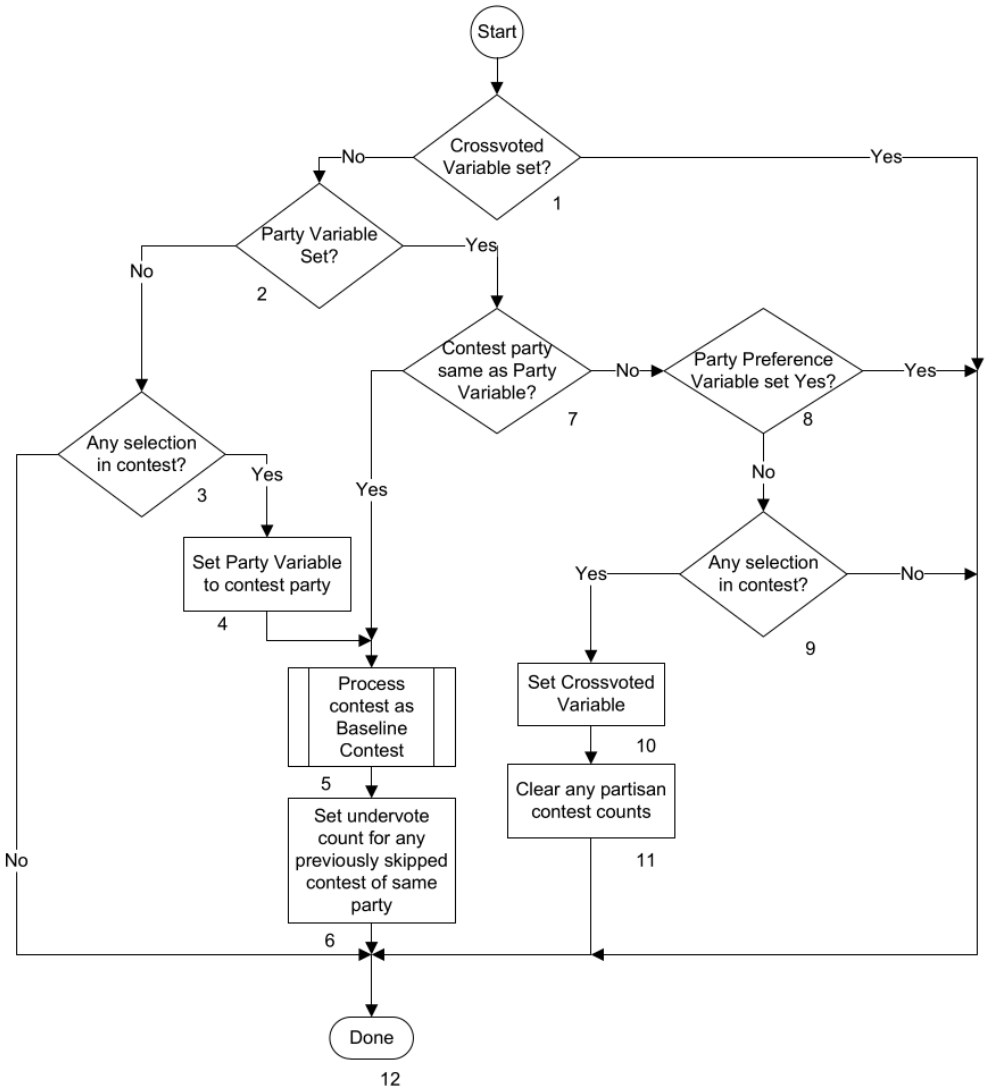


Figure 7 - Open primary partisan contest process flow

Description

1. Cross-voted variable set?

Processing an Open Primary ballot partisan contest requires knowledge of whether the ballot has already been spoiled based on processing of previous contests on the ballot. The cross-voted variable is the term used to remember this condition. If the crossvoted variable is already set, ignore any further processing of partisan contests. Take the Yes path if the variable is set. Take the No path if it is not set.

2. Party Variable set?

Processing an Open Primary ballot partisan contest requires knowledge of whether the ballot already has a party selection. Record this when encountering the first candidate selection in a

partisan contest or, if on a ballot with a Party Preference contest, a party has been selected. The Party Variable is the term being used to identify whether the ballot currently has a selected party affiliation. If the Party Variable is set, take the Yes path. If it is not set, take the No path.

3. Any selection in contest?

If there is no selection in the contest, take the No path (the contest is skipped and not processed) since the ballot is not yet associated with any party. If there is a selection, whether valid or not, take the Yes path.

4. Set Party Variable to contest party

Set the Party Variable to the party that is associated with the current contest. Any contest selection, whether or not valid, sets the ballot to be associated with that party.

5. Process contest as Baseline Contest

After determining that the contest will be processed, process it using the Baseline Contest processing flow. Any candidate votes, undervotes or overvotes are recorded as determined by the processing.

6. Set undervote count for any previously skipped contest of same party

In an Open Primary without a Party Preference contest, a ballot does not become associated with a party until a contest with a choice for a party is processed. When this occurs it is possible that contests of that party without any selections have previously been processed without being tabulated (skipped). Those contests should now be considered tabulated and the number of undervotes set to the number of votes allowed.

7. Contest party same as Party Variable?

Is the party associated with the current contest the same as the Party Variable (selected party associated with the ballot). If it is, take the Yes path and the process the contest using the Baseline Contest process flow. If it is not, take the No path.

8. Party Preference Variable set Yes?

Since the Party Variable does not match the party of the contest being processed, it makes a difference whether the Party Variable was set from a Party Preference contest. If the Party Preference Variable is set to Yes, the contest is skipped because it is not part of the selected party ballot. Take the Yes path. If it is not set to Yes, take the No path.

9. Any selection in contest?

If there is no selection in the contest, the contest is skipped and not tabulated as it is not part of the selected party ballot. Take the No path. If there is any selection, this becomes a cross-voted ballot. Take the Yes path.

10. Set Crossvoted Variable

Since there are selections made in contests associated with more than one party and there is no protection from a party selection made in a party Preference contest (which may not even be on the ballot), the Crossvoted Variable is set to prevent any further processing of partisan contests.

11. Clear any partisan contest counts

Clear all tabulated results from any previously processed partisan contests (including over- and undervotes). The ballot is now a crossvoted/spoiled ballot and nothing partisan is tabulated.

12. Done

All processing of this contest completed.

3.2.13 Process Cumulative Voting Contest

3.2.13.1 Introduction

Cumulative Voting is a voting method that applies to a multi-winner contest that allows a voter to distribute the allowed number of votes to one or more candidates in whole-vote allotments. Each voter is allowed to distribute a number of votes equal to the number of candidates to be elected. Each candidate is associated with as many contest options on a ballot as the number of votes allowed in the contest. The ballot design may be implemented by associating multiple voting positions for a single candidate listing or by repeating the candidate listing in a way that identifies it as the same candidate. Tabulation treats the contest as a Baseline contest, but more than one vote can be allotted to a given candidate.

The flow chart below indicates the processing flow for the contest, and each block in the flow is described.

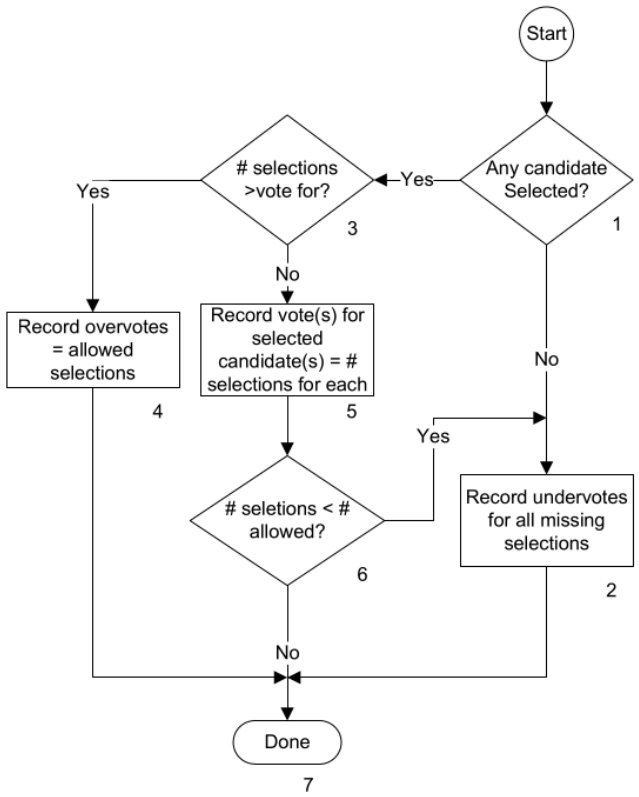


Figure 8 - Cumulative voting process flow

Description

1. Any Candidate Selected?

Is there a vote selection for any candidate in the contest? If there is any candidate selection, take the Yes path. Otherwise, take the No path.

2. Record Undervotes for all missing selections

Execute this process step if there are no selections identified in the contest or if there are fewer selections made than allowed. Record an undervote for each missing selection.

3. # selections > vote for?

Compare the number of total selections made (including multiple selections for the same candidate) to the number of allowed selections. Take the Yes path if the number exceeds the number allowed. Take the No path if the number is less than or equal to the number allowed.

4. Record overvotes = allowed selections

Execute this process step if the number of selections made exceeds the number of selections allowed. Count a number of overvotes for the contest equal to the number of selections allowed in the contest. The count essentially indicates the number of votes lost due to the overvote.

5. Record vote(s) for each selected candidate(s) = # selections for each

Since the number of selections made in total is less than or equal to the number allowed, each selected candidate receives a number of votes equal to the number of selections made for that candidate.

6. # selections < # allowed

Compare the number of total sections made to the number of selections allowed. If it is equal to the number allowed, take the No path. If it is less than the number allowed, take the Yes path.

7. Done

All processing for the contest on the current ballot has been completed.

3.2.14 Process Proportional Voting Contest

3.2.14.1 Introduction

Proportional Voting is a voting method that applies to a multi-winner contest. Similar to a Cumulative Voting contest, it allows a voter to allot more than one vote to a candidate. However, it differs from Cumulative Voting in that the voter cannot show differing levels of support for different candidates. As long as the number of selected candidates is less than or equal to the number of allowed selections, tabulation proportionally allots an equal number of votes to each selected candidate based on the number of votes allowed and the number of candidates selected. Note that this number can be fractional. Proportional Voting is also known as Equal and Even Cumulative Voting. Proportional Voting is considered a semi-proportional voting method.

Proportional Voting was used in electing representatives to the Illinois General Assembly for 110 years until its elimination in 1980. It applied to both the Primary and the General and the voter had 3 votes to distribute. Voters could give three candidates 1 vote each, give two candidates 1½ votes each or give one candidate 3 votes.

The City of Peoria Illinois has used Proportional Voting to elect 5 at-large members to their City Council since 1991. If a voter selected 1, 2, 3, 4 or 5 candidates, each candidate would get 5, 2½, 1 2 /3, 1¼ or 1 vote respectively.

The flow chart below indicates the processing flow for the contest and each block in the flow is described.

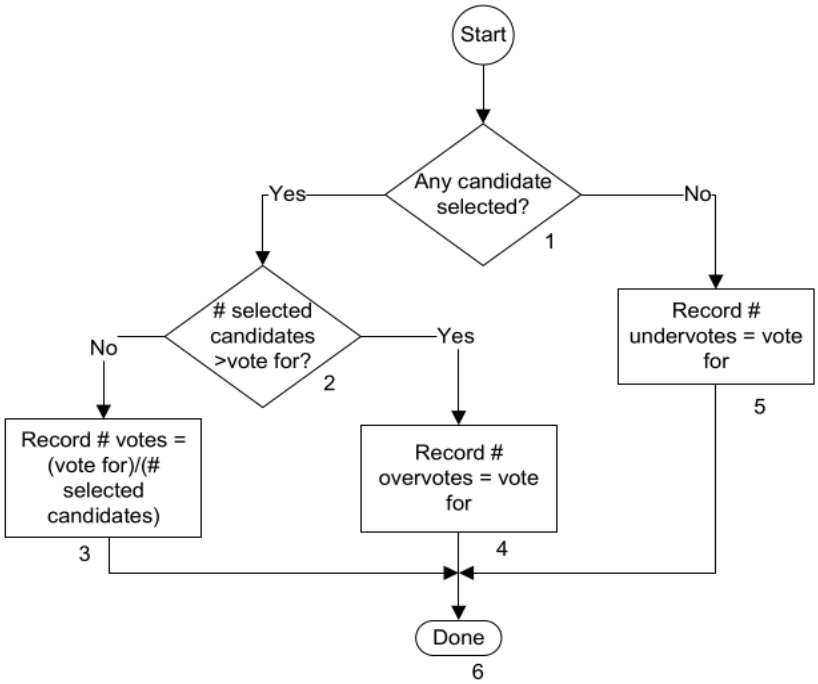


Figure 9 - Proportional Voting process flow

Description

1. Any candidate selected?

If there is no selection in the contest, take the No path. Take the Yes path if the voter selects at least one candidate.

2. # selected candidates > vote for?

Take the Yes path if the number of candidates selected is greater than the number of allowed selections ("vote for"). Take the No path if the number is less than or equal to the number of allowed selections.

3. Record # votes = (vote for)/(# selected candidates)

The votes are proportionally distributed and recorded equally among the selected candidates. The number of votes is equal to the number of votes allowed divided by the number of selections. This will be a whole number of votes if there is only 1 selection (candidate gets all allowed votes) or the number of selections is equal to the votes allowed (each gets 1 vote). In all other cases, the recorded vote may contain fractional votes.

4. Record # overvotes = vote for

Since there are more vote selections than allowed, record a number of overvotes equal to the allowed number of votes.

5. Record # undervotes = vote for

Since there are no vote selections, record a number of undervotes equal to the allowed number of votes. Note that undervotes can never be recorded if there is a selection.

6. Done.
Processing of this contest completed.

3.2.15 Process Recall Controlling Contest

3.2.15.1 Introduction

This section describes the low-level processing associated with a contest controlling whether the controlled contest is eligible to be tabulated.

For a recall controlling contest, there is a contest pair, one controlling and the other controlled that are traditionally called Recall. See the Recall Controlled Contest section below. The Controlling contest must be voted according to one of two rules to tabulate the Controlled contest. Otherwise, it is ignored for processing. Depending on the rules, it is either any vote in the controlling contest or a Yes vote. In traditional use, the Controlling contest is in a referendum format. In the voting method decision outcome if the controlling contest outcome fails, then the controlled contest becomes moot and the voting method decision rule is not executed to determine outcome.

The flow chart below indicates the processing flow for the Recall Controlling contest and each block in the flow is described.

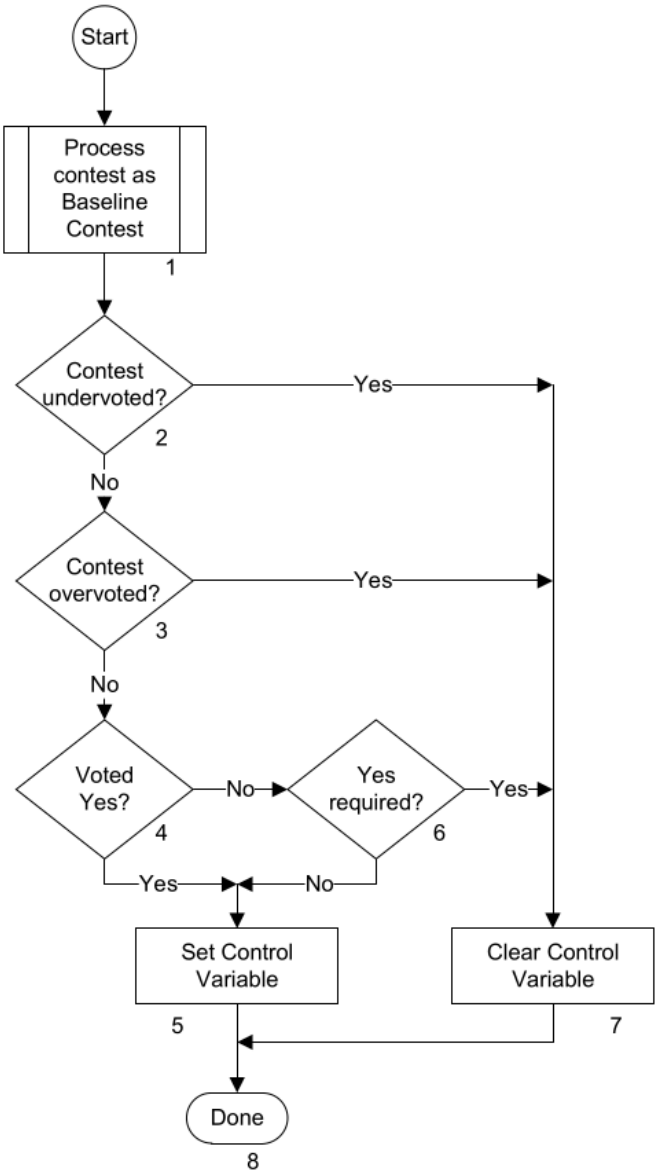


Figure 10 - Recall controlling contest process flow.

Description

1. Process contest as Baseline Contest

The controlling contest is normally a Baseline contest and is processed using the Baseline Contest flow. In traditional use, it is a referendum on recalling a candidate.

2. Contest Undervoted?

If the contest is undervoted, take the Yes path as the vote does not satisfy the requirement to allow tabulation of the controlled contest. If it is not undervoted, take the No path.

3. Contest Overvoted?
If the contest is overvoted, take the Yes path as the vote does not satisfy the requirement to allow tabulation of the controlled contest. If it is not overvoted, take the No path.

4. Voted Yes?
This flow assumes that the contest is a referendum. Take the Yes path if there was a Yes vote or the equivalent thereof (e.g., For). Take the No path if the selection was a No or the equivalent thereof (e.g., Against).

5. Set Control Variable
Since tabulation of this contest is controlling the vote processing in another contest (the Controlled Contest), for description purposes, the Control variable is the term being used to capture the result of tabulating this contest. The Control Variable is set to allow tabulation of the Controlled contest.

6. Yes required?
Entering this test, we know that there was a valid vote, but it was not Yes or an equivalent. This test establishes whether jurisdiction rules require a Yes vote in the Controlling contest to allow tabulation of the Controlled contest. Take the Yes path if it is required thus not allowing tabulation of the Controlled contest. Take the No path if it is not required thus allowing tabulation of the Controlled contest.

7. Clear Control Variable
Clear the Control Variable, preventing tabulation of and effectively skipping the controlled contest.

8. Done
Processing of the contest completed.

3.2.16 Process Recall Controlled Contest

3.2.16.1 Introduction

Recall elections are made up of a pair of contests, one controlling and the other controlled. This section describes the low-level processing associated with the contest that is being controlled by the Recall Controlling contest. See the Recall Controlling Contest, above, section for a full description. The Controlling contest must be voted according to 1 of 2 rules in order to tabulate the Controlled contest; otherwise, it is ignored for processing. Depending on the jurisdiction's rules, this means either any vote in the Controlling contest or only a Yes vote in the Controlling contest. In traditional use, the Controlling contest is in a referendum format.

The flow chart below indicates the processing flow for the Recall Controlled contest and each block in the flow is described.

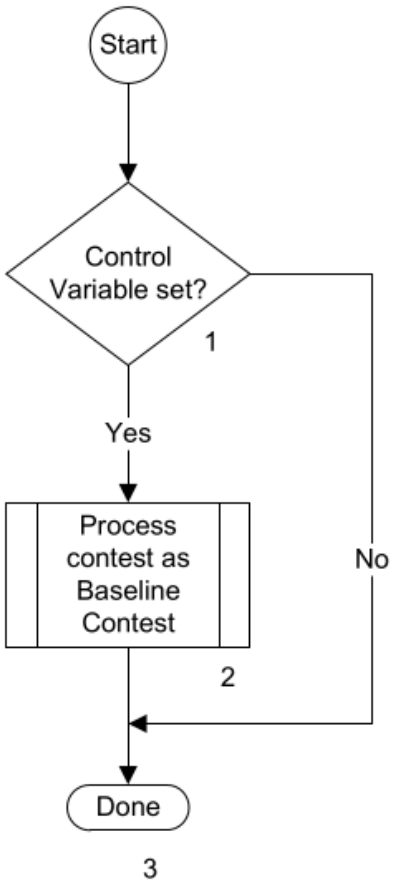


Figure 11 - Recall controlled contest process flow

Description

1. Control Variable set?

Since tabulation of this contest is controlled by vote processing in another contest (the Controlling Contest), for description purposes, the Control variable is the term being used to capture the result of tabulating the Controlling contest. If the Controlling contest set the Control Variable, indicating that the requirement for tabulation has been satisfied, take the Yes path. Otherwise, take the No path; no tabulation (including undervotes) occurs.

2. Process contest as Baseline Contest

The contest is processed as it would be if it were not controlled. Typically this would be as a Baseline contest with no other special rules applied.

3. Done

Processing of the contest completed.

3.2.17 Process Ranked Choice Voting (IRV, STV, Multi-Pass) Contest

3.2.17.1 Introduction

Ranked Choice Voting (RCV) includes voting methods where the voter may identify their contest option choices in order of preference. Ranked choice voting

elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and transferring surplus votes from elected candidates. Counting RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled.

In some STV, or multi-winner variations of RCV, candidates who receive more votes than necessary to be elected – those candidates whose vote totals exceed the threshold of election – are deemed to have surplus votes. If a candidate has a surplus, the next step in counting is to transfer that surplus. Surplus votes are transferred in various ways, but they follow the same logic as elimination transfers: each ballot counting for a candidate with surplus votes transfers vote value to the next highest-ranked continuing candidate on that ballot unless there is an intervening selection that disqualifies the ballot.

There are a number of variations used in processing contest selections to determine the candidates that are elected. Voting the ballot is the same amongst the variants although the presentation of ballot design may vary. This document addresses the methods currently in use in the United States, which fall into three major categories: single-winner Ranked Choice Voting (RCV), multi-winner Ranked Choice Voting and Bloc Preferential Voting (BPV), also known as Multi-Pass IRV. Single-winner RCV is also known as Instant Runoff Voting (IRV). Multi-winner RCV refers to Single Transferable Vote (STV) and Multi-Pass IRV. All use multi-round by round processing methods to determine the outcome(s). Purposes of these methods include avoiding holding a separate runoff election, eliminating the effect of vote splitting or, when using single-transferable vote, implementing proportional representation in multi-winner contests.

3.2.17.2 Ranked Choice Voting (IRV, STV)

The processing used in each IRV round, if there is no apparent winner, is as follows. The candidate with the lowest vote total in the round is considered eliminated. Each CVR containing the eliminated candidate as the current 1st choice is processed to count for the next highest-ranked continuing candidate (one that has not been eliminated). If there are no choices left on a given ballot or if a disqualifying ranking occurs prior to the next ranked continuing candidate, that ballot is considered exhausted and inactive for any subsequent rounds. The new round tabulates totals and determines whether a candidate now has sufficient votes

to be elected. If not, the process is repeated round-by-round until an elected or multiple elected are determined.

In STV elections, candidates who receive more votes than necessary to be elected – those candidates whose vote totals exceed the threshold of election – are deemed to have surplus votes. If a candidate has a surplus, the next step in counting is to transfer that surplus. Surplus votes are transferred in various ways, but they follow the same logic as elimination transfers: each ballot counting for a candidate with surplus votes transfers vote value to the next highest-ranked continuing candidate on that ballot unless there is an intervening selection that disqualifies the ballot.

3.2.17.3 Multi-Pass IRV/Bloc Preferential Voting

Multi-Pass IRV (Bloc Preferential Voting), uses multiple iterations of the single-winner IRV tabulation process for multi-winner contests by electing candidates one at a time. After a candidate is determined to be elected, that candidate is removed from the list of continuing candidates before running a new round-by-round tabulation with the original ballots. This method is a majoritarian voting method and is not designed to produce proportional representation, in contrast to STV which was designed with the intent to provide proportional representation.

There are several variants of each of the IRV and STV tabulation methods. Variants may be grouped by how a variant treats handling of overvotes, repeated ranking, candidate elimination, thresholds, ties and other factors as for the following examples.

- Handling an overvoted choice during round-by-round processing – Is the ballot considered exhausted or is the choice skipped and the ballot checked for a subsequent continuing candidate?
- Handling an omitted choice/ranking during round-by-round processing – Is the choice skipped? Does this cause the ballot to exhaust, are skipped ranking ignored or is one skipped ranking ignored but two skipped ranking in succession exhaust the ballot?
- Handling of a repeat ranking (selecting the same candidate for more than one ranking) during the round-by-round processing – Is the repeated rank skipped in the same way as an omitted ranking or is the ballot considered exhausted?
- Handling of candidate elimination – Are all candidates who have no mathematical chance of advancing concurrently eliminated in the first round only, in any round or is only one candidate eliminated in any round and multiple eliminations of candidates not used?
- Handling of tabulated voted 1st choices – Is tabulation of voted 1st choices used to determine if a candidate(s) has sufficient votes to be elected (thus

avoiding the use of an RCV algorithm) or is the algorithm always used? If used, does calculation of the threshold for election include all ballots cast (include over- and undervote totals) or is it based on total valid selections?

The flow chart below indicates the processing flow for either a single- or multi-winner contest with each block in the flow described.

Process Ranked Choice Voting Contest (IRV or STV)

(below)

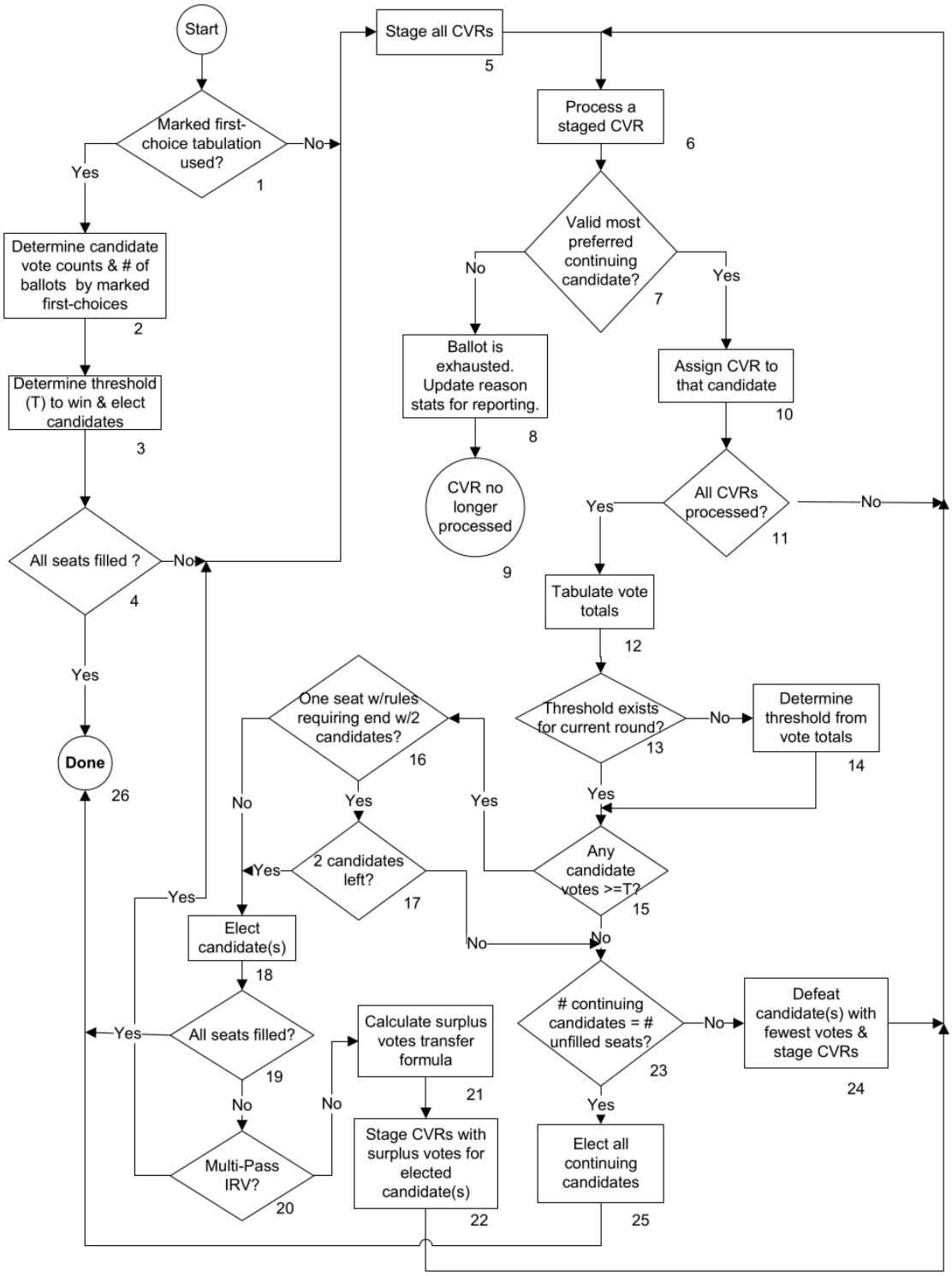


Figure 12 - Ranked choice voting contest process flow

Description

1. Marked first-choice tabulation used?

Some RCV tabulation rules require (St. Paul, Minneapolis and Utah) or conditionally allow (Oakland) a shortcut that can elect candidates by looking only at marked first choices. This can help avoid what may be a more logistically involved process of collecting and examining CVRs to determine effective first choices. If a tabulation of marked first choices elects sufficient candidates, the full RCV tabulation process is not needed, and an EA system may avoid examination of CVRs. Some other jurisdictions use this shortcut as a matter of practice.

If a ballot has a validly marked first choice, that is also the ballot's effective first choice. However, 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. Detailed rules for what counts as an effective first choice can vary by jurisdiction. See the description for block 6 for additional details.

Take the Yes path if using an initial tabulation of marked first choices. Take the No path if the CVRs are used to initially determine the effective first-choice candidate votes.

2. Determine candidate vote counts and number of ballots by marked first-choices.

Votes are tabulated by considering only marked first-choice ballot selections and as if the contest were a vote-for-one contest. If the first choice on a given ballot is not marked or contains an overvote, there is no contribution to candidate votes. The number of ballots used as the base for the threshold calculation is also determined based on jurisdiction-specific rules or practice. St. Paul uses the total number of ballots with validly marked first choices that count for candidates. Minneapolis uses the number of ballots cast. The Oakland rule uses the number of ballots cast except for those with a marked overvote.

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

The threshold identifies how many votes a candidate must have in order to be elected based on the tabulation of marked first choices. If a contest is only electing one candidate, the threshold is typically a majority of the threshold base so that a candidate would need more than 50% of the threshold base to be elected. The static threshold formula for any number of candidates to be elected is:

$T=(V/(N+1))+1$ rounded down to the nearest whole number

where T is the threshold, V is the number of relevant votes, and N is the number of candidates to be elected. Thus in a contest where only one candidate is to be elected, the static threshold is Rounddown0(1 + V/2), and a candidate must have at least that many votes to be elected. In a contest with 4 candidates to be elected, the static threshold would be Rnddown0(1 + V/5), and a candidate must have at least that many marked first-choice votes to be elected, i.e., more than 20%, since the number of marked first-choice votes is a whole number.

Once the threshold and threshold criterion are established, any candidates satisfying that criterion are considered elected for the purposes of this tabulation of marked first choices.

4. All seats filled?

Are sufficient candidates elected because their number of marked first-choice votes is greater than (or equal to, for Minneapolis) the threshold to fill all seats in the contest? Take the Yes path if there are sufficient candidates. Take the No path if one or more positions remain unfilled due to insufficient candidates obtaining the required number of marked first-choice votes. In a single-winner contest, taking the Yes path requires that one candidate be elected. If the No path is taken, a full RCV tabulation is conducted, without using the results of the tabulation of marked first choices. In this scenario candidates elected by marked first choices are considered unelected continuing candidates at the beginning of the full RCV tabulation. If taking the Yes path, the full RCV tabulation is not required.

5. Stage all CVRs

A full RCV tabulation requires every CVR to be examined to determine which, if any, candidate the CVR counts for in the first round. This block stages all CVRs for that determination. All candidates begin the first round as continuing candidates, neither elected nor eliminated, without any votes. If this block is entered from the Yes path of block 20 ("Multi-Pass IRV?"), any candidate already elected is no longer a continuing candidate when all CVRs are staged.

6. Process a staged CVR

A single CVR taken from a collection of staged CVRs is processed to determine the candidate for which the ballot will next count, ie., the CVR's highest-ranked continuing candidate if such a candidate exists. A continuing candidate is a candidate that is neither eliminated (aka defeated) nor elected.

Stage CVRs for processing in this step come from three sources: 1) in block 5, all CVRs are staged for round 1, 2) in block 24 after defeating a candidate, and 3) for multi-winner contests in block 22 after electing a candidate with surplus votes, the votes for a candidate in excess of the threshold. Block 11, "All CVRs processed," checks whether any CVRs remain to be processed. Typically, the CVRs counting for a candidate are staged for reassignment after that candidate is eliminated or elected and before the next round's vote counts are tallied, unless a next round is not needed.

Determining the highest ranked (most preferred) continuing candidate can be fairly straight forward if the voter has ranked candidates in order of preference. However there are several irregular ranking situations that a voter might mark and which are treated according to jurisdiction-specific rules: 1) no candidate ranked at a ranking level, 2) ranking more than one candidate at a ranking level, and 3) ranking a candidate at more than one ranking level. The following describes how various jurisdictions treat these situations:

- Unvoted choice (no candidate ranked at a ranking level)
  - Ballot considered exhausted
  - Ballot 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)
  - Ballot considered exhausted

- Not considered overvoted if doesn't contain more than one continuing candidate (Takoma Park, MD)
  - Skipped and subsequent choice processed, if any
- Repeated ranking of a previously ranked candidate
  - Ballot considered exhausted
  - Skipped and subsequent choice processed, if any

7. Valid most preferred continuing candidate?

Take the No path for the processed CVR when there is no valid selection for a continuing candidate, ie, when the ballot is exhausted. Take the Yes path if there is a valid selection for a continuing candidate.

8. Ballot is exhausted. Update reason stats for reporting

This CVR is not included in any further tabulation processing as there are no more validly ranked continuing candidates available. This may be due to the CVR not containing further ballot ranking selections or that all subsequent ranking selections are for eliminated or already elected candidates. Record the reason for the ballot being considered exhausted for purposes of reporting round results.

9. CVR no longer processed

Since the CVR/ballot does not contain a valid subsequent choice for the highest-ranked continuing candidate, that ballot is not subject to further tabulation. The exception to this rule is that, in Multi-Pass IRV/Bloc Preferential Voting, all CVRs are restaged for processing (as described under block 5), including those that were exhausted in a previous tabulation, as tabulation begins for each seat to be filled.

10. Assign CVR to that candidate

The CVR is assigned to count for its highest-ranked (most preferred) continuing candidate. The ballot contributes one full vote or a surplus fraction of a vote to that candidate's vote total. As noted above, a surplus fraction of a vote may be used in a multi-winner contest.

11. All CVRs processed?

Take the No path to process the next staged CVR if some staged ballots remain to be processed. Take the Yes path if all staged ballots have been processed.

12. Tabulate vote totals

Tally vote totals for each candidate and for any other reporting categories.

If there was a tabulation of marked first choices in block 2, the candidate vote totals for round 1 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.

13. Threshold exists for current round?

Take the No path every time for a single-winner contest. For multi-winner contests, take the No path only for the first round. Take the Yes path for the second and subsequent rounds of a multi-winner contest which reuses the first-round threshold. Typically, a threshold from a tabulation of marked first choices is not reused here, especially if that tabulation used a different threshold base.

14. Determine threshold from vote totals

For single-winner contests, jurisdictions typically use the total votes counting for candidates in the round as the threshold base, i.e., the number of ballots used to calculate the threshold. Jurisdictions typically express the threshold for such contests in terms of a majority of that threshold base, i.e., more than 50%. Most jurisdictions update the threshold in single-winner contests every round.

- If X is one, a contest option can reach the threshold to be elected; in addition, all fractions are ignored.
- If X is zero, a contest option must exceed the threshold to be elected.For multi-winner contests, the threshold can be expressed as:

$$T = (V / (N + 1)) + X, \text{ where } X : \{0, 1\}$$

where T is the threshold, V is the threshold base (valid votes cast) , N is the number of seats to be filled, and X is equal to zero or one, depending on the specific jurisdiction rules. In multi-winner contests the threshold is calculated in the first round only and applies to each round.

Minneapolis and Cambridge rules use the first, more traditional approach while some more recent approaches use the second. Minneapolis rules describe X as being the result of adding one and ignoring any fractional value, i.e., rounding down to the nearest whole number. Recent California and federal legislative proposals round up to the fourth decimal place. The diagram expresses the threshold criterion in terms of the second approach with the understanding that a user can substitute the greater-than-or-equal-to criterion of the first approach as appropriate.

15. Any candidate votes >= T?

Take the Yes path if the vote total for any (continuing) candidate satisfies the threshold criterion, i.e., is greater than the threshold. In a multi-winner election, it is possible for more than one continuing candidate to exceed the threshold in a round. Take the No path if there are not any continuing candidates with a vote total that satisfies the threshold criterion. If multiple candidates meet the threshold in a single round, jurisdiction rules determine how the surpluses are processed.

16. One seat w/rules requiring end w/2 candidates?

For a single-winner contest, the California IRV guidelines and some local jurisdictions require that the recursive process of candidate elimination and promotion of the subsequent highest-ranking continuing candidate continue until only two candidates remain even if a candidate reaches the threshold to be elected. Take the Yes path if these rules apply. Take the No path if the contest is either a multi-winner contest or if using conventional rules for a single-winner contest.

17. 2 candidates left?

This block applies if the rules for single-winner contests require the process to continue until 2 candidates are left. Take the Yes path if there are only two candidates left, and the leading vote getter is indicated. Take the No path if there are more than two candidates left and cause the process to continue.

18. Elect candidate(s)

Elect one or more of the continuing candidates with a vote total that satisfies the threshold criterion. Depending on jurisdiction-specific rules, if there is more than one such candidate, all of them might be elected, or only one might be elected in this round, typically the candidate with the most votes. Jurisdiction-specific rules for resolving a tie for having the most votes 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.

19. All seats filled?

Take the Yes path if this is a single-winner contest or if all required candidates in a multi-winner contest have been elected, indicating the process is complete. (Some rules require, in a multi-winner contest, that any surplus be redistributed before tabulation ends so that all ballot options meeting the threshold end with the same vote totals equal to the threshold). Take the No path if it is a multi-winner contest and seats remain unfilled.

20. Multi-pass IRV?

Take the Yes path if this is a multi-pass IRV contest and there are remaining seats to be filled. Take the no path if this is a multi-winner contest using Single Transferable Vote. In either case, any candidate(s) elected are no longer eligible to receive votes. Any voter selections for already elected candidates are ignored in further ballot tabulation processing.

21. Calculate surplus votes transfer formula

In a round of a multi-winner contest when one or more candidates are elected, but seats remain unfilled, stage CVRs containing excess votes for the elected candidate for further processing along with the other CVRs for continuing candidates. This step determines the formula for how to stage these CVRs.

There are two methods currently used for handling surplus votes (votes for an elected candidate that are over the threshold). In most modern cases, all CVRs for the elected candidate are assigned a transfer vote value that is a fraction of a whole vote that corresponds to the ballot's prorated share of the elected candidate's surplus. The fraction is equal to the ratio of the elected candidate's surplus votes for the round divided by the elected candidate's total votes for that round. Jurisdiction-specific rules may specify the precision and any rounding (typically rounding down) that are associated with this arithmetic operation.

In contrast, Cambridge processes ballots from precincts in a randomly chosen order and selects every Nth ballot where N is the total votes for the elected candidate divided by the excess votes (rounded to the nearest whole number) and transfers the full vote of the selected CVRs to continuing candidates.

22. Stage CVRs with surplus votes for candidates meeting threshold

Stage all CRVs for continuing candidates, including CVRs containing surplus votes for any candidate meeting the threshold in this round according to the formula developed in Block 21.

In Minneapolis, all CVRs are transferred with a vote value fraction times the CVR's previous transferred vote value. Note that the previous transferred value might be a fraction if it was a surplus from a candidate meeting the threshold in a previous round. Jurisdiction specific rules may specify the precision and any rounding (typically rounding down) in this multiplication.

In Cambridge, the CVRs are selected and staged at full vote value, according to the formula determined in Block 21.

Recently adopted tabulation rules redistribute any surplus as a fractional vote for all CVRs that counted for the elected candidate. Cambridge rules, adopted prior to the availability of computer tabulation, distribute surplus as a whole vote per CVR but only for a subset of CVRs.

23. # continuing candidates = # unfilled seats

Take the No path if there are more continuing candidates than the number of unfilled seats. Take the Yes path if the number of continuing candidates is equal to the current number of unfilled seats.

24. Defeat candidates with fewest votes and stage CVRs

A common approach is to eliminate (aka defeat) the candidate with the fewest votes and then transfer that candidate's votes to other candidates. There can be jurisdiction-specific rules for how to resolve a tie for having the fewest votes. Some jurisdictions use a lottery while others look at previous round results and eliminate the candidate with the lowest votes in the most 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-winner contests to allow or require a group of candidates with the fewest votes to be eliminated in a single round (aka batch elimination) if their combined vote totals are less than the candidate with the next higher vote total. Use of this option does not change who is elected, i.e., its use is outcome invariant. For multi-winner elections, the criteria for doing batch eliminations or eliminations while there are candidates with votes over the threshold are more complicated if outcome invariance is to be maintained.

Some rules have exceptional elimination rules. For multi-winner contests, Minneapolis requires certain candidate eliminations, including batch eliminations, even though there might be continuing candidates that satisfy the threshold criterion for being elected. Cambridge requires elimination after any surplus is transferred from candidates elected in the first round and of every candidate with fewer than 50 votes. The 50-vote minimum is derived from 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.

25. Elect all continuing candidates

Elect all continuing candidates to fill the remaining unfilled seats. This allows the election of a contest option not satisfying the threshold criterion.

26. Done

Indicates that candidates have been elected to all positions up for election and the tabulation process is complete.

3.2.18 Process Top-2 IRV Contingent Vote

3.2.18.1 Introduction

Top-2 IRV, also known as Contingent Vote, is a form of instant runoff voting where all but the top two 1st round ballot options are eliminated after the 1st round of tabulation. The traditional IRV tabulation process is a multi-round process that is either always used or is used if no candidate receives sufficient votes in tabulated first choices to meet the requirements to be elected. In each round, the candidate with the least votes is eliminated and, using the cast vote records (CVRs), a vote from the eliminated candidate is counted for the ballot's

next highest-ranked continuing candidate, unless there is an intervening selection that disqualifies the ballot. This process proceeds until the requirements are met to have the candidate(s) elected.

Top-Two IRV simulates a separate runoff election by making only the top 2 candidates after the first round of counting eligible to win. CVRs are processed only for those ballots where the first choice was not for the eligible candidates but which have a valid subsequent vote for an eligible candidate. The ballot is counted for one of the two the eligible candidates and the candidate with the most votes after this additional round wins.

As mentioned in 3.2.17, the traditional method of IRV tabulation is most efficient when all CVRs are centralized to a single location. . This centralization makes it possible to run the round-by-round count all at once using counting software, instead of in a distributed process relying on vote totals being called in from counting centers between each round. In contrast, Top-2 IRV will have a maximum of two rounds: the first round count that determines the top two candidates and the distribution of votes for anyone outside the top two to whichever of the top two candidates is ranked highest on that ballot.

Note that this method can be used for multi-winner contests by electing candidates one at a time and, after a candidate is determined to be elected, removing that candidate from the eligible list before rerunning the election to fill the next available seat.

The following flow chart and ensuing description is an attempt to chart a Contingent Vote process that emulates a separate runoff election as it applies to a single-winner contest. See the Process Ranked Choice Voting Contest (IRV or STV) section for handling the “traditional” methods of RCV processing.

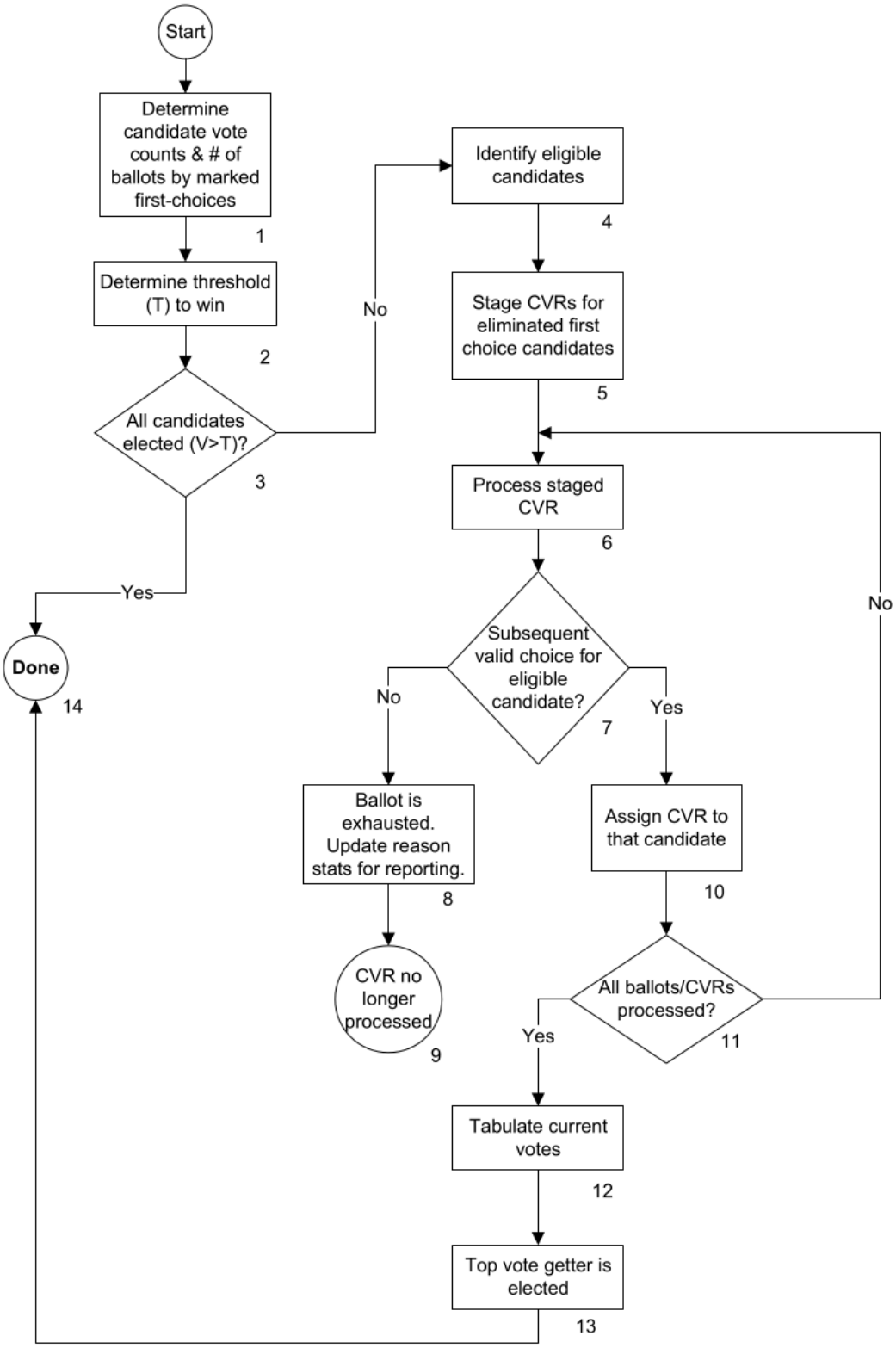


Figure 13 - Process flow for Top-2 IRV Contingent Vote.

Description

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

The number of 1st choice candidate votes used for an initial determination of whether a candidate is determined to be elected is a tabulated count of valid marked 1st choice selections. The tabulated marked 1st choice count of selections is normally the initial tabulator count and does not require processing of the Cast Vote Record. Note that the Top-2 IRV tabulation process is not needed if a candidate has the required number of votes to be elected per step 3.

2. Determine threshold (T) to win

The threshold is the number of votes needed to be elected in the first count in a Top-2 IRV election. For a single winner contest, the threshold is typically at least 50% of the ballots cast. In some jurisdictions criteria for threshold requires 50% of the votes + 1 vote to be elected. However, there can be exceptions depending on jurisdiction-specific rules. The threshold might be based on total votes cast for candidates or might be based on the total number of ballots. The generic formula for a candidate to be elected is:

$$T > B / (2)$$

T is the threshold, B is the number of relevant ballots and N is the number of candidates to be elected. Thus in a contest where only 1 is to be elected the threshold is B/2 and the required number of votes to be elected is one more than that number (50% +1).

3. All candidates elected (V >T)?

Is there a candidate with a number of first-choice votes greater than the threshold? Take the Yes path if this is true. Take the No path if the position has not been filled due to an insufficient candidate number of marked first-choice votes. If a candidate has enough marked first-choice votes to exceed the threshold, the RCV process is not required.

4. Identify eligible candidates

The eligible candidates are the two candidates with the most votes after first-choice tabulation.

5. Stage CVRs for ballots with selected first choice not eligible, i.e. ballots with selected first choice not one of the top two.

CVRs may be processed in groups such as by precinct, by poll or by other relevant groupings. Only those CVRs that do not have a marked first-choice selection for an eligible candidate are processed.

6. Process staged CVR

Determine the next subsequent choice for an eligible candidate, if one exists, for each staged CVR. Note that there are jurisdiction-specific rules when encountering the following conditions on a choice selection while executing the RCV algorithm. The option may also depend on whether the condition is encountered in the 1st choice or subsequent choice selections. These are listed below with the possible options of handling.

- Unvoted choice

    - Ballot considered exhausted
    - Ballot considered exhausted if there are two unvoted choices in succession
    - Skipped and subsequent eligible choice processed, if any
- Overvoted choice
    - Ballot considered exhausted
    - Not considered overvoted if doesn't contain more than one continuing candidate
    - Skipped and subsequent eligible choice processed, if any
- Repeated ranking of a previously ranked candidate
    - Ballot considered exhausted
    - Skipped and subsequent eligible choice processed, if any

7. Subsequent valid choice for eligible candidate?

Take the No path for a processed CVR when there is no valid choice for an eligible candidate. ie., when the ballot is exhausted. Take the Yes path if there is a valid subsequent choice for an eligible candidate.

8. Ballot is exhausted. Update reason stats for reporting

This CVR will not be included in any further processing as there are no more eligible marked candidates selections. This may be due to the CVR not containing further ballot ranking selections or that all subsequent ranking selections are invalid. Record the reason for the ballot being considered exhausted for purposes of reporting round results.

9. CVR no longer processed

Indicates that the ballot is no longer part of any processing.

10. Assign CVR to that candidate

Assign the next highest-ranked choice on the CVR for an eligible candidate to that candidate. This contributes a full vote to the final accumulation.

11. All CVRs processed?

Take the No path to process the next staged CVR if all CVRs have not yet been processed. Take the Yes path if all staged CVRs have been processed.

12. Tabulate current votes

Tabulate a count of all candidate votes from the processed CVRs.

13. Top vote-getter is elected

The candidate with the most votes is elected. Jurisdiction rules apply In the event of a tie.

14. Done

Indicates that a candidate has been elected and the process is complete.

3.2.19 Process Presidential Preference Contest

3.2.19.1 Introduction

The Presidential Preference contest appears on the closed Presidential Preference Primary ballots of two states, Alabama (both Democratic and Republican ballots) and Rhode Island (Democratic ballot only). It allows the voter to select their Presidential choice and controls the Delegate to the party's National Nominating

Convention that the voter may vote for based on their presidential choice. This section describes the low-level processing associated with the contest. See Presidential Delegate Voting Method for a description of this special Presidential Preference Primary.

The flow chart below indicates the processing flow for the contest and each block in the flow is described.

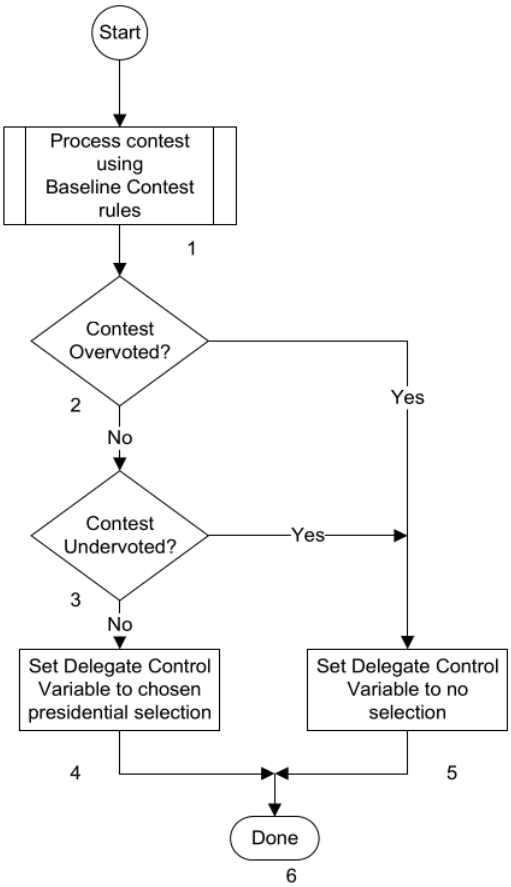


Figure 14 - Presidential preference contest process flow.

Description

1. Process the Contest using Baseline Contest rules.

Although the result of processing this contest has special uses, the processing itself treats it like any other Baseline contest. Selections are always limited to one choice, so there are 3 possible processing results: a presidential choice selected, an undervote (no selection) or an overvote (too many selections).

2. Contest Overvoted?

Was the result of processing the contest an overvote? Take the Yes path if it was. Take the No path if it was not.

3. Contest Undervoted?
Was the result of processing the contest an undervote? Take the Yes path if it was. Take the No path if it was not.

4. Set Delegate Control Variable to chosen Presidential selection
Reach this block only if there was a valid Presidential choice made by the voter. Set the Delegate Control Variable to the chosen Presidential selection. For flow chart purposes, the Delegate Control Variable is the term being used to remember the valid selection made in the Presidential Preference contest. This selection is needed when processing selections from Presidential Delegate contests on the ballot.

5. Set Delegate Control Variable to No and Party Variable to no selection
Reach this block if the contest was either not voted or overvoted. In either case there is no valid Presidential selection. Set the Delegate Control Variable to no selection.

6. Done
Processing of the Presidential Preference contest completed.

3.2.20 Process Presidential Delegate Contest

3.2.20.1 Introduction

The Presidential Delegate contest appears on the closed Presidential Preference Primary ballots of two states, Alabama (both Democratic and Republican ballots) and Rhode Island (Democratic ballot only). It allows the voter to select candidates from various contest groupings to be delegates to the party National Nominating Convention. The allowed selections, however, are limited to those that are representing the presidential candidate chosen by the voter. See Presidential Delegate Voting Method for a description of this special Presidential Preference Primary.

The flow chart below indicates the processing flow for the contest, and each block in the flow is described.

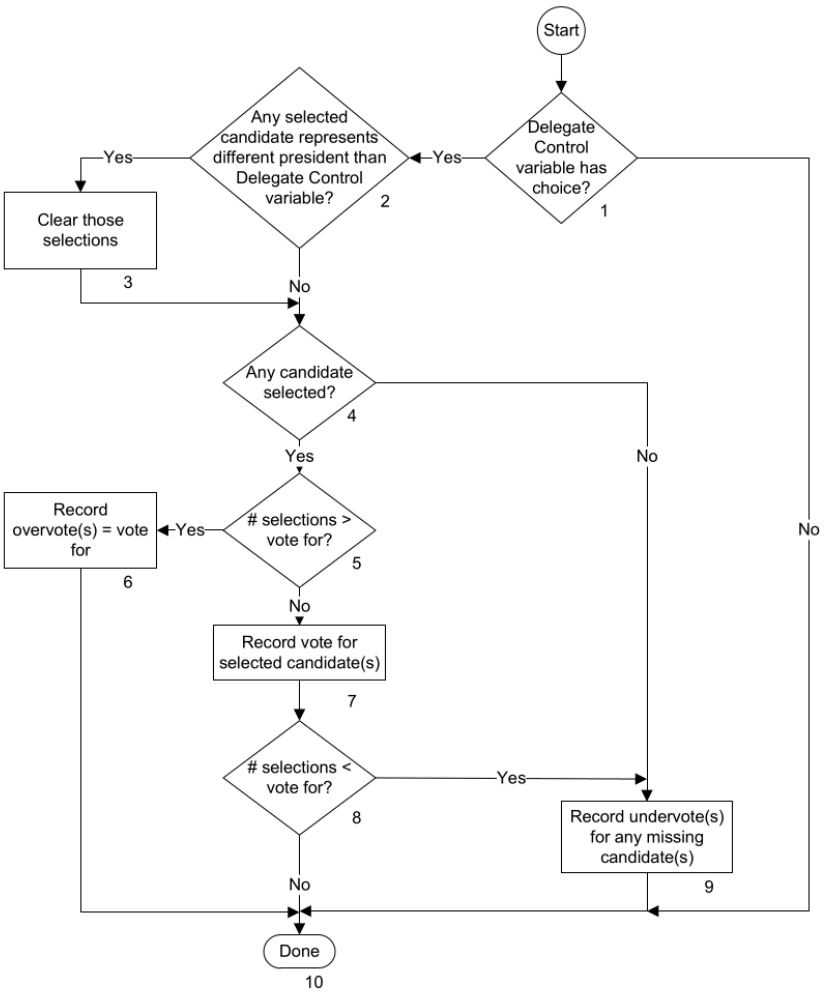


Figure 15 - Presidential delegate contest process flow.

Description

1. Delegate Control variable has a choice selected?

Does the Delegate Control variable contain a Presidential choice? For flow chart purposes, the Delegate Control variable is the term being used to remember the result of processing the Presidential Preference contest. If the variable does not contain a Presidential Preference contest choice, take the No path and do not tabulate the contest. Take the Yes path if there was a choice.

2. Any selected candidate represents a different president than Delegate Control variable?

Do any of the selected candidates represent a different presidential candidate than the presidential candidate selected by the voter as identified in the Delegate Control variable? (For flow chart purposes, the Delegate Control Variable is used to remember the selection in the Presidential Preference contest.) Take the Yes path if there were any. Take the No path if there are none.

3. Clear those selections

Clear the selection of any chosen delegate candidate that represents a different presidential candidate than the one chosen by the voter, and continue processing.

4. Any candidate selected?

If there is no remaining selection after clearing selections, or there was no selection in the contest to begin with, take the No path. Take the Yes path if the voter selected at least one candidate.

5. # Selections > vote for?

Are there more selections of candidates than the allowed number of selections? Take the Yes path if there are. Take the No path if the number of valid selections is less than or equal to the number allowed.

6. Record overvote(s) = vote for

Reach this block if there are more valid selections than the number allowed. Record a number of overvotes equal to the votes allowed in the contest. This completes contest processing.

7. Record vote for selected candidate(s)

All selected candidates that are valid based on presidential candidate selection receive a vote.

8. # selections < vote for?

Is the number of selected candidates that are valid based on the presidential selection less than the number of selections allowed? Take the Yes path if there are unused selections so that undervotes can be tabulated. Take the No path if all choices were used as processing is completed.

9. Record undervote(s) for any missing candidate(s)

If no candidate is selected in the contest or if there are fewer valid candidates selected than allowed, record an undervote for each unallocated vote. Processing is then completed.

10. Done

Processing of the Presidential Delegate contest completed.

3.2.21 Process Massachusetts Group Vote Contest

3.2.21.1 Introduction

Massachusetts Group Voting is a voting method that only appears on a Massachusetts Presidential Primary ballot and is used to elect members of a Town, Village or Ward Committee. These committees have a large number of seats (e.g., 35) with all candidates selected from a single contest. Candidates may run as part of a specific group or run unaffiliated with a group. Due to the large number of candidates that can be selected, the contest format using this method has a group select as the first candidate listed in the group with individual group members listed after the group select. Following the group, there may be a group select for a next group (if there is one) followed by its members. Following all of the groups there would be a list of unaffiliated candidates and finally a list of write-in positions usually equal to the number of

seats. The group select is provided as a convenience, to help the voter avoid individually selecting all candidates in a group. A group selection acts as if all candidates in the group are individually selected. Note that Group Select ballot positions are uniquely identified and each candidate in the group is linked to its group selection position.

Tabulation in this contest determines all candidates that are selected whether by group select or individually. If a group is selected, candidate selections in the same group have no effect. However, if a candidate outside the group is selected, whether they are an unaffiliated candidate, a write-in, or in a different group, and the selected group is full (the number of candidates in the group is equal to the number of seats), an overvote is created. Candidates outside the group may be selected when a group has fewer candidates than the number of seats. Then, the number of selections is limited to the number of missing candidates. If they exist, voters may select multiple groups without causing an overvote as long as the total number of candidates selected combined do not exceed the number of selections allowed.

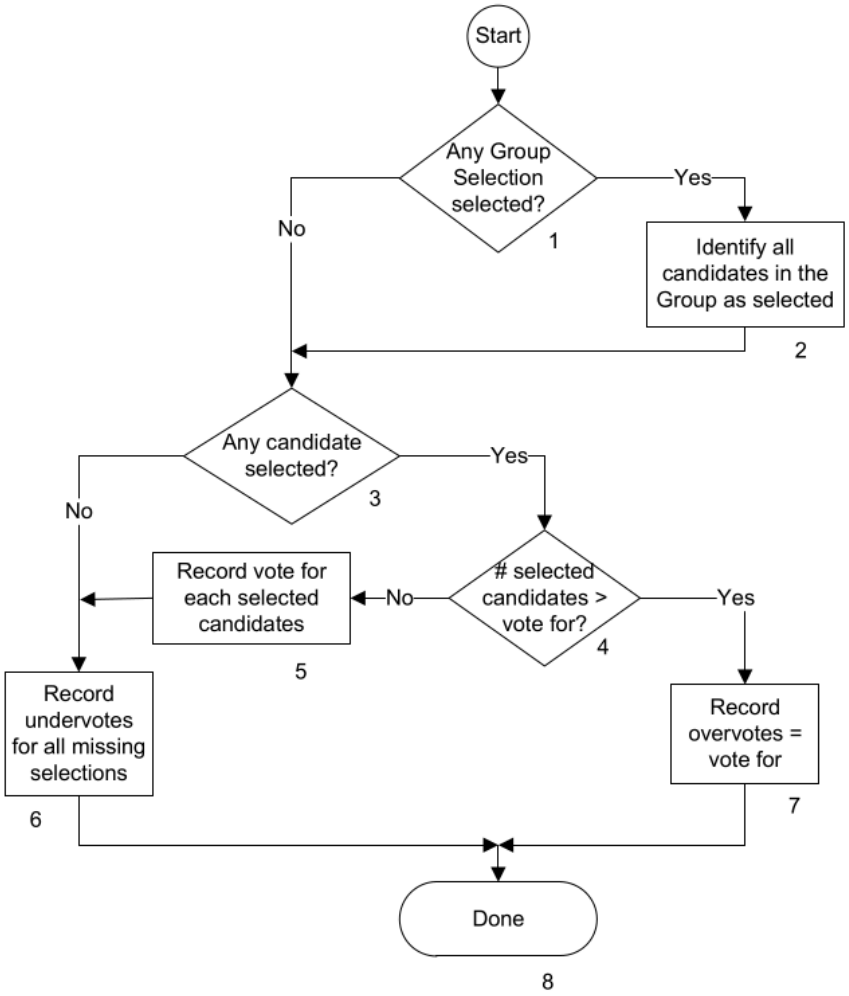


Figure 16 - Massachusetts Group Vote contest process flow

Description

1. Any Group Selection selected?

A Group Selection position is a unique position listed before all the candidates in a defined group that automatically selects all candidates in the group (similar to a straight-party selection). Take the Yes path if there is at least one Group Selection position chosen. Take the No path if there is no Group Selection chosen.

2. Identify all candidates in the Group as selected

Selecting the Group Select is treated the same as if an individual selection is made for every candidate linked to the group. A selection is marked for each candidate in the group just as if a voter individually selected every candidate in the group.

3. Any candidate selected?

Have any candidates been selected, whether individually or as a result of a group selection? Take the No path if there are neither individually selected candidates nor a group selected. This indicates that the contest is unvoted. Take the Yes path if there are one or more individual candidates selected or if a group is selected.

4. The number of selected candidates > vote for?

Whether a candidate is individually selected, selected indirectly by a group selection or is selected by both still only represents one selection. If the total number of candidates selected is greater than the votes allowed, it is an overvote. Take the Yes path. If not, take the No path.

5. Record vote for each selected candidate

Since this box is reached if there was no overvote, each selected candidate, whether individually selected or selected indirectly as part of a group, receives a vote.

6. Record undervotes for all missing selections

If there are no candidates selected or the number of selected candidates is less than the number of allowed selections (vote for number), record a number of undervotes for each unused vote selection.

7. Record overvotes = vote for

Reach this box if the contest has more selections than allowed. Record a number of overvotes equal to the allowed number of selections.

8. Done

Processing of the Massachusetts Group Vote contest completed.

3.2.22 Test Conditions & Expected Outcomes

3.2.22.1 Introduction

Every contest must use one or more methods to tabulate the ballot's contest selections. The various available methods have been flow charted and the tabulation processing described in detail in the above subsections. Other than the Baseline contest, all other tabulation methods are state or jurisdiction-specific and are related to the voting methods identified in the state or jurisdiction legislative rules.

Each spreadsheet in the following set is associated with a different tabulation method. Each identifies a sample contest or set of sample contests (as required by the tabulation method) and a set of voting pattern examples of the defined contests to illustrate tabulation of that voting method for varying contest selections. A sufficient number of examples are provided to cover all conditions that can be encountered using that method. Each example has a column for selections and a column for the tabulated outcome. Rows include candidates in the contest as well as overvotes and undervotes to enable accounting of all eligible votes. Each sheet includes a set of descriptions containing the "ballot" description used and a description of each illustrated example. These spreadsheets function as companions to the flow charts and a description of the associated tabulation method.

3.2.22.2 Baseline Contest

Ballot Description	The ballot used for illustration has 2 baseline contests, one with 4 candidates and 1 selection allowed and the other with 6 candidates and 2 selections allowed.
Example 1	Shows full set of selections in both vote for 1 and vote for multiple (2) contests.
Example 2	Shows 1 more selections than allowed in both the vote for 1 and vote for 2 contests and how overvote is tabulated. (overvotes = vote for)
Example 3	Shows 2 extra selections beyond what's allowed and and how overvote tabulation is independent of the number of excess selections. (overvotes = vote for)
Example 4	Shows undervote tabulation in both vote for 1 and vote for multiple (2) when no selections are made in the contest. (Undervotes = vote for if no selections)
Example 5	Shows 3 extra selections beyond what's allowed and and how overvote tabulation is independent of the number of excess selections. (overvotes = vote for)
Example 6	Shows undervote tabulation in multiple vote for contest when not fully voted. (undervote = vote for minus # selections; 2 -1 = 1)

Regular Contests	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Contest A (vf 1) (1 of 4)												
Candidate 1	X	1			x				x			
Candidate 2									x			
Candidate 3			x		x				x			
Candidate 4			x		x				x			
overvote				1 (=vf)		1 (=vf)				1 (=vf)		
undervote								1 (vf-#sel)				1 (vf-#sel)
Total		1		1		1		1		1		
rule	any 1	# sel = vf	any 2	# sel > vf	any 3	# sel > vf	none	# sel < vf	all	# sel > vf		
Contest B (VF 2) (2 of 6)												
Candidate 5	x	1			x				x			
Candidate 6			x		x				x		x	1
Candidate 7			x						x			
Candidate 8	x	1			x				x			
Candidate 9			x		x				x			
Candidate 10												
overvote				2 (=vf)		2 (=vf)				2 (=vf)		
undervote								2 (vf-#sel)				1 (vf-sel)
Total		2		2		2		2		2		2
rule	any 2	# sel = vf	any 3	# sel > vf	any 4	# sel > vf	none	# sel < vf	all	# sel > vf	any 1	# sel < vf

Table 1 - Baseline contest sample test sets and expected outcomes

3.2.22.3 Approval Voting

Ballot Description	The ballot used for illustration has 2 approval voting contests, both of the contests have 5 candidates but the first is for a single seat while the second is to fill 2 seats. (Note that tabulation in both is the same (M of M) with the allowed selections being equal to the number of candidates and therefore an overvote is not possible. The difference is that the top vote getters equal to the number of seats are elected.)
Example 1	Voter has specific preferences for candidates for each seat
Example 2	Voter like either if 2 candidates for 1st contest seat and any of 3 candidates for the 2 seats in the second contest.
Example 3	Voter is OK with any of 3 candidates for the 1st seat but only likes 1 candidate to fill one of the 2 seats in 2nd contest.
Example 4	Voter doesn't like candidate 2 in first contest nor candidate 9 in second contest. Finds others OK.
Example 5	Voter doesn't dislike any candidate in either election or doesn't care so votes for everyone
Example 6	Votere dislikes all candidates or doesn't care who wins and selects no one.

Regular Contests	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Contest A (vf 5) Single Seat												
Candidate 1	X	1			x	1	x	1	x	1		
Candidate 2									x	1		
Candidate 3			x	1	x	1	x	1	x	1		
Candidate 4			x	1	x	1	x	1	x	1		
Candidate 5							x	1	x	1		
overvote												
undervote		4		3		2		1				5
Total		5		5		5		5		5		5
rule	1 pref		2 pref	5	3 pref		1 dslkd		Don't care		Don't care	
Contest B (VF 5) 2 seats												
Candidate 6			x	1			x	1	x	1		
Candidate 7			x	1			x	1	x	1		
Candidate 8	x	1					x	1	x	1		
Candidate 9	x	1	x	1	x	1			x	1		
Candidate 10							x	1	x	1		
overvote												
undervote		3		2		4		1				5
Total		5		5		5		5		5		5
rule	2 pref		3 pref		likes 1		1 dslkd		Don't care		Don't care	

* Indicates overvote is not possible as allowed selections is the same as the number of candidates.

Table 2 - Approval Voting contest sample test sets and outcomes

3.2.22.4 Straight Party Contest

Ballot Description	The ballot used for illustration contains a Straight Party contest selection and 2 partisan contests, one contest with one selection allowed and 4 candidates, one being nonpartisan, the other contest with 8 candidates and 3 selections allowed. There are two common variations of tabulatiion, one designated Inclusive and the other exclusive. Both are illustrated in the examples. Exclusive nullifies any straight party selection in a contest if there is any direct selections while inclusive combines the two as long as the combination does not cause an overvote. If the combination causes an overvote, it nullifies the straight party selection. This only makes a difference in multiple vote for contests. The second contest only has a complete slate of candidates for the Democratic party to allow tabulation difference to be illustrated.
Example 1	Straight Party selection with no other selections - No difference inclusive & exclusive
Example 2	Straight Party selection with selection for other candidate when party has a full slate. No difference inclusive & exclusive.
Example 3	Straight Party selection with selection for other candidate when party has an incomplete slate. **Inclusive & exclusive different in vf>1.**
Example 4	Straight Party selection with contest selection in same party. **Inclusive & exclusive different in vf>1.**
Example 5	Straight Party overvoted. All contests use baseline tabulation **except in Indiana** where all partisan contests are overvoted.
Example 6	No Straight Party. Contests use baseline tabulation

Contest	Example 1			Example 2			Example 3			Example 4			Example 5				Example 6		
	Choices	Incl Tab	Excl Tab	Choices	Incl Tab	Excl Tab	Choices	Incl Tab	Excl Tab	Choices	Incl Tab	Excl Tab	Choices	Incl Tab	Excl Tab	IN Tab	Choices	Incl Tab	Excl Tab
Straight Party Selection																			
Republican	x	1	1	x	1	1													
Democratic										x	1	1	x						
Libertarian							x	1	1				x						
overvote														1	1	1			
undervote																		1	1
Total		1	1		1	1		1	1			1		1	1	1		1	1
rule	any 1						inc slate						ovrvt				No str pty		
Contest A (vf 1) (1 of 4)																			
Candidate 1 (Rep)		1 (pv=R)	1 (pv=R)														x	1	1
Candidate 2 (Dem)							x	1	1	x	1	1	x	1	1				
Candidate 3 (Lib)				x	1	1													
Candidate 4 (Non)																			
overvote																1 (=vf)			
undervote																			
Total		1	1		1	1		1	1		1	1		1	1			1	1
rule	sp sel			dir sel									reg tab				reg tab		
Contest B (VF 3) (3 of 8)																			
Candidate 5 (Rep)		1 (pv=R)	1 (pv=R)										x	1	1				
Candidate 6 (Rep)		1 (pv=R)	1 (pv=R)														x	1	1
Candidate 7 (Rep)		1 (pv=R)	1 (pv=R)				x	1	1				x	1	1				
Candidate 8 (Dem)											1 (pv=D)						x	1	1
Candidate 9 (Dem)				x	1	1				x	1	1							
Candidate 10 (Dem)											1 (pv=D)								
Candidate 11 (Lib)								1 (pv=L)											
Candidate 12 (Non)							x	1	1								x	1	1
overvote																3(=vf)			
undervote					2	2			1			2		1	1				
Total		3	3		3	3		3	3		3	3		3	3	3		3	3
rule	none		# sel = vf				pty cross			same pty	incl	excl	reg tab			IN SP ov	reg tab		

Table 3 -. Straight-party contest sample test sets and expected outcomes.

3.2.22.5 Open Primary Contest

Ballot Description	An Open Primary ballot has contests for all parties and can also have nonpartisan contests. Although all contests are on the ballot, the voter is only allowed to make selections in one party's contest to avoid nullifying selections in all partisan contests. Open Primary ballots in some states also have a Party Preference contest, which if voted, prevents a voter from "spoiling" their ballot since only those selection in contests of a matching party to their party preference selection will be tabulated. All others will be ignored. If a Party Preference contest is on the ballot but either not voted or overvoted, there is no difference in the tabulation of the remaining contests as the same ballot that does not have the party preference contest but is otherwise voted the same. Examples 1 and 4 explicitly represent a ballot without the Party Preference Contest.
Example 1	Standard Open Primary ballot voted as a Republican. (No Party Preference contest)
Example 2	Open Primary ballot with unvoted Party Preference contests voted as a Democratic.
Example 3	Open Primary ballot with Party Preference contest voted as Nonpartisan.
Example 4	Same as Example 1 with extra dem vote (crossover) No Party Preference contest.
Example 5	Same as Example 4 with unvoted Party Preference Contest
Example 6	Same as Example 4 with Rep voted in Party Preference Contest
Example 7	Same as Example 4 with overvoted Party Preference Contest

Ballot Statistic	Republican		Democratic		Nonpartisan		NonPartisan (xover)		NonPartisan (xover)		Republican		NonPartisan (xover)	
Contest	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Party Preference Selection	N/A						N/A							
Republican											x	1	x	
Democratic													x	
Libertarian														
overvote														1
undervote				1		1				1				
Total				1						1		1		1
rule	N/A		std		std		N/A		std		std		ov	
Contest A - REP (vf 1) (1 of 3)														
Candidate 1	x	1					x		x		x	1	x	
Candidate 2														
Candidate 3														
overvote														
undervote														
Total		1										1		
rule	std						xover		xover		PP		xover	
Contest B - REP (VF 3) (3 of 8)														
Candidate 4	x	1					x		x		x	1	x	
Candidate 5														
Candidate 6	x	1					x		x		x	1	x	
Candidate 7														
Candidate 8														
Candidate 9	x	1					x		x		x	1	x	
Candidate 10														
Candidate 11														
overvote														
undervote														
Total		3										3		
rule	std						xover		xover		PP		xover	
Contest C - DEM (vf 1) (1 of 3)														
Candidate 12														
Candidate 13			x	1			x		x		x		x	
Candidate 14														
overvote														
undervote														
Total				1										
rule			std				xover				PP			
Contest D - DEM (VF 3) (3 of 8)														
Candidate 15														
Candidate 16														
Candidate 17														
Candidate 8			x	1										
Candidate 9														
Candidate 10			x	1										
Candidate 11														
Candidate 12														
overvote														
undervote				1										
Total				3										
rule			std											
Contest F - NON Referendum														
For	x	1	x	1	x	1	x	1	x	1	x	1	x	1
Against														
overvote														
undervote														
Total		1		1		1		1		1		1		1
rule	std		std		std		std		std		std		std	

Table 4 . Open primary contest expected outcomes and sample test sets.

3.2.22.6 Cumulative Contest

Ballot Description	A contest with 8 candidates and 5 votes allowed is used to illustrate both cumulative voting as well as proportional voting so that the two can be compared.
Example 1	Ballot selection designated all allowed votes to one candidate.
Example 2	Ballot selection split all allowed votes between 2 candidates, 3 to one and 2 to the other. (always whole number of votes)
Example 3	Ballot selection split all allowed votes between 3 candidates, 2 each with 2 votes and a third with 1 vote. (always whole number of votes)
Example 4	Ballot selection split all allowed votes between 4 candidates, 3 each with 1 vote and a third with 2 votes. (always whole number of votes)
Example 5	Ballot selection split all allowed votes between 5 candidates, each with 1 vote. (voted like a regular contest)
Example 6	The total vote allocation to the selected candidates exceeds the votes allowed. (4 votes to 1 candidate and 2 to a second candidate exceeds the 5 allowed votes)
Example 7	Selection of no candidate fully undervotes the contest.
Example 8	Ballot selection split all allowed votes between 2 candidates, 4 to one and 1 to the other, a variation of example 2. (illustrates the ability to determine the desired number of votes to allocate.
Example 9	Ballot selection does not allocate all eligible votes. Any unallocated votes become undervotes. (3 votes allocated to 2 candidates leaving 2 unallocated votes)

Contest	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7		Example 8		Example 9	
Cumulative	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Contest (vote for 5)																		
Candidate 1	5	5			2	2			1	1								
Candidate 2					1	1	1	1	1	1	4							
Candidate 3			2	2					1	1					1	1	1	1
Candidate 4					2	2	1	1										
Candidate 5											2							
Candidate 6							1	1										
Candidate 7							2	2	1	1								
Candidate 8			3	3					1	1					4	4	2	2
overvote												5 (=vf)						
undervote														5				2
Total		5		5		5		5		5		5		5		5		5
rule	All to 1	as voted	Split 2can	as voted	split 3can	as voted	splt 4 can	as voted	splt 5 can	as voted	ov	# sel > vf	full uv		large split	as voted	partial uv	

Table 5. Cumulative voting expected outcomes and sample test sets.

3.2.22.7 Proportional Voting Contest

Ballot Description	A contest with 8 candidates and 5 votes allowed is used to illustrate both cumulative voting as well as proportional voting so that the two can be compared.
Example 1	Single ballot selection credits all allowed votes to selected candidate.
Example 2	Selection of 2 candiates credits half of allowed votes to each candidate. (may be fractional)
Example 3	Selection of 3 candidates credits 1/3 of allowed votes to each candidate. (may be fractional)
Example 4	Selection of 4 candidates credits 1/4 of allowed votes to each candidate. (may be fractional)
Example 5	Selection of a number of candidates equal+B5 to allowed votes credits 1 vote to each candidate.
Example 6	Selection of more candidates than allowed creates an overvote as with any other contest.
Example 7	Selection of no candidate fully undervotes the contest and is the only way an undervote can be created.

Contest	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7	
Proportional	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Contest (vote for 5)														
Candidate 1	X	5			x	1 & 2/3			x	1				
Candidate 2							x	1 & 1/4	x	1	x			
Candidate 3			x	2 & 1/2	x	1 & 2/3			x	1	x			
Candidate 4							x	1 & 1/4			x			
Candidate 5														
Candidate 6							x	1 & 1/4			x			
Candidate 7							x	1 & 1/4	x	1	x			
Candidate 8			x	2 & 1/2	x	1 & 2/3			x	1	x			
overvote												5 (=vf)		
undervote														5
Total		5		5		5		5		5		5		5
rule	any 1	All votes	any 2	ea = 1/2 vf	any 3	ea = 1/3 vf	any 4	ea = 1/4 vf	any 5	each = 1/5 vf	ov		uv	

Table 6 - Proportional Voting expected outcomes and sample test sets.

3.2.22.8 Controlling and Controlled Contest Pair

Ballot Description	The contest pair used for illustration contains a controlling contest with a For and Against selection and a controlled contest with 4 candidates and 2 to be elected. This is a generic ability and is not necessarily for recall of a elected official and a vote for a replacement where the controlled contest would be a vote for 1. Another example of use would be the establishment of a district and the selection of council members/administrators of the district. The controlled contest being a vote for 2 allows the illustration to be more generalized as it shows the consequence of incomplete selection. The rule requires a selection in the controlling contest to allow tabulation of selections in the controlled contest.
Example 1	No selection in Controlling contest with votes in the controlled contest. Controlling contest undervoted. Controlled contest undervoted (selections invalid).
Example 2	Selection in Controlling contest with no selections in controlled contest. Controlling contest tabulated as voted. Controlled contest undervoted.
Example 3	Controlling contest overvoted with votes in controlled contest. Controlling contest tabulated as overvoted and controlled contest undervoted (selections invalid).
Example 4	Valid selection in controlling contest with controlled contest overvoted. Controlling contest tabulated as voted & controlled contest overvoted.
Example 5	Both contests overvoted. Controlling contest overvoted. Controlled contest undervoted as selections are invalid (not tabulated).
Example 6	Valid selection in controlling contest but controlled contest not fully voted. Controlling contest tabulated as voted. Controlled contest tabulated as voted with missing vote tabulated as undervote.
Example 7	Fully voted valid selection in both contests.

Contests	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Recall Control														
For Measure			x	1	x		x	1	x		x	1	x	1
Against Measure					x				x					
overvote						1 (=vf)				1 (=vf)				
undervote		1 (=vf)												
Total		1		1		1		1		1		1		1
rule	unvoted	# sel = vf												
Linked Contest (VF 2)														
Candidate 1	x				x		x		x					
Candidate 2							x		x		x	1	x	1
Candidate 3							x		x					
Candidate 4	x				x		x		x				x	1
overvote								2 (=vf)						
undervote		2 (=vf)		2 (=vf)		2 (=vf)				2 (=vf)		1		
Total				2		2		2		2		2		2
rule	not elig	No tab	uv	normal	not elig	No tab	all (ov)	Normal	all	no tab	any 1	# sel < vf	any 1	# sel < vf

Table 7 - Controlling and Controlled Contest expected outcomes and sample test sets.

3.2.22.9 Ranked Choice Voting Contest

Incorporated above in 3.1.3.16 and 3.1.3.17 are the currently applied variants in jurisdiction specific processing rules for irregular voter ballot marking, i.e., not in strict compliance with the voting instructions. This section addresses each of these variants and provides examples of the expected outcomes of CVR processing using each of the variants.

While the examples provided below depict single-winner contest processing, the counting rule variants operate identically when applied to multi-winner contests.

3.2.22.9.1 Unvoted choice (no candidate ranked at a ranking level)

3.2.22.9.1.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet0_Skipped1stChoice

Steps 1 and 2 discuss the options of counting “marked” first choices vs. “effective” first choices. The following table provides

an example of the differences in outcomes of these two options where there is an unvoted first choice.

Round 0 shows the expected outcome of counting “marked” first choices as distinct from”effective” first choices as shown in Round 1.

Test Set 0	12 ballots listed. Counting Rules Applied: 1) Omitted 1st choice (ballot 6), Advance to next highest ranked choice; 2) Overvoted 2nd choice (ballot 9), Exhaust on overvote.
Description round 0	Shows what a tabulator tape would report for totals of the 12 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates.
Description round 1	This would match round 0 if all ballots had valid 1st choice votes. Since ballot 6 had no 1st choice selection, the 2nd choice selection (candidate B) is used in the candidate count. A new count is determined. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new count is determined after candidate D is eliminated on ballot 10 and the 2nd choice (candidate B) is used instead. There still is no winner and candidate C will now be eliminated having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 3, 9 and 11. Candidate B is used on ballots 3 and 11 being the 2nd choice. Ballot 9 is exhausted and becomes inactive as the 2nd choice is overvoted. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated selected 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6		B	C	A		B	C	A		B	C	A		B	C	A
7	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
8	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
9	C	A/D	B		C	A/D	B		C	A/D	B		C	A/D	B	
10	D	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C
11	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
12	A	B	C	C	A	B	C	C	A	B	C	C	A	B	C	C
Cand	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	4	36%			4	33%			4	33%			4	36%		
B	3	27%			4	33%	1		5	42%	1		7	64%	2	
C	3	27%			3	25%			3	25%			0	0%	-3	
D	1	9%			1	8%			0	0%	-1		0	0%		
Votes Counted	11				12				12				11			
Overvote	0				0				0				0			
Undervote	1				0		-1		0				0			
Inactive Ballot	0				0				0				1		1	
Total Ballots	12				12				12				12			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 8 - Ranked choice voting expected outcomes and sample test sets for a ballot with a skipped ranking in the first rank.

Step 6 in the RCV process flow description enumerates the following options for processing skipped choices: Exhaust ballot, Advance to next continuing candidate or exhaust ballot if there are two consecutive skipped choices. The following three sample test data sets demonstrate the processing of each of these variants respectively.

3.2.22.9.1.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet3_Skipped Choice_Exhaust Ballot

Parameters: electing 3 seats			
max skipped ranking =1			
Elected if Vote Total >= threshold			
T=(V/(S+1)+n, 4)			7.2501
0<n<1	=	0.0001	
Adjusted Vote Totals	V=T		

Elected

Eliminated

How each ballot counts in each round with totals by round

	Round 0 1st Choice only	Round 1--Elim D		Round 2--Elect C		Round 3--Redist Surplus from C, Elim F						
Candidate		Vote Total		Vote Total		Surplus	Surplus Fraction		Round 3	Change	Vote Total	
A	6	6		6					6	0.0000	6.0000	
B	5	5		5					5	0.0000	5.0000	
C	6	7		8	Elect	0.7499	0.093738	0.0937	8	-0.7496	7.2504	
D	2	2	elim	0					0.0000		0.0000	
E	6	6		6					6	0.6559	6.6559	
F	3	3		3					3	0.0937	3.0937	elim
Valid	28	29		28					28	0.0000	28.0000	
Inactive	2	1		2					2	0	2.0000	
Residual												
Total	30	30		30					30	0.0000	30.0000	

CVRs — Active Ballot — Inactive Ballot

Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	5th Choice	6th Choice
1	D	undervote	C	E	F	A
2	E	D	F	A	B	C
3	C	E	A	D	B	F
4	F	C	B	D	undervote	undervote
5	B	undervote	undervote	undervote	undervote	A
6	F	A	B	D	E	C
7	B	E	C	D	undervote	undervote
8	A	B	C	undervote	E	F
9	undervote	C	F	D	B	A
10	undervote	undervote	undervote	undervote	undervote	E
11	C	E	D	undervote	undervote	undervote
12	F	C	F	D	B	A
13	C	E	undervote	undervote	F	A
14	E	C	B	undervote	F	D
15	A	B	D	C	F	E
16	D	undervote	undervote	E	F	A
17	E	D	F	A	B	C
18	C	E	A	D	B	F
19	A	C	B	D	undervote	undervote
20	B	undervote	undervote	undervote	undervote	A
21	E	A	B	D	F	C
22	B	E	C	D	undervote	undervote
23	A	B	C	undervote	E	F
24	E	C	F	D	B	A
25	A	undervote	undervote	undervote	undervote	E
26	C	E	D	undervote	undervote	undervote
27	E	C	F	D	B	A
28	C	E	undervote	undervote	F	A
29	B	C	E	undervote	F	D
30	A	B	D	C	F	E

Ballot #	Round 0	Round 1	Original Vote Value	Round 2	Original Vote Value				Round 3	Non-Transfer Value	+	Transfer Value
1	D	D	1	C	1				C	0.9063	E	0.0937
2	E	E	1	E	1				E	1		
3	C	C	1	C	1				C	0.9063	E	0.0937
4	F	F	1	F	1				F	1		
5	B	B	1	B	1				B	1		
6	F	F	1	F	1				F	1		
7	B	B	1	B	1				B	1		
8	A	A	1	A	1				A	1		
9	undervote	C	1	C	1				C	0.9063	F	0.0937
10	undervote	Inactive	1	Inactive	1				Inactive	1		
11	C	C	1	C	1				C	0.9063	E	0.0937
12	F	F	1	F	1				F	1		
13	C	C	1	C	1				C	0.9063	E	0.0937
14	E	E	1	E	1				E	1		
15	A	A	1	A	1				A	1		
16	D	D	1	Inactive	1				Inactive	1		
17	E	E	1	E	1				E	1		
18	C	C	1	C	1				C	0.9063	E	0.0937
19	A	A	1	A	1				A	1		
20	B	B	1	B	1				B	1		
21	E	E	1	E	1				E	1		
22	B	B	1	B	1				B	1		
23	A	A	1	A	1				A	1		
24	E	E	1	E	1				E	1		
25	A	A	1	A	1				A	1		
26	C	C	1	C	1				C	0.9063	E	0.0937
27	E	E	1	E	1				E	1		
28	C	C	1	C	1				C	0.9063	E	0.0937
29	B	B	1	B	1				B	1		
30	A	A	1	A	1				A	1		

Test Set 3	Skipped choice - Exhaust ballot. 15 ballots listed. 1st choice skipped (ballot 1). 2nd choice skipped (ballot 13). 3rd choice skipped (ballot 14).
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvote and 1 undervotes)
Description round 1	This would match round 0 if all ballots had valid 1st choice votes. Since 1st choice on ballot 1 is skipped, ballot 1 is exhausted and becomes inactive. It is omitted from further counting. A new count is determined. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new count is determined after Candidate D is eliminated on ballot 14 and the 2nd choice, candidate C, is selected. There still is no winner and candidate C will now be eliminated, having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 3, 13 and 14. Candidate B is selected on ballot 3, being the 2nd choice. Ballot 13 is exhausted and becomes inactive as the 2nd choice is not voted. Ballot 14 is exhausted and becomes inactive as the 3rd choice is not voted. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1		B	C			B	C			B	C				C	
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	A	B	C		A	B	C		A	B	C		A	B	C	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	C			B	C			B	C			B	C			B
14	D	C		B	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	36%			5	36%			5	36%	0		5	42%	0	
B	6	43%			6	43%			6	43%	0		7	58%	1	
C	2	14%			2	14%			3	21%	1		0	0%	-3	
D	1	7%			1	7%			0	0%	-1		0	0%	0	
Votes Counted	14				14				14				12			
Overvote	0				0				0				0			
Undervote	1				0		-1		0				0			
Inactive Ballot	0				1		1		1				3		2	
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 9 - Ranked choice voting expected outcomes and sample test sets for a ballot exhausting after a single skipped ranking.

3.2.22.9.1.3 Expected_Outcome_RCV Test Sets_Single-winner - TestSet4_Skipped Choice_Advance to next

Test Set 4	Skipped choice - Skip to next continuing candidate. 15 ballots listed. No 3rd or 4th choice on ballot 1. No 4th choice ballot 8. No 2nd or 3rd choice ballot 13. No 3rd choice ballot 14.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvotes and 0 undervotes)
Description round 1	This matches round 0 since all ballots have valid 1st choice votes. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new candidate count is determined after Candidate D is eliminated on ballot 14 and the 2nd choice, candidate C, is selected. There still is no winner and candidate C will now be eliminated, having the least votes. Skipped choices are advanced to next highest ranked continuing candidate.
Description round 3	A new count is determined after candidate C is eliminated on ballots 3, 13 and 14. Candidate B is selected on ballot 3, being the 2nd choice. Candidate B, the 4th choice on ballot 13, is selected since the 2nd and 3rd choices are omitted. The 3rd choice in ballot 14 was not voted so candidate B, the 4th choice, is selected. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	B	C			B	C			B	C			B	C		
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C		D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	A	B	C		A	B	C		A	B	C		A	B	C	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	C			B	C			B	C			B	C			B
14	D	C		B	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	33%			5	33%			5	33%	0		5	33%	0	
B	7	47%			7	47%			7	47%	0		10	67%	3	
C	2	13%			2	13%			3	20%	1		0	0%	-3	
D	1	7%			1	7%			0	0%	-1		0	0%	0	
Votes Counted	15				15				15				15			
Overvote	0				0				0				0			
Undervote	0				0				0				0			
Inactive	0				0				0				0			
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 10 - Ranked choice voting expected outcomes and sample test sets for advancing to the next valid ranking when encountering skipped ranking.

Test Set 5	Skipped choice - Exhaust if 2 successive skipped choices. Otherwise skip to next continuing candidate. 15 ballots listed. No 1st choice on ballot 1. No 4th choice ballot 8. No 2nd or 3rd choice ballot 13. No 3rd choice ballot 14. Single skipped choices are advanced to next highest ranked continuing candidate.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvotes and 1 undervote)
Description round 1	This would match round 0 if all ballots had valid 1st choice votes. Since 1st choice on ballot 1 is skipped, candidate B the 2nd choice is selected and a new count determined. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new candidate count is determined after Candidate D is eliminated on ballot 14 and the 2nd choice, candidate C, is selected. There still is no winner and candidate C will now be eliminated, having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 3, 13 and 14. The 2nd choice on ballot 3 is selected. Ballot 13 is exhausted and becomes inactive since the 2nd and 3rd choices are consecutive skipped choices. The 3rd choice in ballot 14 was not voted so candidate B, the 4th choice, is selected. Candidate B is the winner having more than 50% of the total remaining votes cast

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1		B	C			B	C			B	C			B	C	
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	A	B	C		A	B	C		A	B	C		A	B	C	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	C			B	C			B	C			B	C			B
14	D	C		B	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	36%			5	33%			5	33%	0		5	36%	0	
B	6	43%			7	47%	1		7	47%	0		9	64%	2	
C	2	14%			2	13%			3	20%	1		0	0%	-3	
D	1	7%			1	7%			0	0%	-1		0	0%	0	
Votes Counted	14				15				15				14			
Overvote	0				0				0				0			
Undervote	1				0		-1		0				0			
Inactive	0				0				0				1		1	
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 11 - Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering consecutive skipped ranking.

3.2.22.9.2 Overvoted choice (more than one candidate ranked at a ranking level)

While three variants are mentioned in step 6, in the RCV process flow description, only two, exhausting the ballot or advancing to the next continuing candidate, are actually being employed in US jurisdictions presently. The following sample test data sets demonstrate the processing of each of these variants respectively.

3.2.22.9.2.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet1_Overvote_Exhaust Ballot

Test Set 1	Overvote - Exhaust ballot. 15 ballots listed. 1st choice overvoted (ballot 1). 2nd choice overvoted (ballot 8). Skipped choices are exhausted by rule in this test.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not uniquely assigned to candidates. (1 overvote and 0 undervotes)
Description round 1	Since 1st choice on ballot 1 is overvoted, ballot 1 is exhausted and becomes inactive. It is omitted from further counting. A new count is determined which matches Round 0 since all ballots other than the overvote had valid 1st choice votes. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new count is determined after Candidate D is eliminated. Candidate C is used on ballot 14 being the 2nd choice. There still is no winner and candidate C will now be eliminated having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 3, 8 and 14. Candidate B is used on ballot 3 being the 2nd choice. Ballot 8 is exhausted and becomes inactive as the 2nd choice is overvoted. The 3rd choice in ballot 14 was not voted so by rule ballot 14 is exhausted and becomes inactive. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	A,B		C	D	A,B		C	D	A,B		C	D	A,B		C	D
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	C	B,D	A		C	B,D	A		C	B,D	A		C	B,D	A	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	A			C	A			C	A			C	A			C
14	D	C		B	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Cand	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	36%			5	36%			5	36%	0		5	42%	0	
B	6	43%			6	43%			6	43%	0		7	58%	1	
C	2	14%			2	14%			3	21%	1		0	0%	-3	
D	1	7%			1	7%			0	0%	-1		0	0%	0	
Votes Counted	14				14				14				12			
Overvote	1				0		-1		0				0			
Undervote	0				0				0				0			
Inactive Ballot	0				1				1				3		2	
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 12. Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering an overvoted ranking.

3.2.22.9.2.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet2_Overvote_Advance to next

Test Set 2	Overvote - Skip to next continuing candidate. 15 ballots listed. 1st choice overvoted (ballot 1). 2nd choice overvoted (ballot 8). Skipped choices are advanced to next highest ranked continuing candidate.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (1 overvote and 0 undervotes)
Description round 1	This would match round 0 if all ballots had valid 1st choice votes. Since 1st choice is overvoted in ballot 1 and a second chioce was omitted, both are skipped and C is selected. A new count is determined. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new candidate count is determined after Candidate D is eliminated on ballot 14 and the 2nd choice (candidate C) is selected instead. A new count is then determined. There still is no winner and candidate C will now be eliminated, having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 1, 3, 8 and 14. Ballot 1 is exhausted and becomes inactive as the 4th choice is candidate D, who has already been eliminated, and there are no more choices. Candidate B is selected on ballot 3, being the 2nd choice. Candidate A, the 3rd choice, is selected on ballot 8 as the 2nd choice is skipped due to being overvoted. The 3rd choice in ballot 14 was not voted so candidate B, as the 4th choice, is selected. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	Round 1	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	A,D		C	D	A,D		C	D	A,D		C	D	A,D		C	D
2	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	C	B,D	A		C	B,D	A		C	B,D	A		C	B,D	A	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	A			C	A			C	A			C	A			C
14	D	C		B	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	36%			5	33%			5	33%	0		6	43%	1	
B	6	43%			6	40%			6	40%	0		8	57%	2	
C	2	14%			3	20%	1		4	27%	1		0	0%	-4	
D	1	7%			1	7%			0	0%	-1		0	0%	0	
Votes Counted	14				15				15				14			
Overvote	1				0		-1		0				0			
Undervote	0				0				0				0			
Inactive Ballot	0				0				0				1		1	
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 13a - Ranked choice voting expected outcomes and sample test sets for skipping to the next valid ranking when encountering an overvoted ranking.

3.2.22.9.2.3 Expected_Outcome_RCV Test Sets_Single-winner - TestSet2_Overvote_Advance to next

With Batch Elimination

Test Set 2	Overvote - Skip to next continuing candidate. 15 ballots listed. 1st choice overvoted (ballot 1). 2nd choice overvoted (ballot 8). Skipped choices are advanced to next highest ranked continuing candidate.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (1 overvote and 0 undervotes)
Description round 1	This would match round 0 if all ballots had valid 1st choice votes. Since 1st choice is overvoted in ballot 1 and a second chioce was omitted, both are skipped and C is selected. A new count is determined. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new candidate count is determined after Candidate D is eliminated on ballot 14 and the 2nd choice (candidate C) is selected instead. A new count is then determined. There still is no winner and candidate C will now be eliminated, having the least votes. A new count is determined after candidate C is eliminated on ballots 1, 3, 8 and 14. Ballot 1 is exhausted and becomes inactive as the 4th choice is candidate D, who has already been eliminated, and there are no more choices. Candidate B is selected on ballot 3, being the 2nd choice. Candidate A, the 3rd choice, is selected on Ballot 8 as the 2nd choice is skipped due to being overvoted. The 3rd choice in ballot 14 was not voted so candidate B, as the 4th choice, is selected. Candidate B is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0)				Round 1				Round 2			
Ballot #	**1st Choice**	**2nd Choice**	**3rd Choice**	**4th Choice**	**1st Choice**	**2nd Choice**	**3rd Choice**	**4th Choice**	**1st Choice**	**2nd Choice**	**3rd Choice**	**4th Choice**
1	A,D		C	D	A,D		C	D	A,D		C	D
2	B	A	D	C	B	A	D	C	B	A	D	C
3	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D
8	C	B,D	A		C	B,D	A		C	B,D	A	
9	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C
13	A			C	A			C	A			C
14	D	C		B	D	C		B	D	C		B
15	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	**Total**	**%**	**Transfer**		**Total**	**%**	**Transfer**		**Total**	**%**	**Transfer**	
A	5	36%			5	33%			6	43%	1	
B	6	43%			6	40%			8	57%	2	
C	2	14%			3	20%			0	0%	-3	
D	1	7%			1	7%			0	0%	-1	
Votes Counted	14				15				14			
Overvote	1				0		-1		0			
Undervote	0				0				0			
Inactive Balllot	0				0				1		1	
Total Ballots	15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked counttinuing candidate

Table 13b - Ranked choice voting expected outcome and sample test sets for skipping to the next valid ranking when encountering an overvote and using batch elimination.

3.2.22.9.3 Repeated ranking of a previously ranked candidate

Step 6, in the RCV process flow description, enumerates two variants in processing repeat candidate ranking: Exhaust Ballot or Advance to next continuing candidate. The following sample test data sets demonstrate the processing of each of these variants respectively.

3.2.22.9.3.1 Expected_Outcome_RCV Test Sets_Single-winner - TestSet6_Repeat Ranking_Exhaust Ballot

Test Set 6	Repeat choice - Exhaust ballot. 15 ballots listed. 2nd choice a repeat of 1st choice on ballot 1.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvotes and 0 undervotes)
Description round 1	This matches round 0 since all ballots have valid 1st choice votes. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new count is determined after Candidate D is eliminated on ballots 1 and 14. Ballot 1 is exhausted and becomes inactive as the 2nd choice is a repeat of the 1st choice. On ballot 14 the 2nd choice, candidate B, is selected. There still is no winner and candidate C will now be eliminated, having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 2, 3, 11 and 13. Ballot 2 is exhausted as the 2nd choice is skipped having previously been eliminated and the 3rd choice being a duplicate of the 1st choice. Candidate B is selected on ballots 3 and 13, being the 2nd choice on both. Ballot 11 is also exhausted and becomes inactive due to 2nd choice being skipped since it was previously eliminated and the 3rd choice is a duplicate of the 2nd choice. B is the winer having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect B			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	D	D	A		D	D	A		D	D	A		D	D	A	
2	C	D	C	A	C	D	C	A	C	D	C	A	C	D	C	A
3	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	A	B	C		A	B	C		A	B	C		A	B	C	
9	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	C	D	D	A	C	D	D	A	C	D	D	A	C	D	D	A
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	C	B			C	B			C	B			C	B		
14	D	B	A		D	B	A		D	B	A		D	B	A	
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Candidate	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	33%			5	33%			5	36%	0		5	42%	0	
B	4	27%			4	27%			5	36%	1		7	58%	2	
C	4	27%			4	27%			4	29%	0		0	0%	-4	
D	2	13%			2	13%			0	0%	-2		0	0%	0	
Votes Counted	15				15				14				12			
Overvote	0				0				0				0			
Undervote	0				0				0				0			
Inactive	0				0				1		1		3		2	
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 14 - Ranked choice voting expected outcomes and sample test sets for exhausting a ballot when encountering a repeat ranking.

3.2.22.9.3.2 Expected_Outcome_RCV Test Sets_Single-winner - TestSet7_Repeat Ranking_Advance to next

Test Set 7	Repeat choice - Advance to next continuing candidate. 15 ballots listed. 2nd choice a duplicate of 1st choice on ballot 1.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvotes and 0 undervotes)
Description round 1	This matches round 0 since all ballots have valid 1st choice votes. This round does not produce a winner and candidate D, having the least votes, will be eliminated.
Description round 2	A new count is determined after Candidate D is eliminated on ballots 1 and 14. Candidate A, the 3rd choice, is selected on ballot 1 as the 2nd choice is skipped being a duplicate of the eliminated 1st choice. On ballot 14 the 2nd choice, candidate C, is selected. There still is no winner and candidate C will now be eliminated, having the least votes.
Description round 3	A new count is determined after candidate C is eliminated on ballots 2, 9, 13 and 14. Candidate A is selected on ballot 2, being the second choice and on ballot 14, being the next choice. Candidate B is selected on ballots 9 and 13, being the 2nd choice on both. Candidate A is the winner having more than 50% of the total remaining votes cast.

	Tabulated designated 1st choice-Round 0				Round 1-Elim D				Round 2-Elim C				Round 3-Elect A			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	D	D	A		D	D	A		D	D	A		D	D	A	
2	C	A	D	B	C	A	D	B	C	A	D	B	C	A	C	B
3	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
4	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D	B	A	C	D	B	A	C	D
8	A	B	C		A	B	C		A	B	C		A	B	C	
9	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D
10	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C	B	A	D	C	B	A	D	C
13	C	B			C	B			C	B			C	B		
14	D	C	A		D	C	A		D	C	A		D	C	A	
15	A	B	D	C	A	B	D	C	A	B	D	C	A	B	D	C
Skipped/Eliminated	Total	%	Transfer		Total	%	Transfer		Total	%	Transfer		Total	%	Transfer	
A	5	33%			5	33%			6	40%	1		8	53%	2	
B	5	33%			5	33%			5	33%	0		7	47%	2	
C	3	20%			3	20%			4	27%	1		0	0%	-4	
D	2	13%			2	13%			0	0%	-2		0	0%	0	
Votes Counted	15				15				15				15			
Overvote	0				0				0				0			
Undervote	0				0				0				0			
Inactive Ballot	0				0				0				0			
Total Ballots	15				15				15				15			

Current Active Choice	
Inactive Choice	`
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate

Table 15 - Ranked choice voting expected outcomes and sample test sets for advancing to the next valid ranking on a ballot when encountering a repeat ranking.

3.2.22.9.4 Threshold Calculation

Distinct from the above counting rule variants, which govern the tabulation of voter choices, are two variants involving the calculation of thresholds in multi-winner ranked choice voting (STV). These variants include:

3.2.22.9.4.1 **Whole # Threshold**

This test set uses a version of the Weighted Inclusive Gregory Method (WIGM) in which the threshold is calculated on a whole number basis.

[(Total ballots in round 1)/(Number of seats+1)]+1

and fractions ignored.

Surplus fractions are rounded down at the fourth decimal place and truncated.

Tabulation of multi-winner test data using whole # threshold, p. 1

Multi-winner with whole-number threshold	This test set uses a version of the Weighted Inclusive Gregory Method (WIGM) in which the threshold is calculated on a whole-number basis. (Total ballots in round 1)/(Number of seats+1)+1 with fractions ignored. Surplus fractions are truncated to four decimal places. The "exhaust after 2 consecutive skipped choices" rule is employed.
Description Round 0	Shows what a tabulator tape would report for totals of the 30 ballots. Only 1st choices would be reported. Overvotes and undervotes are shown to account for votes not assigned to candidates (2 undervotes). While candidate C has 8 first-choice votes, enough to be elected, some jurisdictions choose to run the RCV tabulation before declaring winners. As this example demonstrates, candidates may gain votes in the RCV tabulation that were not reflected in the straight 1st choice tabulation.
Description Round 1	Employing the rule allowing a maximum of one skipped vote, ballot 9 is counted for C and ballot 10 is "exhausted" and counted as permanently "inactive." A new count is determined. In this round, candidate C was elected with 9 votes, a surplus of 1 vote.
Description Round 2	C's surplus is divided by the number of votes for C to calculate the "surplus fraction" (SF) to be distributed to each next continuing candidate on C's ballots. This yields a SF value of .1111 which is redistributed to these candidates and a new count is determined. No new candidates are elected so candidate D, having the fewest votes, will be eliminated.
Description Round 3	Candidate D 's ballots are counted for the next continuing candidate on each. On ballot 1, the 2nd choice is an undervote so the 3rd choice (candidate E) is selected instead. On ballot 2, D's transfer value from C's surplus is counted for candidate F. On ballot 16, D is followed by 2 skipped choices and is thus exhausted and counted as "inactive." A new count is then determined and no new candidate is elected. Candidate F, having the fewest number of votes, will be eliminated.
Description Round 4	Candidate F 's ballots are counted for the next continuing candidate on each. Candiate B received full votes from candidate F from ballots 4 and 12 and the surplus transfer value from ballot 9. Candidate A received a full vote from candidate F from ballot 6 and the surplus transfer value from ballot 2. A new count is then determined and no new candidate is elected. Candidate E, having the fewest number of votes, will be eliminated.
Description Round 5	Candidate E's ballots are counted for the next continuing candidate on each. Candidate A received full votes from candidate E's ballots from ballots 1, 17 and 21 and the surplus transfer value from ballots 3 and 18. Candidate B received full votes from candidate E's ballot from ballots 24 and 27 and the surplus transfer value from ballot 14. Candidates A and B are elected with vote totals exceeding the threshold.
Description Round 6	This round is optional. Jurisdictions wishing to show all winning candidates as receiving the threshold, so that none appear to have more votes than others, may do so by exercising this option. The surpluses for A & B are calculated by subtracting the threshold from their respective vote totals and the surplus fraction is calculated by dividing the surpluses by the vote totals for each candidate. The surplus fraction is then multiplied by the transfer value of the vote on each respective ballot and transferred to the next continuing candidate. In this case, there are no remaining continuing candidates thus, all surplus transfers are exhausted ballots and are counted as inactive.

Table 16.1 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold. Round descriptions.

Tabulation of multi-winner test data using whole # threshold, p. 2

Parameters: electing 3 seats	
max skipped ranking =1	
Elected if vote total >= threshold	
T = int(V/(S+1)) + 1	
T=	8.0000
Adjusted Vote Totals	V=T

Elected
Eliminated

How each ballot counts in each round with totals by round

	Round 0	Round 1--Elect C		Round 2--Redist Surplus from C, Elim D						
Candidate	Vote Totals	Vote Totals		Surplus	Surplus Fraction		Total Votes	Adjusted Total Votes	Change from Prev Round		
A	6	6					6.0000	6.0000	0.0000		
B	5	5					5.0000	5.0000	0.0000		
C	8	9	elect	1.0000	0.11111	0.1111	8.0001	8.0000	-1.0000		
D	2	2					2.1111	2.1111	0.1111	elim	
E	4	4					4.7777	4.7777	0.7777		
F	3	3					3.1111	3.1111	0.1111		
Valid	28	29					29.0000	28.9999	-0.0001		
Inactive	2	1					1.0000	1.0000	0.0000		
Residual							0.0000	0.0001	0.0001		
Total	30	30					30.0000	30.0000	0.0000		

CVRs Active Ballot Inactive Ballot

Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	5th Choice	6th Choice
1	D	undervote	E	F	A	C
2	C	D	F	A	B	E
3	C	E	A	D	B	F
4	F	C	B	D	undervote	undervote
5	B	undervote	undervote	undervote	undervote	A
6	F	A	B	D	E	C
7	B	E	C	D	undervote	undervote
8	A	B	C	undervote	E	F
9	undervote	C	F	D	B	A
10	undervote	undervote	undervote	undervote	undervote	E
11	C	E	D	undervote	undervote	undervote
12	F	C	F	D	B	A
13	C	E	undervote	undervote	F	A
14	C	E	B	undervote	F	D
15	A	B	D	C	F	E
16	D	undervote	undervote	E	F	A
17	E	D	F	A	B	C
18	C	E	A	D	B	F
19	A	C	B	D	undervote	undervote
20	B	undervote	undervote	undervote	undervote	A
21	E	A	B	D	F	C
22	B	E	C	D	undervote	undervote
23	A	B	C	undervote	E	F
24	E	C	F	D	B	A
25	A	undervote	undervote	undervote	undervote	E
26	C	E	D	undervote	undervote	undervote
27	E	C	F	D	B	A
28	C	E	undervote	undervote	F	A
29	B	C	E	undervote	F	D
30	A	B	D	C	F	E

Ballot #	Round 0	Round 1	Original Vote Value				Round 2		Non-Transfer Value	+	Transfer Value
1	D	D	1				D		1		
2	C	C	1				C		0.8889	D	0.1111
3	C	C	1				C		0.8889	E	0.1111
4	F	F	1				F		1		
5	B	B	1				B		1		
6	F	F	1				F		1		
7	B	B	1				B		1		
8	A	A	1				A		1		
9	undervote	C	1				C		0.8889	F	0.1111
10	undervote	inactive	1				inactive		1		
11	C	C	1				C		0.8889	E	0.1111
12	F	F	1				F		1		
13	C	C	1				C		0.8889	E	0.1111
14	C	C	1				C		0.8889	E	0.1111
15	A	A	1				A		1		
16	D	D	1				D		1		
17	E	E	1				E		1		
18	C	C	1				C		0.8889	E	0.1111
19	A	A	1				A		1		
20	B	B	1				B		1		
21	E	E	1				E		1		
22	B	B	1				B		1		
23	A	A	1				A		1		
24	E	E	1				E		1		
25	A	A	1				A		1		
26	C	C	1				C		0.8889	E	0.1111
27	E	E	1				E		1		
28	C	C	1				C		0.8889	E	0.1111
29	B	B	1				B		1		
30	A	A	1				A		1		

Table 16.2 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold, Rounds 0–2.

Tabulation of multi-winner test data using whole # threshold, p. 3

	Round 3--Elim F				Round 4--Elim E				Round 5--Elect A & B			Round 6--Redist Surplus from A & B								
Candidate	Vote Totals	Change from Prev Round			Vote Totals	Change from Prev Round			Vote Totals	Change from Prev Round		Surplus	Surplus	Fraction	Vote Totals	Adjusted Vote Totals	Change from Prev Round			Transfer Value
A	6.0000	0.0000			7.1111	1.1111			10.3333	3.2222	elect	2.3333	#######	0.2258	8.0003	8.0000	-2.3333			0.025086	0.0250
B	5.0000	0.0000			7.1111	2.1111			9.2222	2.1111	elect	1.2222	#######	0.1325	8.0003	8.0000	-1.2222			0.014721	0.0147
C	8.0000	0.0000			8.0000	0.0000			8.0000	0.0000					8.0000	8.0000	0.0000				
D	0.0000	-2.1111			0.0000	0.0000			0.0000	0.0000					0.0000	0.0000	0.0000				
E	5.7777	1.0000			5.7777	0.0000	elim		0.0000	-5.7777					0.0000	0.0000	0.0000				
F	3.2222	0.1111	elim		0.0000	-3.2222			0.0000	0.0000					0.0000	0.0000	0.0000				
Valid	27.9999	-1.0000			27.9999	0.0000			27.5555	-0.4444					24.0006	24.0000	-3.5555				
Inactive	2.0000	1.0000			2.0000	0.0000			2.4444	0.4444					5.9993	5.9993	3.5549				
Residual	0.0001	0.0000			0.0001	0.0000			0.0001	0.0000					0.0001	0.0007	0.0006				
Total	30.0000	0.0000			30.0000	0.0000			30.0000	0.0000					30.0000	30.0000	0.0000				
Ballot #	Round 3	Non-Transfer Value	+	Transfer Value	Round 4	Non-Transfer Value	+	Transfer Value	Round 5	Non-Transfer Value	+	Transfer Value			Round 6		Non-Transfer Value	+	Transfer Value	+	Secondary Transfer Value
1	D		E	1	D		E	1.0000	D		A	1			D			A	0.7742	inactive	0.2258
2	C	0.8889	F	0.1111	C	0.8889	A	0.1111	C	0.8889	A	0.1111			C		0.8889	A	0.0861	inactive	0.0250
3	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	A	0.1111			C		0.8889	A	0.0861	inactive	0.0250
4	F	1.0000			F		B	1.0000	F		B	1			F			B	0.8675	inactive	0.1325
5	B	1.0000			B	1.0000			B	1.0000					B		0.8675	inactive	0.1325		
6	F	1.0000			F		A	1.0000	F		A	1			F			A	0.7742	inactive	0.2258
7	B	1.0000			B	1.0000			B	1.0000					B		0.8675	inactive	0.1325		
8	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		
9	C	0.8889	F	0.1111	C	0.8889	B	0.1111	C	0.8889	B	0.1111			C		0.8889	B	0.0964	inactive	0.0147
10	inactive	1.0000			inactive	1.0000			inactive	1.0000					inactive		1.0000				
11	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	inactive	0.1111			C		0.8889	inactive	0.1111		
12	F	1.0000			F		B	1.0000	F		B	1			F			B	0.8675	inactive	0.1325
13	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	inactive	0.1111			C		0.8889	inactive	0.1111		
14	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	B	0.1111			C		0.8889	B	0.0964	inactive	0.0147
15	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		
16	inactive	1.0000			inactive	1.0000			inactive	1.0000					inactive		1.0000				
17	E	1.0000			E	1.0000			E		A	1			E			A	0.7742	inactive	0.2258
18	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	A	0.1111			C		0.8889	A	0.0861	inactive	0.0250
19	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		
20	B	1.0000			B	1.0000			B	1.0000					B		0.8675	inactive	0.1325		
21	E	1.0000			E	1.0000			E		A	1			E			A	0.7742	inactive	0.2258
22	B	1.0000			B	1.0000			B	1.0000					B		0.8675	inactive	0.1325		
23	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		
24	E	1.0000			E	1.0000			E		B	1			E			B	0.8675	inactive	0.1325
25	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		
26	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	inactive	0.1111			C		0.8889	inactive	0.1111		
27	E	1.0000			E	1.0000			E		B	1			E			B	0.8675	inactive	0.1325
28	C	0.8889	E	0.1111	C	0.8889	E	0.1111	C	0.8889	inactive	0.1111			C		0.8889	inactive	0.1111		
29	B	1.0000			B	1.0000			B	1.0000					B		0.8675	inactive	0.1325		
30	A	1.0000			A	1.0000			A	1.0000					A		0.7742	inactive	0.2258		

Table 16.3 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a whole number threshold, Rounds 3–6.

3.2.22.9.4.2 **Fractional threshold**

This test set uses a version of the Weighted Inclusive Gregory Method (WIGM) in which the threshold is calculated on a fractional basis to four (4) decimal places.

(Total ballots in round 1/(Number of seats+1)+.0001)

Surplus fractions are rounded down at the fourth decimal and truncated.

Tabulation of mult-winner test data using fractional threshold, p. 1

Multi-winner with fractional surplus	This test set uses a version of the Weighted Inclusive Gregory Method (WIGM) in which the threshold is calculated on a fractional basis to four (4) decimal places: (Total ballots in round 1)/(Number of seats+1)+.0001. Surplus fractions are truncated to four decimal places. The "exhaust after 2 consecutive skipped choices" rule is employed.
Description Round 0	Shows what a tabulator tape would report for totals of the 30 ballots. Only 1st choices would be reported thus 2nd thru 6th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (2 undervotes)
Description Round 1	Employing the rule allowing a maximum of one skipped vote, ballot 9 is counted for C and ballot 10 is "exhausted" and counted as permanently "inactive." A new count is determined. This round does not produce a winner and candidate D, having the fewest votes, will be eliminated.
Description Round 2	Candidate D 's ballots are counted for the next continuing candidate on each. On ballot 1, the 3rd choice (candidate C) is selected instead. On ballot 16, D is followed by 2 skipped choices and is thus exhausted and counted as "inactive." A new count is then determined. Candidate C, having 8 votes is elected with a surplus of .7499.
Description Round 3	C's surplus is divided by the number of votes for C to calculate the "surplus fraction" (SF) to be distributed to each next continuing candidate on C's ballots. This yields a SF value of .0937 which is redistributed to these candidates and a new count is determined. No new candidates are elected so candidate F, having the fewest votes, will be eliminated.
Description Round 4	A new candidate count is determinined after Candidate F is eliminated on ballots 4, 6, 9 and 12. Ballots 4 and 12 are counted for B with a transfer value of 1 each. Ballot 9 is counted for B with a transfer value of .0937. Ballot 6 is counted for A with a transfer value of 1. The round does not produce a winner and candidate E, with the fewest votes, will be eliminated.
Description Round 5	Candidate E's ballots are counted, at their "transfer value," for the next continuing candidate on each. Ballots 2, 17 and 21 are counted for A with a transfer value of 1 each. Ballotw 1, 3 and 18 are counted for A with a transfer value of .0937 each. Ballots 14, 24 and 27 are counted for B with a transfer value of 1 each. Ballots 11,13, 26 and 28 are exhausted and counted as inactive with a transfer value of .0937 each. Candidates A and B are elected with vote totals exceeding the threshold.
Description Round 6	This round is optional. Jurisdictions wishing to show all winning candidates as receiving the threshold, so that none appear to have more votes than others, may do so by exercising this option. The surpluses for A & B are calculated by subtracting the threshold from their respective vote totals and the surplus fraction is calculated by dividing the surpluses by the vote totals for each candidate. The surplus fraction is then multiplied by the transfer value of the vote on each respective ballot and transferred to the next continuing candidate. In this case, there are no remaining continuing candidates thus, all surplus transfers are exhausted ballots and are counted as inactive.

Table 17.1 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Round descriptions.

Tabulation of multi-winner test data using fractional threshold, p. 2

Parameters: electing 3 seats			
max skipped ranking =1			
Elected if Vote Total >= threshold			
T=(V/(S+1)+n, 4)			7.2501
0<n<1	=	0.0001	
Adjusted Vote Totals	V=T		

Elected

Eliminated

How each ballot counts in each round with totals by round

	Round 0 1st Choice only	Round 1--Elim D		Round 2--Elect C		Round 3--Redist Surplus from C, Elim F						
Candidate		Vote Total		Vote Total		Surplus	Surplus Fraction		Round 3	Change	Vote Total	
A	6	6		6					6	0.0000	6.0000	
B	5	5		5					5	0.0000	5.0000	
C	6	7		8	Elect	0.7499	0.093738	0.0937	8	-0.7496	7.2504	
D	2	2	elim	0					0.0000		0.0000	
E	6	6		6					6	0.6559	6.6559	
F	3	3		3					3	0.0937	3.0937	elim
Valid	28	29		28					28	0.0000	28.0000	
Inactive	2	1		2					2	0	2.0000	
Residual												
Total	30	30		30					30	0.0000	30.0000	

CVRs — Active Ballot — Inactive Ballot

Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	5th Choice	6th Choice
1	D	undervote	C	E	F	A
2	E	D	F	A	B	C
3	C	E	A	D	B	F
4	F	C	B	D	undervote	undervote
5	B	undervote	undervote	undervote	undervote	A
6	F	A	B	D	E	C
7	B	E	C	D	undervote	undervote
8	A	B	C	undervote	E	F
9	undervote	C	F	D	B	A
10	undervote	undervote	undervote	undervote	undervote	E
11	C	E	D	undervote	undervote	undervote
12	F	C	F	D	B	A
13	C	E	undervote	undervote	F	A
14	E	C	B	undervote	F	D
15	A	B	D	C	F	E
16	D	undervote	undervote	E	F	A
17	E	D	F	A	B	C
18	C	E	A	D	B	F
19	A	C	B	D	undervote	undervote
20	B	undervote	undervote	undervote	undervote	A
21	E	A	B	D	F	C
22	B	E	C	D	undervote	undervote
23	A	B	C	undervote	E	F
24	E	C	F	D	B	A
25	A	undervote	undervote	undervote	undervote	E
26	C	E	D	undervote	undervote	undervote
27	E	C	F	D	B	A
28	C	E	undervote	undervote	F	A
29	B	C	E	undervote	F	D
30	A	B	D	C	F	E

Ballot #	Round 0	Round 1	Original Vote Value	Round 2	Original Vote Value				Round 3	Non-Transfer Value	+	Transfer Value
1	D	D	1	C	1				C	0.9063	E	0.0937
2	E	E	1	E	1				E	1		
3	C	C	1	C	1				C	0.9063	E	0.0937
4	F	F	1	F	1				F	1		
5	B	B	1	B	1				B	1		
6	F	F	1	F	1				F	1		
7	B	B	1	B	1				B	1		
8	A	A	1	A	1				A	1		
9	undervote	C	1	C	1				C	0.9063	F	0.0937
10	undervote	Inactive	1	Inactive	1				Inactive	1		
11	C	C	1	C	1				C	0.9063	E	0.0937
12	F	F	1	F	1				F	1		
13	C	C	1	C	1				C	0.9063	E	0.0937
14	E	E	1	E	1				E	1		
15	A	A	1	A	1				A	1		
16	D	D	1	Inactive	1				Inactive	1		
17	E	E	1	E	1				E	1		
18	C	C	1	C	1				C	0.9063	E	0.0937
19	A	A	1	A	1				A	1		
20	B	B	1	B	1				B	1		
21	E	E	1	E	1				E	1		
22	B	B	1	B	1				B	1		
23	A	A	1	A	1				A	1		
24	E	E	1	E	1				E	1		
25	A	A	1	A	1				A	1		
26	C	C	1	C	1				C	0.9063	E	0.0937
27	E	E	1	E	1				E	1		
28	C	C	1	C	1				C	0.9063	E	0.0937
29	B	B	1	B	1				B	1		
30	A	A	1	A	1				A	1		

Table 17.2 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Rounds 0-3.

Tabulation of mult-winner test data using fractional threshold, p. 3

	Round 4--Elim E				Round 5--Elect A & B				Round 6-Redistribute A & B Surpluses								
Candidate	Vote Total	Adjusted Vote Total	Change		Vote Total	Adjusted Vote Total	Change		Surplus	Surplus Fraction		Vote Total	Adjusted Vote Total	Change		Secondary Transfer Value	
A	6.0000	7.0000	1.0000		7.0000	10.2811	3.2811	Elect	3.0310	0.294813	0.2948	10.2811	7.2501	-3.0310		0.027623	0.0276
B	5.0000	7.0937	2.0937		7.0937	10.0937	3.0000	Elect	2.8436	0.281720	0.2817	10.0937	7.2501	-2.8436		0.026395	0.0263
C	7.2504	7.2501	-0.0003		7.2501	7.2501	0.0000					7.2501	7.2501	0.0000			
D	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000					0.0000	0.0000	0.0000			
E	6.6559	6.6559	0.0000	elim	6.6559	0.0000	-6.6559					0.0000	0.0000	0.0000			
F	3.0937	0.0000	-3.0937		0.0000	0.0000	0.0000					0.0000	0.0000	0.0000			
Valid	28.0000	27.9997	-0.0003		27.9997	27.6249	-0.3748					27.6249	21.7503	-5.8746			
Inactive	2.0000	2.0000	0.0000		2.0000	2.3748	0.3748					2.3748	8.2489	5.8741			
Residual		0.0003	0.0003		0.0003	0.0003	0.0000					0.0003	0.0008	0.0005			
Total	30.0000	30.0000	0.0000		30.0000	30.0000	0.0000					30.0000	30.0000	0.0000			

Ballot #	Round 4	Non-Transfer Value	+	Transfer Value	Round 5	Non-Transfer Value	+	Transfer Value				Round 6	Non-Transfer Value	+	Transfer Value	+	Secondary Transfer Value
1	C	0.9063	E	0.0937	C	0.9063	A	0.0937	0.066100	0.0661		C	0.9063	A	0.0661	Inactive	0.0276
2	E	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
3	C	0.9063	E	0.0937	C	0.9063	A	0.0937	0.066100	0.0661		C	0.9063	A	0.0661	Inactive	0.0276
4	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
5	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
6	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
7	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
8	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
9	C	0.9063	B	0.0937	C	0.9063	B	0.0937	0.067400	0.0674		C	0.9063	B	0.0674	Inactive	0.0263
10	Inactive	1.0000		0.0000	Inactive	1.0000		0.0000				Inactive	1.0000				
11	C	0.9063	E	0.0937	C	0.9063	Inactive	0.0937				C	0.9063	Inactive	0.0937		
12	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
13	C	0.9063	E	0.0937	C	0.9063	Inactive	0.0937				C	0.9063	Inactive	0.0937		
14	E	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
15	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
16	Inactive	1.0000		0.0000	Inactive	1.0000		0.0000				Inactive	1.0000	Inactive			
17	E	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
18	C	0.9063	E	0.0937	C	0.9063	A	0.0937	0.066100	0.0661		C	0.9063	A	0.0661	Inactive	0.0276
19	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
20	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
21	E	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
22	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
23	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
24	E	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
25	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948
26	C	0.9063	E	0.0937	C	0.9063	Inactive	0.0937				C	0.9063	Inactive	0.0937		
27	E	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
28	C	0.9063	E	0.0937	C	0.9063	Inactive	0.0937				C	0.9063	Inactive	0.0937		
29	B	1.0000		0.0000	B	1.0000		0.0000				B	0.7183			Inactive	0.2817
30	A	1.0000		0.0000	A	1.0000		0.0000				A	0.7052			Inactive	0.2948

Table 17.3 - Ranked choice voting expected outcomes and sample test sets for tabulation of multi-winner test data using a fractional threshold, Rounds 4-6.

3.2.22.9.5 Top-Two IRV Contingent Vote

For this voting method, after the initial count of first choices, all but the top two candidates are eliminated. The ballots of the eliminated candidates are redistributed to the higher ranked of the two continuing candidates on each ballot.

Top-2 IRV	After the initial count of first choices, this voting method eliminates all but the top two candidates and reallocates the ballots of the eliminated candidates to the next highest continuing candidate on the ballot.
Description round 0	Shows what a tabulator tape would report for totals of the 15 ballots. Only 1st choices would be reported thus 2nd thru 4th choices grayed out. Overvotes and undervotes are shown to account for votes not assigned to candidates. (0 overvotes and 0 undervotes)
Description round 1	Candidates C and D are eliminated and a new count is determined. Candidate A, the 3rd choice, is selected on ballot 1 as the 2nd choice is skipped being a repeat selection of the eliminated 1st choice. Candidate A is selected on ballot 2, being the second choice. Candidate B is selected on ballots 3 and 13, being the 2nd choice on both. Candidate A is selected on ballot 14, being the 3rd choice. Candidate A is the winner having more than 50% of the total remaining votes cast.
	Ballot advanced to next highest ranked continuing candidate

	Tabulated designated 1st choice-Round 0				Round 1			
Ballot #	1st Choice	2nd Choice	3rd Choice	4th Choice	1st Choice	2nd Choice	3rd Choice	4th Choice
1	D	D	A	undervote	D	D	A	undervote
2	C	A	D	B	C	A	D	B
3	C	B	A	D	C	B	A	D
4	A	C	B	D	A	C	B	D
5	B	C	A	D	B	C	A	D
6	A	C	B	D	A	C	B	D
7	B	A	C	D	B	A	C	D
8	A	B	C	undervote	A	B	C	undervote
9	B	C	A	D	B	C	A	D
10	A	B	D	C	A	B	D	C
11	B	C	A	D	B	C	A	D
12	B	A	D	C	B	A	D	C
13	C	B	undervote	undervote	C	B	undervote	undervote
14	D	C	A	undervote	D	C	A	undervote
15	A	B	D	C	A	B	D	C
Candidate	Total	%			Total	%		
A	5	33%			8	53%		
B	5	33%			7	47%		
C	3	20%			0	0%		
D	2	13%			0	0%		
Votes Counted	15				15			
Inactive	0				0			
Total Ballots	15				15			
Overvote	0							
Undervote	0							
Current Active choice								
Skipped/Eliminated Choice	Ballot advanced to next highest ranked continuing candidate							

Table 18 - Top-Two IRV Contingent Vote expected outcomes and sample test sets.

3.2.22.9.6 Presidential Preference Contests

Ballot Description	The ballot used for illustration contains a presidential preference contest and 2 delegate contests, both a vote for 3 with one for male delegates and the other for female delegates. For a delegate selection to count, the selected candidate must be affiliated with the presidential selection.
Example 1	Pres vote for Cand B. Only Cand B Delegates voted. Male Delegates fully voted. Only 2 Female Delegates voted causing undervote.
Example 2	Pres vote for Cand A. Only Cand A Delegates considered. B3 & Uncom1 Male and D1 Female Delegates ignored causing undervotes
Example 3	Pres vote for Uncommitted. Only Uncommitted Delegates considered. All 3 B Male Delegates ignored causing undervotes. Only 2 Female Uncommitted delegates causing undervote
Example 4	Pres. Selection overvoted. No delegate vote will be counted. Both Male & Female Delegates fully undervoted.
Example 5	No vote for Pres. Selection. No delegate vote will be counted. Both Male & Female Delegates fully undervoted.
Example 6	Pres vote for cand B. Male delegate, even though all cand B, it is overvoted. Female delegate counted since all cand B
Example 7	Pres vote for Cand A. Only Cand A Delegates considered. Even though Male Delegate overvoted, extra vote is ignored since Cand B.

Contests	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Party 1 Presidential Selection														
Candidate A			x	1			x						x	1
Candidate B	x	1									x	1		
Candidate C														
Candidate D														
Uncommitted					x	1	x							
overvote								1						
undervote										1				
Total		1		1		1		1		1		1		1
rule	Any 1 ok		Any 1 ok		Any 1 ok		ov		uv		any 1 ok		Any 1 ok	
Delegate Cong nn Male (vf 3)														
Candidate A1													x	1
Candidate A2			x	1			x		x					
Candidate A3							x		x				x	1
Candidate A4							x		x				x	1
Candidate B1	x	1			x						x			
Candidate B2					x						x			
Candidate B3	x	1	x		x						x		x	
Candidate B4											x			
Candidate B5	x	1												
Candidate C1														
Candidate C2														
Candidate C3														
Candidate UnCom1			x											
Candidate UnCom2														
overvote												3		
undervote				2		3		3		3				
Total		3		3		3		3		3		3		3
rule	All B can		only A can		no Uncom		no pres		no pres		ov		only A can	

Delegate Cong nn Female (vf 3)														
Candidate A1							x		x				x	1
Candidate A2			x	1			x		x				x	1
Candidate A3			x	1			x		x				x	1
Candidate B1	x	1									x	1		
Candidate B2	x	1									x	1		
Candidate B3											x	1		
Candidate C1														
Candidate C2														
Candidate C3														
Candidate D1			x											
Candidate D2														
Candidate D3														
Candidate D4														
Candidate UnCom1					x	1								
Candidate UnCom2					x	1								
overvote														
undervote		1		1		1		3		3				
Total		3		3		3		3		3		3		3
rule	1 uv B		only A can		2 sel		no pres		no pres		normal		normal	

Table 19 - Presidential Preference/Delegate contests sample test sets and expected outcomes.

3.2.22.9.7 Massachusetts Group Vote

Ballot Description	Ballot contains a full group, 2 incomplete groups, candidates running out outside a group and write-ins. This illustrates the different ballot conditions that may be encountered although not all actual ballots have multiple groups. Tabulation of group selects is optional and is not included in total contests votes. It is shown in the table as an "o" to indicate recognition but not necessarily results tabulation. Note that any group selection is equivalent to selecting all candidates in the group inidvidually.
Example 1	Selection of a full group only. All candidates in the group receive a vote.
Example 2	Selection of an incomplete group plus additional individual candidates for fill the available votes. The additional candidates may not be in a group, may be a write-in or even in another group.
Example 3	Selection of a full group and an additonal candidate. The additional candidate creates an overvote when added to the candidates selected by the group select.
Example 4	Selection of a group plus selection of some of the individual candidates within the group. The individual selections are inconsequention as all group candidates are always selected by the group selection.
Example 5	Selection of an incomplete group only that shows undervote tabulation.
Example 6	Selection of individual candidates without any group selections.
Example 7	Selection of 2 groups that result in an overvote
Example 8	Selection of 2 groups that do not result in an overvote

Contest	Example 1		Example 2		Example 3		Example 4		Example 5		Example 6		Example 7		Example 8	
	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated	Choices	Tabulated
Town Committee (vote for 15)																
Group A Select	x	o			x		x	o					x			
Candidate 1-GA		1						1								
Candidate 2-GA		1						1			x	1				
Candidate 3-GA		1					x	1								
Candidate 4-GA		1					x	1								
Candidate 5-GA		1					x	1								
Candidate 6-GA		1						1								
Candidate 7-GA		1						1			x	1				
Candidate 8-GA		1						1								
Candidate 9-GA		1					x	1			x	1				
Candidate 10-GA		1					x	1								
Candidate 11-GA		1					x	1			x	1				
Candidate 12-GA		1					x	1								
Candidate 13-GA		1						1								
Candidate 14-GA		1						1			x	1				
Candidate 15-GA		1						1								
Group B Select			x	o					x	o					x	o
Candidate 1-GB				1						1	x	1				1
Candidate 2-GB				1						1						1
Candidate 3-GB				1	x					1	x	1				1
Candidate 4-GB				1						1	x	1				1
Candidate 5-GB				1						1	x	1				1
Candidate 6-GB				1						1						1
Candidate 7-GB				1						1						1
Candidate 8-GB				1						1	x	1				1
Candidate 9-GB				1						1						1
Candidate 10-GB				1						1						1
Candidate 11-GB				1						1	x	1				1
Candidate 12-GB				1						1						1
Group C Select													x		x	o
Candidate 1-GC																1
Candidate 2-GC			x	1							x	1				1
Candidate 3-GC																1
Ungrouped Cand																
Candidate 1-UnGr																
Candidate 2-UnGr			x	1							x	1				
Candidate 3-UnGr																
Candidate 4-UnGr											x	1				
Candidate 5-UnGr																
Candidate 6-UnGr																
Write-in 1			x	1												
Write-in 2																
Write-in 3																
Write-in 4																
Write-in 5																
Write-in 6																
Write-in 7																
Write-in 8																
Write-in 9																
Write-in 10																
Write-in 11																
Write-in 12																
Write-in 13																
Write-in 14																
Write-in 15																
overvote						15								15		
undervote										3		1				
Total		15		15		15		15		15		15		15		15
rule	vt grp		grp + ind		grp+ov		GrpA+CA		Inc grp		Ind sel		ov grp		vt 2 grps	

Table 20 - Massachusetts Group Vote sample test sets and expected outcomes.

4 Voting Methods Models

Introduction

This section provides voting methods specification modules with descriptive definitions of voting methods variants in use in the United States today.

The voting methods modules in this section are representations of the plain language algorithmic definitions extracted from local jurisdiction legislation, rules, and other documentation that define voting methods as implemented in practice and operation in the United States of America. The modules presented in this section are representations designed to be sufficiently general to be instantiated for the variety of voting method variants as they may occur in practice in government-operated elections in the United States. The voting methods modules defined and illustrated in this section are based on a descriptive model of existing legislation, local jurisdiction rules, and practice rather than a prescriptive model of each voting method.

Any use of 'shall' and the permissive 'may' in this standard specification are intended to provide for the use of this specification either as a voluntary guideline or as requirements for stakeholders choosing to adopt this standard for their requirements specification officially. Use cases for this specification as requirements include where law requires a local jurisdiction to use a particular voting method variant, where a jurisdiction may decide to use elements of this speccification as contractually required by a voting systems vendor, or where a manufacturer seeks to adopt this specification as a standard for the voting or tabulation methods that the manufacturer product supports or is certified as supporting. A use case for legislators is a specification in the legislation of voting and tabulation methods where the legislators choose to align with a standard. To support such use cases, each voting method module has a unique identifier that may be used to incorporate the desired voting method specification by reference. In further support of the use cases, we have endeavored to make each voting method module a complete specification.

4.1.1 Terminology

The purpose of this subsection section is to present terminology definitions. Throughout this section, Section 4, Voting Methods Models, wherever the terms "elected," "winner," or "passed" are used, the intention is to describe the apparent outcome of a contest, as determined by the application of the voting method's decision rule set to the contest counts and statistics produced by application of that same voting method's tabulation process set - distinct from the actual determination of contest outcome that is the prerogative of elections officials. To be clear, we do not intend to imply that any software/hardware system is authorized to declare official winners of a contest nor whether a measure has officially passed.

We briefly introduced three key definitions in section 2. Background and Overview. Here we provide more information about why and how we developed the definitions of voting methods, tabulation methods, and we provide a comprehensive list with descriptions of the

tabulation scenarios.

The Voting and Tabulation Methods Working Group (VMWG) defined voting methods, tabulation methods, and tabulation scenarios for the purpose of solving specific problems known to occur in practice. The standards or specifications that are adopted by jurisdictions and other election stakeholders must be understandable without ambiguity when those stakeholders implement election operations and systems.

Our model of the voting methods and tabulation universe treats the concepts of "voting method" and "tabulation method" as distinct and separate. This separation is driven by the variety of tabulation scenarios that have been implemented in practice. Tabulation methods must be defined as a concept separate from voting methods because a voting method has a broader scope of functionality than a tabulation method that is used by a voting method or practical tabulation scenarios that are implemented by jurisdictions.

A voting method contains a set of one or more tabulation methods. A key differentiator from a tabulation method is that a voting method also includes a decision rule set applied to determine a contest outcome.

Any particular voting method encapsulates a set of tabulation methods that the voting method uses to produce tabulation outcomes as sources of input to the voting method's application of its decision rule set. When viewing the voting method as a process, it is the voting method that orchestrates the operation of executing the applicable set of one or more tabulation methods that satisfy the requirements and constraints of the voting method. Consider the following as one way of demonstrating the aforementioned relationship between a voting method and its tabulation method(s) and understanding the utility provided by how the VMWG has defined a voting method, tabulation method, and tabulation scenario as distinct concepts.

In the set of common in-use tabulation scenarios, many of the scenarios do not, cannot, and are not intended to determine the outcome of a contest.

The most common tabulation scenario variants do not and cannot result in a total contest count because the tabulation is performed as a batch operation or as a distributed and batched set of operations.

An example use case is tabulating totals at the precinct level. The count for the precinct (or ward) – a subset of the total contest – is reported as the outcome of that particular "run" or execution of a tabulation process. The subset tabulation count and associated statistics form a data set and are transmitted for the purpose of being combined or aggregated with the subset counts of other precincts. The multiple subsets of a whole contest count are transmitted to aggregate to a county-wide count. A similar illustrative example is county-wide counts aggregated through a higher-level process for arriving at a state-wide

count. A third example use case is when the law requires a jurisdiction to produce reporting and vote selection data subsets for a contest on partial or incremental counts.

Multiple voting methods in practice may use the same tabulation methods even though the voting methods are distinct and differ in ways required by their governing legislative and local rules. In other words, tabulation methods and their corresponding tabulation process flow are a set of tools in the toolbox used by the voting method to achieve whole contest outcome determination within the particular constraints of a tabulation scenario.

For example, most voting methods use an N-of-M, or baseline, tabulation method.

- For a particular N-of-M voting method variant, the means of marking ballots, determining the validity of vote selections on ballots, determining what constitutes a blank ballot, and other factors may differ among distinct voting method variants using the same N-of-M tabulation method. Irrespective of those differences, all N-of-M voting method variants use the same tabulation method. Illustrative examples include straight party or party preference contests and plurality.

- The contest outcome decision rules set can differ for distinct voting method variants even though the variants use the same N-of-M tabulation method. For example, plurality and majority voting methods differ in their decision rules for determining outcomes, but the tabulation method applied is the same.

Tabulation scenarios must be feasible to operationalize across the set of voting method variants for which they are applicable. In addition to contest tabulation, the complete set of use cases for tabulation includes auditing, ballot accounting, capacity planning, and L&A testing, among others.

The definitions and their relationships as distinct concepts are designed to solve practical problems identified as common in the field and to provide utility to stakeholders and fitness to purpose.

One of the practical problems is the reduction of ambiguity in language. Ambiguity is known to increase risk and cost where there is a need for precise common understanding in legislation or election legal code, RFPs/RFIs, and contracts between EOs and manufacturers and ISVs. Other areas where precise specification helps to reduce cost and risk include interpretation of and compliance with standards and other documentation and rules jurisdictions follow for operating elections or implementing election systems.

Our definitions of the voting method, tabulation method, and tabulation scenario are as follows, and as adopted by VVSG 2.0 following the VMWG's presentation to the TGDC.

Voting Method
A voting method encompasses a set of partially ordered process steps, tabulation methods, and decision rules. Each voting method has at least one associated tabulation method and

at least one election outcome decision rule set consisting of at least one decision rule. Executing the appropriate tabulation method and election contest decision rule set is in the scope of the voting method.

Tabulation Method
A tabulation method is a set of partially ordered process steps and operations for performing counting, accumulation, aggregation and other data and math operations on vote selection data sets. Each tabulation method supports at least one tabulation scenario.

Tabulation Scenario
A situation and sequence of events or activities intended by Elections Administrators to be sufficient to achieve a tabulation goal.

A tabulation scenario is a situation and sequence of events, steps in process, and tabulation state changes. Sets of ballots, in the form of NIST 1500-103 Cast Vote Record Cast Vote Record data sets, are operated upon according to the jurisdiction's defined process of tabulation being executed, with the objective of producing an output count and related statistics, and other reporting for a particular instance of tabulation.

A tabulation scenario may be intended to count only a subset of the total number of ballots needed to count for an election contest result that is determined and output by a Voting Method. Examples of tabulation scenarios include but are not limited to configuration or logic and accuracy testing, batch tabulation, precinct tabulation, central tabulation, county or statewide 'rollups' or aggregations of tabulation batches, recounts, and audits. Tabulation scenarios are also use cases for the voting methods that execute the applicable set of one or more tabulation methods.

This specification currently supports tabulation scenarios as follows.

Tabulation Scenarios

- **Batch:** a collection of voted ballots that are processed as a group for tabulation, results reporting, and/or auditing. For example, a batch may consist of a set or sets of ballots from a specific polling place, absentee ballots received during a specific day, or overseas ballots.
- **Batch with Accumulation:** a batch of ballots with the report of results obtained from the tabulation of the ballots. It may be used for auditing.
- **Election Contest:** a contest or measure that is being submitted for voter preference by voters eligible to make a choice during an official election. A tabulation scenario where the tabulation goal is tabulating a particular contest and may have a goal to tabulate any subset of the contest or the entire contest.
- **Reporting Unit:** The smallest geographical subdivision by which results are reported. Typically, the subdivision is by Precinct. Precincts are also known as Wards or Election Districts in some jurisdictions, or Precinct/Split in jurisdictions

where precincts may be split by one or more districts that are active in the election. The tabulation goal is to tabulate the ballots in the form of CVR for the specific reporting unit.

- **Aggregation:** The process by which election results are "rolled up" into a higher level for reporting of consolidated results. In other words, counties aggregate polling place results; states aggregate their multiple-county results. Aggregation, as defined here for tabulation scenarios, is a term of plain language used by election officials in the sense of combining data sets in a database. Thus, for the purpose of this specification, aggregation is defined differently than the term defined for statistical aggregate mathematical functions or operations. The tabulation goal is to produce the jurisdiction's specified reporting of counts and statistics of consolidated results.
- **Compositions of Batch and Contest variants with Accumulation and Aggregation:** results may be aggregated into contest results by Reporting Unit, by category, by group, or by batch. These categories may be election day, early voting, voting by mail, absentee voting, and voting from overseas. The detail within the category is typically kept by reporting unit elements such as by Precinct. The tabulation goal is to produce the specified composition of multiple tabulation scenarios that the jurisdiction uses.
- **Multi-Contest Multi-Voting Method per Ballot:** A ballot where more than one contest is represented on a single ballot, and the contests use different voting methods. A variant of the Multi-Contest Multi-Voting Method per Ballot is a ballot that contains contests where voting on one contest affects what can be voted/tabulated on another contest on the same ballot, illustrating multiple voting methods with a controlling contest and one or more corresponding controlled contests. Illustrative examples of multi-contest multi-voting method per ballot include the following:
    - A ballot typically contains an assortment of contests unless it is a special election for an office where the office holder has to be replaced before the next scheduled election or the ballot contains an emergency referendum.
    - General elections usually contain contests with candidates that are party-affiliated and may also contain referenda that are nonpartisan.
    - When the straight party voting method is one of the voting methods in use on the ballot, then the combination of multiple contests illustrates multiple voting methods per ballot with the controlling contest variation because the voter has the option of making selections either individually for each contest or as a straight party selection.
    - When a cumulative, Approval, or ranked voting contest is on a ballot with other types of contests, then there are multiple different voting methods on one ballot.
    - Open Primary ballots, with or without a Party Preference contest.
- **Distributed tabulation scenarios:** An instance where ballots are voted in different

manners or locations and under different conditions. These conditions include different voter categories, different location categories, different systems for voting and different tabulation systems. Example: A polling place (location), whether used for Election Day or Early Voting/In-Person Absentee voting (voter category), would have an ADA unit, which can be a ballot marking device (BMD) or DRE (voting systems), for accessibility purposes as well as a precinct count optical scanner (PCOS) (tabulation systems) if paper is used in the polling place irrespective of whether all voters use a BMD.

- **Precinct tabulation:** The designation is usually given to the smallest election results reporting unit in a jurisdiction. This name can vary in some states. Other names are Ward (Wisconsin) and Election District (New York). In this scenario, tabulation occurs for the vote selection data set of the precinct.
- **Central Tabulation:** The designation is given to the process by which voted ballots are collected and tabulated, and results are reported at a central location by the jurisdiction administering the election. Typically, this tabulation scenario applies to paper-based ballots. This process is used in most states for absentee ballots that are mailed in or hand-delivered.

Tabulation scenarios for the auditing use cases

- **Tabulation scenarios for the auditing use case:** The auditing process regarding ballot tabulation accuracy varies by state. Some jurisdictions or states require re-tabulation of a percentage of polling places chosen "at random" to audit digital results. Others may recommend or require that the re-tabulation be conducted using a different tabulator unit or a different type of tabulator than originally used.

  Some states use an RLA process to ensure that the tabulation results in the correct outcome. The content of a given number of randomly selected ballots, the number based on the vote difference between the apparent winner and loser, is compared to the content of the associated CVRs to determine accuracy. In all audit methods, discrepancies between the paper record and the digital record should be investigated.

- Auditing tabulation scenarios and corresponding tabulation processes are used to ensure that a jurisdiction's results are correct and that vote-counting tabulation results in the proper contest outcomes. Audit scenarios occur in three categories: tabulation of ballots as intended by the voter, auditing for the validity of all ballots cast, and consolidation of results being accurate.

- Auditing the accuracy of ballot tabulation as intended by the voter varies by state and primarily applies in practice to paper ballot systems or DRE units with a paper trail. Some states require re-tabulation of a percentage of polling places (e.g., 5%) chosen at pseudo-random to verify original results. Some states (e.g. PA)

recommend re-tabulation on a different unit (or different type of unit) than the unit originally used. Ohio re-tabulates all contests and ballots, considering the initial tabulation as "the unofficial" tabulation and the re-tabulation as "the official" tabulation. Other states use a risk-limiting audit (RLA) process to ensure that the apparent elected or passed measure is determined correctly and with an appropriate degree of certainty. The content of a number of randomly chosen ballots is compared to the content of the associated Cast Vote Record (CVR) to determine accuracy. The number of chosen ballots is based on the vote difference between the apparent winner and loser, the specified degree of certainty, and other factors depending upon the particular case, including but not limited to ballot comparison, batch comparison, or ballot polling. It is outside the scope of this standard to set a standard for RLAs. When using DRE units with a paper trail, tabulation may involve tabulating the contest selections from the paper tape record and verifying the results against the electronically tabulated results. In all audit methods, discrepancies between the paper record and the digital record should be investigated.

- Auditing the validity of the tabulated ballots is intended to ensure that all ballots that have been voted have been included in the results and that no ballots have been substituted for legally voted ballots nor illegally voted ballots added to the tabulated results. Some jurisdictions or EAs label this type of audit "ballot accounting." Methods for achieving this vary by state or jurisdiction, but there are two considerations for this audit. One is to verify at each polling/tabulation location that the number of voters that have cast ballots as indicated in the poll book matches the number of ballots tabulated at the location. The second is an audit of the disposition of the inventory of official ballots accounts for all that were printed and issued. At a polling place, this ensures that the total number of ballots voted, the number of ballots that were spoiled, and the number of unused ballots equals the number of ballots issued to the polling place.

- Auditing the accuracy of the consolidation of results applies to virtually all jurisdictions that are administering an election but certainly applies whenever contest results must be consolidated or aggregated from a multiplicity of these jurisdictions to count a complete set of results (e.g., a state aggregating results from counties for state offices). A jurisdiction typically uses results from multiple tabulators to aggregate into jurisdiction-wide results. Tabulation results to be aggregated for a jurisdiction include results from each polling place, which may have been tabulated by one or more units at the polling place, as well as any centrally located tabulators handling mail-in ballots or ballots from other sources that may be tabulated by hand. Requirements for this audit vary by state or jurisdiction. Illinois, for example, requires polling place tapes (which have precinct-level results if the polling place services multiple precincts) to be verified

against the aggregate canvass report that lists results by precinct.

- **Tabulation scenarios for the recount use case:** A recount is a process for verification of an execution of a voting method or a tabulation method or scenario subsequent to an original initial tabulation of a contest or set of contests.

  A recount may be required when a certified result is challenged and ordered by the courts or when a contest meets automatic recount requirements indicated in state statute. It is context-specific, and one or more recounts may be conducted concurrently if more than one contest requires a recount. Multiple types of systems may be involved in a recount. If the recount will be done electronically, the election definitions for equipment used in the recount may be modified to include only the contests being retabulated, the ballot styles that the contest(s) appear on and the precincts using those ballot styles. For paper-based ballots, whether hand-voted or voted on a BMD, one or more central count optical scan (CCOS) units may be used if available. Smaller jurisdictions may use one or more PCOS units for this purpose. If DREs are used in the election, the system may be used to reaggregate the results from the CVRs using the media where the CVRs are stored. If the DRE has a paper trail (VVPAT or RTAL), the paper trail can be used to manually or automatically re-tabulate the results from the printed paper trail (which may have printed bar codes for this purpose). Jurisdiction recount rules apply. For close elections, a full manual recount may be required. Florida, for example, requires an automatic machine recount if the apparent winner and highest loser's results are less than 0.5% of the votes cast. If the margin is less than 0.25%, Florida requires a manual recount. Florida also has a process by which election officials separate out all ballots for the recounted contest that are not fully voted (undervoted or overvoted). These are then manually reviewed.

- **Tabulation for logic and accuracy testing:** Logic and Accuracy (L&A) testing is performed prior to each election. Its purpose is twofold. It verifies the correct election-specific setup or configuration for each unit that will be used for voting and/or tabulation in the upcoming election, as well as to verify that each unit's hardware and software are functioning correctly. It usually is comprised of simulated voting a complete set of ballots with known results such that every ballot style is voted and that each candidate in every contest receives votes in a way that no two candidates within a contest receive the same number of votes. Tabulated results are compared to the expected results. Complete end-to-end testing is recommended so that the results aggregation/accumulation process is also validated to produce the expected results. Most states require a "Public Test" where a portion of this testing is open to the public and performed at a publicized time and location.

- **Tabulation for configuration testing:** Configuration testing is performed to verify that the system can be used with all types of elections that the system is specified to accommodate. Configuration testing is similar to L&A testing but is both less

specific and more comprehensive. Testing includes using a suite of elections, each with ballots that are voted with known results so that the outcomes can be verified against the expected results in each type of election. This type of testing may be used by a jurisdiction when accepting delivery of a new system or during certification testing for Federal or State approval.

See Section 3.1.3 Tabulation Methods for more information on tabulation methods.

For specification of election contest outcome tests for voting methods covered in this specification, see Section 3.1.3 Test Conditions and Expected Outcomes.

For additional terminology, see NIST Voting Glossary at https://collaborate.nist.gov/voting/bin/view/Voting/Glossary.

With the relevant terminology defined, this specification now turns to describing how the various voting methods, tabulation methods, and tabulation scenarios operate together to produce contest outcomes.

Legislative links provided represent examples of the voting method implementation presented and are not intended to constrain the options adopted by any local jurisdiction.

4.2 Voting Methods Modules

4.2.1 Domain of discourse, notation, and dictionary

Our Domain-Specific Language (DSL) is based on mathematical logic operators and standard math operators combined with labels for variables and constants, data structures, and objects. The labels for constants and variables in the elections universe are selected to correspond to the language and terms of art in use in the field for operating US elections. The format of the labels is aligned with principles of readability and self-documentation in the field of software engineering. It incorporates the 'camelCase' formatting of labels as a readability feature.

Our decisions for the design of DSL were made to provide the best effort for unambiguous and human-readable DSL that minimizes the need for a reader to look back and forth to consult the dictionary. We sought to create a symbology that was understandable to users without special training and adequately represented constraints in the form of constraint logic. Translation into another DSL, for example, a DSL specific to a particular constraint satisfaction solver th, theorem prover, or some other representation framework, is thusly supported.

VMWG Domain Specific Language Dictionary

The complete VMWG DSL dictionary may be found in the Appendix at this link:
https://docs.google.com/document/d/1iXHWsFBqNdnZG8XcjFM1MdHVEflTeA9a8pKu4jLNkYM/edit?tab=t.0#bookmark=id.dsc4qf5e2xj

Domain Specific Language (DSL) Notation

Symbol	Meaning

∀	**For all; for any; for each; universal quantification**
∃	**There exists; existential quantification**
∃!	**There exists precisely one; singleton existential quantification**
∄	**There does not exist;**
∧	**Logical AND** **Logical conjunction**
∨	**Logical OR** **Logical inclusive disjunction**
⊻	**Logical XOR; exclusive OR**
⇒	**implication; implies; if…then**
⇔	**if and only if; iff**
¬	**negation**
T (or True)	**True**
F (or False)	**False**
()	**Precedence grouping**
≡	**Equivalence**
+. -, *, /, <, >, =, etc.	**Usual meanings in arithmetic**
+. -, *, /, <, >, ≥ ≤ ≪ ≫ =, ≈ ∼ ≠ ± ± ÷ % mod Δ	**Usual meanings in arithmetic**
∩ ∪ ⊆ ⊂ ⊄ ⊇ ⊃ ⊅ { } Ø ∈ ∉ | 𝕌	**Usual meanings in set theory**
∑ ∑∑ ∏	**Usual meanings, summation and multiplicative**
min, max, argmax	**The usual meanings**

Symbols, notation representing numerical values

Term	Definition
M m, m_i	M: Available Contest Options m: Number of contest options in the list available to the voter. Discussion: A contest option may be Yes or No or may be For or Against in a Referendum/Question or a specific candidate running for an elected office.
N n, n_j	N: Number of Allowed Contest Option Selections n: The number of selections allowed to be allocated by the voter in the contest and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n). Discussion: For a Yes/No Referendum/Question or a plurality single-seat contest, N would be 1. In a plurality multi-seat contest, N would equal the number of seats. In Approval Voting N would be the same as M. In Limited Voting, N would be less than the number of seats to fill.
V v, v	V_{Mi} = number of votes for the ith listing of candidate M or the ith contest option for candidate M. V_{Cii} = number of votes for the ith listing of contest option C or the ith contest option for contest option C, and S = sum of V_C for all the contest options.

4.2.2 Constraints that generally apply to most Voting Methods

4.2.2	Constraints that generally apply to most Voting Methods, High-Level
a.	**Description** This module contains constraints on counting rules and rules applicable to counter operations that generally apply to most voting method variants. Where these generally applicable rules do not apply, the counting rules are specified in this section's specific applicable voting method variant module.
b.	**Constraints** A voting method shall produce the correct result by applying its specified tabulation method set and decision rule set for the jurisdiction's desired tabulation scenarios or those supported by the implementation of the voting method. The voting method shall correctly execute the tabulation methods process set defined for the voting method for one or more applicable tabulation scenarios.

The decision rule set for the voting method that an election contest is configured for or the jurisdiction's legislation requires that it shall be correctly executed to determine whether elected or passed when all voter selections of the contest have been counted.

Counting Rules Constraints:
See also the specific Voting Method Variant specification in this section where applicable.
All voter selections, or votes, are deemed to have been counted when the election contest tabulation scenario has reached the point in the process where it has resulted in a complete unofficial canvass for the election contest.

All statistics required by jurisdictions governing legislation or rules must be counted, calculated, and reported.

The system must provide counters for each contest for all variables required to be counted or reported.

The total of those counters must be consistent with ballot, contest, or contest selection accounting reconciliation. For example, vote selection counters must be equal to no more than the total number of allowed votes on ballots cast.

All counters must be configured to be initially set to zero and constrained to monotonically incrementing operations during the election.

All counters are required to detect and prevent an impending overflow condition and suspend the operation of the voting method or execution of its tabulation method when an impending or actual overflow is detected.

Counters must be structured with sufficient capacity and precision for the range of expected values.

Overflows or exceptions of counters must be logged and reported.

All contest outcomes produced by their Decision Rules and Decision Rule Sets are subject to the determination of election officials.

Common Constraints on Counting Rules:

Plain Language:
A count is a multistep process by which:

1. Voter selections on each contest on a ballot are processed to determine valid votes for each selection in each contest according to selection rules, contest constraints, and

contest tabulation rules.

2. A cast vote record is recorded showing the detailed results from the ballot.
3. The number of votes received is added to the accumulated count for each contest option receiving votes.

In a typical N of M contest, the amount increment is 1. However, this can vary in special contest types, such as cumulative voting and proportional voting contests. All counters must be configured to be initially set to zero and constrained to monotonically incrementing operations during the election.

Mathematical logic DSL:

For the case where monotonically incrementing operations are constrained to one.

∀ contest (validBallotContestCount) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (contestCounterInitialValue = 0)
∧ (contestCounterIncrement = 1)
∧ (contestCounterValue = (contestCounterValue + contestCounterIncrement))
∧ (contestCounterValue <= ballotsCastCount) }

Where N is the number of selections allowed to be allocated by the voter in the contest

For the case where monotonically incrementing operations are not constrained to one:

∀ contest (validBallotContestCount) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (contestCounterInitialValue = 0)
∧ (contestCounterIncrement > 0) ∧ (contestCounterIncrement >= N)
∧ (contestCounterValue = (contestCounterValue + contestCounterIncrement))
∧ (contestCounterValue <= ballotsCastCount) }

Where N is the number of selections allowed to be allocated by the voter in the contest

Overvote

	Plain Language: Overvote is True if and only if the voter selects more contest options than allowed by the jurisdiction for a contest or a part of a contest, e.g., a ranking. **Mathematical logic DSL:** ∀ contest (overVote) ⇔ { (selectedBallotContestOptionsCount > MaxContestSelectionsAllowed) } **Undervote** **Plain Language:** Undervote is True if and only if the voter selects fewer contest options than allowed by the jurisdiction for a valid non-RCV ballot contest or if the voter ranks no contest options in an RCV contest. **Mathematical logic DSL:** ∀ contest (underVote) ⇔ { (selectedBallotContestOptionsCount < MinContestSelectionsAllowed) }
c.	**Decision Rule Set:** See Section 4.2.3 Decision Rule Constraints and the specific applicable voting method variants in Section 4. **Determine Outcome Rule:** See Section 4.2.3 Outcome Rule Constraints and the specific applicable voting method variants in Section 4.
d.	**Authoritative References** **Overvote:** N.Y. Comp. Codes R. & Regs. tit. 9, § 6210.13(7). N.C. Gen. Stat. § 163-182.1(a)(4). **Undervote:** N.Y. Comp. Codes R. & Regs. tit. 9 6210.13(6). **Zero report:** Election Assistance Commission, Voluntary Voting System Guidelines 2.0 § 1.1.3-B

4.2.3 High-Level Decision Rule Constraints Applicable to Voting Methods

4.2.3	**High-Level Decision Rule Constraints Applicable to Voting Methods**
a.	**Description** **Determine Prevailing Candidate(s)** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes in the final tally is the prevailing candidate presented to election officials to be certified as elected. For a multi-winner contest, the candidates who meet the specified criteria, e.g., threshold, are presented as the prevailing candidates. Depending on the variant of the voting method, additional or substitute criteria may have to be met to determine a prevailing candidate. **Example: Majority Rules Variant:** The prevailing candidate is the one with more than fifty percent (> 50%) of the thresholdBase, typically the total votes cast for the contest. **Determine Outcome of Ballot Measure** For passage of a ballot measure with only yes/for and no/against contest options, the percentage of yes votes must exceed the required threshold, e.g., 50% or 55% of the total yes and no votes. The outcome is presented to election officials to certify the passage or failure of the measure. If a ballot measure consists of multiple parts that contradict each other or multiple contradictory measures are on one ballot, a typical decision rule is that the part of the measure with the most votes that ALSO exceeds the required threshold is the prevailing measure. Subject to certification by the certifying authority.
b.	**Constraints** In addition to the constraints that are high level and apply to most voting methods, described in Sec. 4.2.2, Constraints that generally apply to most Voting Methods, the following constraints are most common for Decision Rules and Decision Rule Sets. Additionally, Sec. 4.2.4, Ties and Tiebreaking and Sec. 4.2.5, Interrelated Contests or Connected Contests may apply for some voting method variants. All outcomes of contests produced by their Decision Rules and Decision Rule Sets are subject to the determination of elections officials and certifying authorities. **Selection Criteria:**

Plain Language:
The count of valid selections for a single contest on a single ballot must be equal to the minimum count of selections required for a valid cast vote and less than or equal to the maximum count of selections allowed for a valid cast vote for the contest for one ballot.

Mathematical Logic DSL:

∀ contest (castVote) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1) }

Where N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods

Decision Rule Set:
The decision rule set is the decision rule set of the voting method variant.

Determine Outcome Rule:

Plain Language:
In a single-winner contest, if one contest option meets the threshold criterion of the voting method if a threshold exists, and receives more votes than any other contest option, then it prevails (wins).

Mathematical Logic DSL:
For the case where no threshold is specified:

∀ singleWinnerContest (win) ⇔
{ (∃ contestOption i ∈ (contestOptionVoteCount i = max (contestOptionVoteCount j ∈)))
∧ ∀ k ∈ (contestOptionVoteCount i > contestOptionVoteCount k | k ≠ i) }

For the case where a threshold is specified:

∀ singleWinnerContest (win) ⇔
{ (∃ threshold)
∧ (∃ contestOption i ∈ [1 to n] (contestOptionVoteCount i = max (contestOptionVoteCount j ∈ [1 to n])) ∧ (contestOptionVoteCount i ≥ threshold))

	∧ ∀ k [1 to n] (contestOptionVoteCount i > contestOptionVoteCount k | k ≠ i) } Where N is the number of selections allowed to be allocated by the voter in the contest, j and k are distinct indexes ranging over all contest options, and i is the distinct index of the contest option with the maximum vote count. **Plain Language:** In a multi-winner contest, with n winners, up to N contest options receiving the most votes prevail (win). Any remaining winners must be determined using other (e.g., tie breaking) methods. **Mathematical Logic DSL:** ∀ multiWinnerContest (win) ⇔ {(∃ max (contestOptionVoteCount ∈[j])) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: $(n_j, n_{j+1}, \ldots, n_n)$.
c.	**Tabulation Methods Process Set** N/A
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** NIST 1500-107, Voting Method and Tabulation Universe, Voting Methods Baseline and Variations Resources tab **Single-winner majority requirement:** S.C. Code Ann. § 5-15-62(a)(1) Ga. Code Ann. § 21-2-501(a) **Single-winner plurality requirement:** Ga. Code Ann. § 21-2-501(b) **Multi-winner majority requirement:** S.C. Code Ann. § 5-15-62(a)(2)
f.	**Test Conditions & Expected Outcomes For This Variant** N/A

4.2.4 Ties and Tiebreaking

4.2.4	**Ties and Tiebreaking**

a.	**Description** A tie occurs when two or more contest options have the same number of votes. Typically, tiebreaking, as a result of detecting a tie, is only necessary if the tie impacts the outcome.
b.	**Constraints** A tied contest is true if the count of votes for potential winners of contest options are equal. A tiebreaker may be needed if the tie affects the outcome. Typically, tiebreaking does not apply to measures. In an RCV contest, the comparison to determine if a tie has occurred can either be applied at the contest outcome level or between contest options to be eliminated in a round.. See Sec. 4.2.12.1, Requirements that generally apply to most RCV Voting Methods, RCV Tiebreaking **Decision Rule Set:** **See Tiebreaking Mechanisms (below)** **Determine if Tied rule** **Plain Language:** A tie occurs when two or more distinct contest options have the same number of votes. **Mathematical logic DSL:** $\forall$ (tie) $\Leftrightarrow$ $\{(\exists$ contestOptionVoteCount$\{1\ldots j\}$ s.t. $\exists$ ((contestOptionVoteCount $\in[j_h]$) = (contestOptionVoteCount $\in[j_k]$))) $\wedge$ $(h \neq k)\}$ Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: $(n_j, n_{j+1}, \ldots, n_n)$. **Tie Impact** It is possible that a tie occurs and the contest outcome is impacted, or the contest outcome is not impacted. Examples include: ● Impact: ○ When there is a tie between the top two candidates in a single-winner contest ○ When a third and fourth candidate in vote order are tied in a three-seat contest ○ RCV-specific conditions: ● See Sec. 4.2.12.1, Requirements that generally apply to most RCV Voting Methods, RCV Tiebreaking for more discussion. ● No impact: ○ When a tie is detected but neither candidate in a single-winner contest have met

	the threshold requirement ○ When the second and third candidate in vote order are tied in a three-seat contest In general when a tie is detected for a measure contest, tiebreaking does not apply because the measure fails. **Tiebreaking Mechanisms** When a tie occurs that impacts the contest outcome determination, the mechanism for breaking ties is a manual process that most commonly involves chance. Examples of tiebreaking methods that have been used are as follows: ● Flipping a coin ● Drawing straws ● Rolling of dice either physically or digitally ● Picking the high card from a deck of shuffled cards ● High/low number in random number generation (Pseudo-Random) In some cases, rules may require a new election.
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative references** VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Rules Tab, Determine if Tied **Resolve ties by lot:** Mich. Comp. Laws § 168.851 N.C. Gen. Stat. § 163-182.8(2) **Resolve by elected body:** Cal. Elec. Code Div. 15, Ch. 10, § 15654 **Resolve by runoff:** N.C. Gen. Stat. § 162-182.8(1)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.2, Expected Outcomes and Sample Test Sets, Baseline Contest

4.2.5 Interrelated Contests or Connected Contests

4.2.5	**Determine Interrelated or Connected Contests**
a.	**Description** Interrelated Contests have a relationship between contests. Precisely, the relationship between the contests and selections of the voter determines some aspect of the process flow of tabulation or the interpretation of selections or filtering of allowable contests in which

	selections may be made. The cases indicate a particular one of a set of branches in the process flow dependent upon selections made by the voter or dependent on the aggregate outcome of voter selections. The precise details vary by the type of contest. In the category of interrelated contests, the relationships may be one-to-one (one contest is always controlling, and it controls one related contest), one-to-many (one contest is always the controlling contest and it controls one or more contests), or many-to-many (the bidirectional relationship case). **Controlling Contest / Relationship**: A controlling contest is a contest on a ballot where a contest choice affects choices that can be made in other contests on the ballot. It may automatically make selections of contest choices in other contests, restrict contest choices in other contests on the ballot, or automatically deselect contest choices in other contests on the ballot. **Controlled Contest / Relationship**: A contest on a ballot where a contest choice in another contest controls the validity of choice selections that are made in it. **Filtering Relationship**: A filtering relationship is one where a contest selection enables or blocks selections in the controlled contest. For example, a Party Preference selection in an Open Primary filters out any selections in a different party's contest. A different example which voids the party primary ballot is when no Party Preference selection is selected and a voter selects candidates from more than one political party **Selecting Relationship**: Choosing one option selects a group of contest options, sometimes in more than one contest. For example, a Straight Party selection automatically selects every candidate of that party in all connected contests on the ballot.
b.	**Constraints** The decision rule set to determine elected is a controlling decision rule set of the voting method decision rule subsets that are linked to it. **Counting Rules** The counting rules are those of the applicable voting method. N-of-M (where the case is 1 of M) counting rules apply. Overvote is > 1 selection, undervote if none.
c.	**Tabulation Methods Process Set** N-of-M
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios Typically the same as for N-of-M The primary tabulation scenarios are Precinct, Central, Distributed, and Batch.

e.	**Legislation Examples of Jurisdictions Using Interrelated Contests** **[examples of the types of contest already in Universe and section 3]** NIST 1500-107, Voting Method and Tabulation Universe, Baseline & Variations Resources, Ln. 30 **Controlling contest:** Ala. Code §§ 45-4-111(d)(1)(d), 17-7-21(b)(3) Okla. Stat. tit. 26, § 7-119 S.C. Code Ann. § 7-13-330 Mich. Comp. Laws § 168.736c Ind. Code § 3-11-14-23(c) **Controlled contest:** S.C. Code Ann. § 7-13-330 Mich. Comp. Laws § 168.736c **Filtering relationship:** Wis. Code 5.91(6) (party selection on open primary ballot) **Selecting relationship:** Ala. Code §§ 45-4-111(d)(1)(d), 17-7-21(3) Okla. Stat. tit. 26, § 7-119 S.C. Code Ann. § 7-13-330 Tex. Elec. Code Ann. § 64.004 Utah Code Ann. § 20A-3-106 Mich. Comp. Laws § 168.736c Ind. Code § 3-11-14-23(c)
f.	**Test Conditions & Expected Outcomes For This Variant** Typically the same as N-Of-M

4.2.6 N-of-M (Baseline) Contests

4.2.6.1 Plurality

4.2.6.1.1 Determine Outcome: Plurality, Candidates, Single Seat

4.2.6.1.1	**Determine Outcome: Plurality, Candidates, Single Seat**
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected.
b.	**Constraints**

See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Selection Criteria

Plain Language:
At most one candidate selection is allowed per ballot for the contest.

Mathematical Logic DSL:

∀ singleWinnerContest (selectionValid) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (N = 1)}

Where N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language:
The prevailing contest option is the contest option with the maximum vote count.

Mathematical Logic DSL:

∀ singleWinnerContest (win) ⇔
{ (∃ max (contestOptionVoteCount ∈[j])) }

For the case where a threshold is specified:

∀ singleWinnerContest (win) ⇔
{ (∃ threshold)
∧ (∃ max (contestOptionVoteCount ∈[j]))
∧ (∃ max (contestOptionVoteCount ∈[j])) >= threshold)}

	Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n:.(n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine if Elected, Plurality, Candidates, Single Seat N.C. Gen. Stat. § 163-292
f.	**Test Conditions & Expected Outcomes For This Variant** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes, Baseline Contest**

4.2.6.1.2 Determine Outcome: Plurality, Candidates, Multiple Seats

4.2.6.1.2	**Determine Outcome: Plurality, Candidates, Multiple Seats**
a.	**Description** The generic high-level decision rule specifies that, for a multi-winner contest, the candidates with the largest count of votes are the prevailing candidates presented to election officials to be certified as elected.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The voter is allowed to vote for as many candidates as there are seats to fill. **Mathematical logic DSL:** { ∀ contest (selectionValid) ⇔ (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) } Where N is the number of selections allowed to be allocated by the voter in the contest.

	Counting Rules: See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** In a multi-winner contest, with N winners, up to N contest options receiving the most votes prevail (win). Any remaining winners must be determined using other (e.g., tie-breaking) methods. **Mathematical logic DSL:** ∀ multiWinnerContest (win) ⇔ { ∃ max (contestOptionVoteCount ∈[j]))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: $(n_j, n_{j+1}, \ldots, n_n)$.
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine if Elected, Plurality Ballot Measure N.C. Gen. Stat. § 163-182.15(d)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.2, Expected Outcomes and Sample Test Sets, Baseline Contest

4.2.6.2 Majority

4.2.6.2.1 Determine Outcome: Majority, Candidates, Single Seat

4.2.6.2.1	**Determine Outcome: Majority, Candidates, Single Seat**

a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with more than half the thresholdBase thresholdBase, typically the number of valid votes cast, is the prevailing candidate presented to election officials to be certified as elected. If no candidate is elected, jurisdiction rules determine the next steps, e.g., top-two runoff election.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** At least one and at most one candidate selection is allowed by each voter for the contest. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = 1) ∧ (MinContestSelectionsAllowed = 1) } Where N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing candidate is the one with more than a threshold of fifty percent (> 50%) of the thresholdBase, and thresholdBase is typically the count of votes cast but may be the count of ballots cast. **Mathematical Logic DSL:**

	∀ singleWinnerContest (outcomeDetermined ⇔ { [(∃ (threshold) ∧ (threshold = (thresholdBase * 50%)) ∧ (∃ max (contestOptionVoteCount ∈[j])) ∧ (∃ max (contestOptionVoteCount ∈[j])) > threshold)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine Outcome, Majority S.C. Code Ann. § 5-15-62
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.2, Expected Outcomes and Sample Test Sets, Baseline Contest

4.2.6.2.2 Determine Outcome: Majority, Measure

4.2.6.2.2	**Determine Outcome: Majority, Measure**
a.	**Description** A standard ballot measure with only Yes/For and No/Against selection options is presented to election officials as passed if the count of all Yes votes is greater than the count of all No votes, and is presented to election officials as tied if the count of all Yes votes is equal to the count of all No votes.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** At most one selection of Yes (Y) or No (N) is allowed by each voter for the measure contest. **Mathematical logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)

	∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) } Where N is the number of selections allowed to be allocated by the voter in the contest. Note the two allowable options may be labeled in a variety of ways on a ballot, for example, 'for' or 'against'. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the one with more than fifty percent (> 50%) of the thresholdBase for the contest. In the event of a tie, the measure fails. **Mathematical Logic DSL:** ∀ singleWinnerContest (outcomeDetermined) ⇔ { (∃ (threshold) ∧ (∃ max (contestOptionVoteCount ∈[j])) ∧ (∃ max (contestOptionVoteCount ∈[j])) > threshold)} ∧ (threshold = (thresholdBase * 50%)) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: $((n_j, n_{j+1}, \ldots, n_n)$. Note the two allowable options may be labeled in a variety of ways on a ballot, for example, 'for' or 'against'.
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine if Passed, Plurality Ballot Measure N.C. Gen. Stat. § 163-182.15(d)

f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.2, Expected Outcomes and Sample Test Sets, Baseline Contest

4.2.6.3 Super-Majority

4.2.6.3.1 Determine Outcome: Super-Majority, Measure

4.2.6.3.1	Determine Outcome: Super-Majority, Measure
a.	**Description** A super-majority voting method may be any voting method where the requirement to be passed is greater than a simple majority of > 50%. In contrast to a simple majority requiring a threshold defined as > 50% of the thresholdBase, a specified super-majority, for example, may be greater than ⅔ or ¾ threshold.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** At most one selection of Yes (Y) or No (N) is allowed by each voter for the measure contest. **Mathematical logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount = 1) } **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods Overvote - See Sec. 4.2.2, Constraints that generally apply to most Voting Methods. Undervote - See Sec. 4.2.2, Constraints that generally apply to most Voting Methods. **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

	Plain Language: The prevailing outcome is the contestOption which has the maximum contestOptionVoteCount that exceeds the specified super-majority threshold defined in terms of the specified threshold base. The threshold is greater than a specified percentage, and the threshold base is typically defined as either the count of votes cast or the count of ballots cast. **Mathematical Logic DSL:** ∀ singleWinnerContest (outcomeDetermined ⇔ { (∃ (threshold) ∧ (∃ max (contestOptionVoteCount ∈[j])) ∧ (∃ max (contestOptionVoteCount ∈[j])) > threshold) ∧ (threshold > 50%)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n). Note the two allowable options may be labeled in a variety of ways on a ballot, for example, 'for' or 'against'.
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Supermajority Fl. Const. art. XI sec. 7 City of Fargo, N.D., Charter art. 11 amend. 1
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.2, Expected Outcomes and Sample Test Sets, Baseline Contest

4.2.6.4 Approval Voting

4.2.6.4.1 Determine Outcome: Approval Voting, Candidates, Single Seat

4.2.6.4.1	**Determine Outcome: Approval Voting, Candidates, Single Seat**

a.	**Description** Approval Voting uses an N (number of selections allowed)-of-M (number of choices available) contest with N equal to M. The voter is allowed to select all candidates that they "approve" of including all candidates. There is no special setup for defining the contest other than setting the allowed votes to equal the number of candidates. The candidate elected to fill the available seat is the one with the most votes (most approved). To measure the approval percentage for each candidate, divide the number of votes the candidates received by the number of ballots containing the relevant contest that were cast in the election.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. An exception is that a voter may select 0, 1, 2, 3, … , or up to M candidates from a list of M contest choices in the contest without invalidating the ballot for that contest, and all valid selections are counted in the final tally. Approval Voting only applies when there are 3 or more candidates. **Plain Language:** The voter is allowed to vote for as many contest options as are on the ballot. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = M) } Where M is the number of contest options in the list available to the voter, and N is the number of selections allowed to be allocated by the voter in the contest **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods. **Overvote** **Plain Language:** It is not possible to overvote the contest since every contest option may be selected and

	counted. The number of overvotes on the tabulation counter always equals zero in Approval Voting. **Mathematical Logic DSL:** ∀ contest (∄ overVote) { ∧ (overVoteCounter = 0) } **Undervote** **Plain Language:** A true undervote occurs only if there are no selections. For ballot accounting purposes, N-of-M tabulation will report the number of unselected contest options as undervotes. **Mathematical Logic DSL:** ∀ contest (underVote)⇔ {(selectedBallotContestOptionsCount < MinContestSelectionsAllowed)} **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The candidate elected to fill the available seat is the one with the most votes (most approved). **Mathematical Logic DSL:** ∀ singleWinnerContest (win) ⇔ { (∃ max (contestOptionVoteCount ∈[j])) } Where j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios Approval Voting is compatible with all N-of-M tabulation scenarios.
e.	**Authoritative References** 2018 Fargo, ND, Measure 1 adopted for use beginning in 2020 See VMWG NIST 1500-107-Voting Methods Universe, Approval Voting

	City of Fargo, N.D., Charter art. 11
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.3, Expected Outcomes and Sample Test Sets, Approval Voting

4.2.6.4.2 Determine Outcome: Approval Voting, Candidates, Multiple Seats

4.2.6.4.2	**Determine Outcome: Approval Voting, Candidates, Multiple Seats**
a.	**Description** Approval Voting uses an N (number of selections allowed)-of-M (number of choices available) contest with N equal to M. The voter is allowed to select all candidates that they "approve" of including all candidates. There is no special setup for defining the contest other than setting the allowed votes to equal the number of candidates. The candidates elected to fill the available seats are the ones with the most votes (most approved). To measure the approval percentage for each candidate, divide the number of votes the candidates received by the number of ballots containing the relevant contest that were cast in the election.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. An exception is that a voter may select 0, 1, 2, 3, … , or up to M candidates from a list of M contest choices in the contest without invalidating the ballot for that contest, and all valid selections are counted in the final tally. **Plain Language:** The voter is allowed to vote for as many contest options as are on the ballot. **Mathematical Logic DSL:** ∀ multiWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = M) } Where M is the number of contest options in the list available to the voter and N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:

Overvote - See Sec. 4.2.2, Constraints that generally apply to most Voting Methods.

Plain Language:
It is not possible to overvote the contest since every contest option may be selected and counted. The number of overvotes always equals zero in Approval Voting.

Mathematical Logic DSL:

$\forall$ contest ($\nexists$ overVote)
{ $\wedge$ (overVoteCounter = 0) }

Undervote - See Sec. 4.2.2, Constraints that generally apply to most Voting Methods.

Plain Language:
Undervotes have no significance since the number of votes allowed do not relate to the number of seats to be elected. For ballot accounting purposes, N-of-M tabulation will report the number of unselected contest options as undervotes.

Mathematical Logic DSL:

$\forall$ contest (underVote)
$\Leftrightarrow$ (selectedBallotContestOptionsCount < MaxContestSelectionsAllowed))

Decision Rule Set:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language:
The candidates elected to fill the number of available seats are the ones with the most votes (most approved).

Mathematical Logic DSL:

$\forall$ multiWinnerContest (win) $\Leftrightarrow$
{($\exists$ contestOptionVoteCount$_{\{1\ldots n\}}$
$\wedge$ ($\exists$ max(contestOptionVoteCount $\in$[j]))}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n:.(n_j, n_{j+1}, … , n_n).

c.	**Tabulation Methods Process Set** See Sec. 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios Approval Voting is compatible with all N-of-M tabulation scenarios.
e.	**Authoritative References** 2018 Fargo, ND, Measure 1 in use since 2020 2020 St Louis, MO, Prop D in use since 2021 See VMWG NIST 1500-107-Voting Methods Universe, Approval Voting City of Fargo, N.D., Charter art. 11 City of St. Louis, MO., Ordinances § 2.08.460
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.3, Test Conditions and Expected Outcomes, Approval Voting

4.2.7 Straight Party

4.2.7.1 Determine Party Indication, Straight Party

4.2.7.1	**Determine Party Indication, Straight Party**
a.	**Description** A party selection defines the set of candidates that the voter selects by default in the connected contests, unless overridden by a voter selection within the connected contest.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods See Sec. 4.2.5. Interrelated Contests or Connected Contests **Plain Language:** Only a valid straight party selection has an effect on the connected contests. (The straight-party selection casts a vote for all candidates affiliated with the selected party in the connected partisan contests unless overridden by a manual selection in one or more specific contests.) **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >=MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = 1) }

	Where N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** **Overvote** **Plain Language:** Overvote is True if and only if the voter selects more contest options than allowed. If more than one party is selected, then the individual connected/controlled contests must be voted independently of party indication. **Mathematical Logic DSL:** { ∀ contest (overVote) ⇔ (selectedBallotContestOptionsCount > MaxContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = 1)) } **Undervote** **Plain Language:** Undervote is true if and only if no party selection is made. If there is no selection of party, then no impact on connected/controlled contests. **Mathematical logic DSL:** { ∀ contest (underVote) ⇔ (selectedBallotContestOptionsCount < MinContestSelectionsAllowed) ∧ (MinContestSelectionsAllowed = 1)) } **Decision Rule Set:** For this variant, the decision rule set of the connected contests is the decision rule set that corresponds to the voting method of the controlled contest(s). **Determine Outcome Rule:** The result is not the outcome of a contest; it is reporting the count of valid selections of the straight party indication for individual ballots.
	Tabulation Methods Process Set See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios

	The primary tabulation scenarios are Precinct, Central, Distributed, and Batch.
e.	**Authoritative References** See VMWG NIST 1500-107-Voting Methods Universe, Rules Tab, Single Round Tabulation, Straight Party Voting Ala. Code §§ 45-4-111(d)(1)(d), 17-7-21(3) Okla. Stat. tit. 26, § 7-119 S.C. Code Ann. § 7-13-330 Tex. Elec. Code Ann. § 64.004 Utah Code Ann. § 20A-3-106 Mich. Comp. Laws § 168.736c Ind. Code § 3-11-14-23(c) **Overvoted straight party selection:** Ind. Code § 3-12-1-7(f)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.3, Expected Outcomes and Sample Test Sets, Straight Party Voting

4.2.7.2 Determine Effect of Straight Party Indication on Connected Contests

4.2.7.2	**Determine Effect of Straight Party Indication on Connected Contests**
a.	**Description** A party selection in the controlling Straight Party (SP) contest indicates the potential selection of the candidates for the selected party in all of the connected contests. However, direct selections in the contest may void these selections. With Exclusive rules, any contest selection nullifies a straight party selection. With Inclusive rules, any straight party selections are initially combined with direct selections. If an overvote exists, then the straight party selections are nullified and the direct selections are processed. Even though there is no explicit distinction between single and multi seat, it really only manifests itself in the multi-seat situation. A particular example of the Inclusive rule is a cumulative voting contest in conjunction with straight party selection (Alabama). A straight party selection only selects 1 vote for each candidate associated with the selected party. It is then tabulated like a multi-seat contest using Inclusive rules. This allows a candidate to get more than 1 vote, from a direct selection and a straight party selection, as long as the combination doesn't exceed the allowed selections.
b.	**Constraints** See Sec. 4.2.5. Interrelated Contests or Connected Contests

	Plain Language: **Exclusive Rule:** Any selection in the individual connected contest overrides the straight party selection(s). **Inclusive Rule:** In a single-seat contest, any selection in the individual connected contest overrides the straight party selection. In multi-seat contests, if the straight party selections in combination with the direct contest selection(s) do not result in an overvote in the contest, then the straight party selections are also counted. **Counting Rules** The counting rules are not the outcome of a contest, the connected contest counting rules are those applicable for the connected contest voting method. The counting rules in this case are for the purpose of determining the impact of valid selections of the straight party indication for individual ballots on the connected contest. A valid selection in a Straight Party creates an indirect selection for every candidate affiliated with the selected party in every contest linked to the straight party. If there is no direct selection in a contest linked to the straight party, the contest is tabulated and votes accumulated as if the indirect selections as if they were directly selected. If there are any direct selections in a single seat contest or a multi-seat contest that is tabulated using Exclusive rules, the contest is tabulated using its standard rules and indirect selections are ignored. When a multi-seat contest is tabulated using Inclusive rules all direct selection are processed in tabulation and any indirect selections are only included if they are for candidates other than the ones directly selected and only of any overvote is not created by including them. As with Exclusive rules, included indirect selections are tabulated as if they were directly selected. **Decision Rule Set:** The decision rules are not the decision rules for the outcome of a contest, the connected contest counting rules are those applicable for the connected contest voting method. **Plain Language:** 1. If there is no valid selection in the straight party contest: All connected contests are tabulated according to contest rules. 2. If there is a valid selection in the straight party contest and a connected contest has no candidate affiliated with the selected party: That connected contest is tabulated according to contest rules.

	3. If there is a valid selection in the straight party contest and a connected contest does have one or more candidates affiliated with the selected party and there are no direct candidate selections in the contest: That contest is processed according to contest rules using the straight party selected candidates as the selected candidates. 4. If there is a valid selection in the straight party contest and a connected contest does have one or more candidates affiliated with the selected party and there is one or more directly selected candidates in that contest: All straight party selected candidates are ignored and that contest is tabulated according to contest rules IF Exclusive straight party rules are being used OR if that contest is for a single seat and Inclusive straight party rules are used. 5. If there is a valid selection in the straight party contest and Inclusive straight party rules are being used and a connected contest is a multi-seat contest and that contest has one or more candidates affiliated with the selected party and there is one or more directly selected candidates: If the number of candidates selected whether directly or by the straight party is fewer or equal to the number of allowed selections, that contest is processed according to contest rules using all selected candidates. If the number of candidates selected whether directly or by the straight party selection exceeds the allowed number of selections, all straight party selections are ignored and that contest is processed according to contest rules using only direct selections. **Determine Outcome Rule:** The decision rule for determining a contest outcome corresponds to the voting method of the controlled contest(s).
c.	**Tabulation Methods Process Set** See Sec 3.2.3, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios The primary tabulation scenarios are Precinct, Central, Distributed, and Batch.
e.	**Authoritative References** VMWG NIST 1500-107-Voting Methods Universe, Rules Tab, Single Round Tabulation, Straight Party Voting **Effect of straight party selection on votes cast:** Ala. Code §§ 45-4-111(d)(1)(d), 17-7-21(3) Okla. Stat. tit. 26, § 7-119

	S.C. Code Ann. § 7-13-330 Tex. Elec. Code Ann. § 64.004 Utah Code Ann. § 20A-3-106 Mich. Comp. Laws § 168.736c Ind. Code § 3-11-14-23(c) **Split ticket (single seat):** Ala. Code § 17-6-36 Okla. Stat. tit. 26, § 7-127(5), 25 Pa. Cons. Stat. § 1003 S.C. Code Ann. §§ 17-13-330, 17-13-1340(b) Tex. Elec. Code Ann. § 65.007 Utah Code Ann. § 20A-3-106 Mich. Comp. Laws § 168.736c Ind. Code § 3-12-1-7(b)(3)(B) **Split ticket (multi-seat inclusive):** Ala. Code § 17-6-37 Okla. Stat. tit. 26 § 7-127 Tex. Elec. Code Ann. § 65.007 Utah Code Ann. § 20A-3-106(1)(?) Mich. Comp. Laws § 168.736c **Split ticket (multi-seat exclusive):** Ind. Code §§ 3-12-1-7(d)–(e) **Excluding multi-seat contests:** Ind. Code § 3-11-2-10(c)(2) **Independent ticket limitation:** Ind. Code § 3-11-2-10(d)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.3 , Test Conditions and Expected Outcomes, Straight Party Contest

4.2.8 Open Primary

4.2.8.1 Determine Open Primary

4.2.8.1	**Determine Open Primary**
a.	**Description** An Open Primary election is one that contains all partisan contests from each party that is holding a Primary election. The ballot may also contain nonpartisan contests (election of candidates that are running for nonpartisan contests) as well as measures. This discussion relates to how this ballot is tabulated and is independent of the rules for when this kind of

	ballot is issued. A voter is only allowed to vote in the partisan contests for one party.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. The voting method's constraints apply to the interrelated contests: See Sec. 4.2.5 Interrelated Contests or Connected Contests While a voter's ballot contains the contests of all contesting parties, a voter is only allowed to vote for candidates of contests affiliated with one party. For tabulation of both contests and party statistics, the ballot automatically becomes affiliated with the party that is chosen based on the party affiliation of the one or more contests in which the voter has made candidate selections. If the voter selects candidates of more than one political party, even if the selection is invalid, the entire partisan section of the ballot is voided. The ballot is considered a nonpartisan ballot for statistical reporting purposes. Only nonpartisan contests and referenda, if any, will be tabulated. In this event, a Party Crossover statistic may also be incremented for identification of this exception. If a voter does not make any candidate selection in a partisan contest, the ballot is considered a nonpartisan ballot for statistical reporting purposes, even if it does not contain any nonpartisan contests. In the latter case, it will also be considered a blank ballot. If a ballot is partisan, as defined above, all contests affiliated with that party (and any nonpartisan contests) will be tabulated using the voting method's rules. Contests affiliated with other parties will not be tabulated. Some states add a Party Preference Contest to the top of the ballot. A valid party selection in this contest causes any option selection in a contest of a party other than the selected one to be ignored. A valid Party Preference Contest selection protects the voter from inadvertently spoiling the partisan section of the ballot. Contest selections, including undervote and overvotes, are tabulated according to the rules of the contest's voting method for the contests that are affiliated with the party associated with the voted ballot and any Nonpartisan contests. All other party contests are ignored. See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The voter is allowed to vote in at most one party's contests.

	Mathematical Logic DSL: ∀ contest (castVote) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSetSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSetSelectionsAllowed) ∧ (MaxContestSetSelectionsAllowed = 1) ∧ (MinContestSetSelectionsAllowed = 1)} **Counting Rules:** The counting rule of the connected contests are those of the voting methods of the connected contests. [corresponds to the voting method of the contest(s] [If the ballot is cross voted then all are ignored] **Decision Rule Set:** For this variant, the decision rule set of the connected, partisan, contests is the decision rule set that corresponds to the voting method of the contest(s). **Determine Outcome Rule:** For this variant, the decision outcome rule of the connected, partisan, contests is the decision outcome rule that corresponds to the voting method of the contest(s).
c.	**Tabulation Methods Process Set** Each individual interrelated contest has a tabulation process set that is defined by its voting method.
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** Wis. Stat. § 5.91(6) - contains Party Preference Minn. Stat. §§ 204D.05, 204D.08, 8220.0750(J) - does not contain Party Preference VMWG NIST 1500-107-Voting Methods Universe, Links to Primary Election Resources Tab
f.	**Test Conditions & Expected Outcomes For This Variant** See **Sec.** 3.2.22.4, Expected Outcomes and Sample Test Sets, Open Primary Contest

4.2.8.1.1 Determine Open Primary Party Preference, Controller

4.2.8.1.1	**Determine Open Primary Party Preference, Controller**
a.	**Description**

	Open Primaries (those where a single ballot contains contests from all eligible parties) with a Party Preference controller question allow the voter to select the party whose ballot they intend to vote in the privacy of the voting booth. When the voter selects a party preference option, any votes cast for candidates of a different party are ignored. This voting method does not have any distinct set of counting rules, tabulation methods or decision rule sets independent of the connected partisan contests decision rule sets.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **Plain Language:** A voter is allowed to select at most one party preference. **Mathematical Logic DSL:** ∀ contest (castVote) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = 1) ∧ (MinContestSelectionsAllowed = 1) } **Counting Rules:** Each individual interrelated contest has a tabulation process set that is equal to the tabulation process set for its voting method. See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Decision Rule Set:** The decision rule set is the decision rule set for each individual contest that is qualified given application of the Interrelated Contests filter and the contest's voting method, as described in that module. **Determine Outcome Rule:** See the outcome rule for each individual contest that is qualified given application of the Interrelated Contests filter and the contest's voting method, as described in that module: See Sec. 4.2.5. Interrelated Contests or Connected Contests
c.	**Tabulation Methods Process Set** See Sec 3.2.11, Open Primary Party Preference Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios

	The primary tabulation scenarios are Precinct, Central, Distributed, and Batch.
e.	**Authoritative References** Wis. Stat. § 5.91(6) - does contain Party Preference VMWG NIST 1500-107-Voting Methods Universe, Links to Primary Election Resources Tab
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.4, Expected Outcomes and Sample Test Sets, Open Primary Contest

4.2.8.1.2 Determine Open Primary Partisan Contest, Controlled

4.2.8.1.2	**Determine Open Primary Partisan Contest, Controlled**
a.	**Description** All Open Primary partisan contests are controlled. If the ballot contains a Party Preference contest and the voter marks one and only one party selection, then only contests of the selected party will be tabulated. The Party Preference is the controlling contest described immediately above in Sec. 4.2.8.1.1. All partisan contests associated with any other party will be ignored. If the ballot does not have a Party Preference contest or there is a Party Preference contest without a valid selection, the following rule applies: If the ballot contains selections in contests associated with more than one party, no partisan contests will be tabulated or included on election results reports.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **Plain Language:** A voter may only vote in the contests of a single political party. **Mathematical Logic DSL:** ∀ contest (castVote) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSetSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSetSelectionsAllowed) ∧ (MaxContestSetSelectionsAllowed = 1) ∧ (MinContestSetSelectionsAllowed = 1)} **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods All partisan contests of a single designated party will be tabulated if:

	1. there is a selection in at least one contest of that party and no selection in any other partisan contests. 2. there is a party preference contest on the ballot with a single selection for the designated party If either of the two conditions are satisfied, the ballot will be tabulated for the designated party. If neither of the two conditions exist, the ballot will be tabulated as a nonpartisan ballot and all partisan contests will be ignored and not tabulated. If case 2 is satisfied, all partisan contests other than the designated or chosen party contests will be ignored and not tabulated. If the Open Primary ballot does not contain a Party Preference contest or if there is one but it does not contain a valid party selection, partisan contests are only tabulated if there is at least one candidate selection in one partisan contest and there is no selection and any partisan contest associated with another party. If this condition is met, or if the ballot contains a Party Preference contest with a valid party selection, the ballot is counted for that party and all partisan contests associated with that party, as well as nonpartisan contests, are tabulated using the tabulation rules associated with each contest. If there are any selections made in contests of more than one party, valid or invalid, the ballot is tabulated as nonpartisan, and only nonpartisan contests are tabulated. If the ballot contains a Party Preference contest with a valid party selection, only partisan contests of that party and nonpartisan contests are tabulated and contests of any other party are ignored whether or not they contain a selection. Each individual interrelated contest is tabulated according to the contest's tabulation process. See Sec 4.2.6, N-of-M (Baseline) Contests **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods The decision rule set is the decision rule set for each individual contest's voting method that is qualified given application of the Interrelated Contests filter, as described in that module. **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods**.** The determine outcome rules is the determine outcome rules for each individual contest that is qualified given application of the Interrelated Contests filter and the contest's voting method, as described in that module: See Sec. 4.2.5. Interrelated Contests or Connected Contests
c.	**Tabulation Methods Process Set** See Sec 3.2.12, Open Primary Partisan Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios

e.	**Authoritative References** VMWG NIST 1500-107-Voting Methods Universe, Links to Primary Election Resources Tab Wis. Stat. § 5.91(6) - does contain Party Preference
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.4, Expected Outcomes and Sample Test Sets, Open Primary Contest

4.2.9 Cumulative Voting

4.2.9.1 Determine Outcome: Cumulative Voting, Candidates, Multiple Seats

4.2.9.1	**Determine Outcome: Cumulative Voting, Candidates, Multiple Seats**
a.	**Description** Cumulative Voting is used in a multi-seat contest. Voters have as many votes as there are seats to be elected, and can assign them in any variety of whole-vote allotments in any combination across the candidates in the race. The winners are the candidates with the most votes.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The voter is allowed to vote for 1 to N selections in the contest, where N is the number of candidates to be elected. Multiple selections for a candidate may be made as long as total selections in the contest doesn't exceed N. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsAllowedPerContestOption = N) } Where N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods Distinct to cumulative voting is that multiple selections are allowed for a contest option.

	Plain Language: As long as no more than N contest options selections are made in total, the number of selections made for each option will be added to the count for that contest option. **Mathematical Logic DSL:** ∀ contest (validBallotContestCount) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (contestCounterInitialValue = 0) ∧ (contestCounterIncrement >= 1) ∧ (contestCounterIncrement <= N) ∧ (MaxSelectionsAllowedPerContestOption = N) ∧ (contestVoteCounterValue = (contestVoteCounterValue + contestCounterIncrement)) ∧ (contestVoteCounterValue <= ballotsCastCount) } Where N is the number of selections allowed to be allocated by the voter in the contest. **Overvote** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods **Undervote** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods
c.	**Tabulation Methods Process Set** See Sec. 3.2.3, Process N-of-M Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios The main tabulation scenarios are Precinct, Central, Distributed, and Batch.
e.	**Authoritative References** Port Chester, NY

	VMWG NIST 1500-107-VotingMethodAndTabulationUniverse, Rules, Cumulative Voting VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Links to Cumulative & Proportional Voting Resources Tab *United States v. Port Chester*, S.D.N.Y., Case No. 06 Civ. 15173, Consent Decree at 3.
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.5, Expected Outcomes and Sample Test Sets, Cumulative Contest

4.2.10 Proportional Voting

4.2.10.1 Determine Outcome: Proportional Voting, Candidates, Multiple Seats

4.2.10.1	**Determine Outcome, Proportional Voting, Candidates, Multiple Seats**
a.	**Description** Proportional Voting is used in a multi-seat contest but allows the voter to cast more than a single vote for a candidate by proportionally allotting the allowable votes to the number of candidates selected. As long as the number of selections doesn't exceed the maximum, the allowed number of votes is proportionally allotted to each selected candidate. The allocated vote value may be a fractional number if a voter makes fewer selections than the number allowed. Undervotes are not possible unless no selections are made. Overvotes occur if the voter makes more selections than allowed. Proportional Voting is also known as Equal and Even Cumulative Voting.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The voter is allowed to select up to N contest options where N is the number of candidates to elect. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsAllowedPerContestOption = 1) } Where N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language: The number of votes tabulated for each contest option depends on the number of contest options selected. Votes counted for each selection are the allowed number for the contest divided by the number of contest options selected.
The allocated vote value may be a fractional number if a voter makes fewer selections than the number allowed.

Mathematical Logic DSL:

∀ contest (validBallotContestCount) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestVoteSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (contestCounterInitialValue = 0)
∧ (contestCounterIncrement >= 1)
∧ (contestCounterIncrement <= N)
∧ (MaxSelectionsAllowedPerContestOption = N)
∧ (contestVoteCounterValue = (contestVoteCounterValue + contestCounterIncrement))
∧ (contestVoteCounterValue <= ballotsCastCount)
∧ (contestCounterIncrement = N/∑(validContestOptionSelections))}

Where N is the number of selections allowed to be allocated by the voter in the contest.

Overvote
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods

Undervote

Plain Language: An undervote in a proportional contest occurs only when no selections are made.

Mathematical Logic DSL:

∀ contest (underVote) ⇔
{ (selectedBallotContestOptionsCount = 0) }

	Decision Rule Set: See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The candidates elected to fill the number of available seats are the ones with the most votes. **Mathematical Logic DSL:** ∀ multiWinnerContest (win) ⇔ { (∃ max (contestOptionVoteCount ∈[j])) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: .(n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.4, Process N-of-M Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios The main tabulation scenarios are Precinct, Central, Distributed, and Batch.
e.	**Authoritative References** Peoria, IL VMWG NIST 1500-107-Voting Methods Universe, Rules, Proportional Voting *Banks v. Peoria*, C.D. Ill. No. 87-2371, Consent Decree Exh. B at 1–2.
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.6, Expected Outcomes and Sample Test Sets, Proportional Voting Contest

4.2.11 Recall Contest

4.2.11.1 Determine Outcome: Recall Controlling Contest

4.2.11.1	**Determine Outcome: Recall Controlling Contest**
a.	**Description** For a Recall Controlling contest, there is a contest pair, one controlling and the other controlled that are traditionally together called Recall, but this structure can be used in other situations. See also the Recall Controlled Contest section below. The controlling contest may use one of two rules to control the controlled contest.

	Rule 1: vote yes in the controlling contest means your vote in the controlled contest will count Rule 2: any vote in the controlling contest means your vote in the controlled contest will count The controlling contest is tabulated as an N-of-M contest. If the controlling contest is voted according to the applicable rule (1 or 2), the controlled contest will also be tabulated as an N-of-M contest. If not, the controlled contest is ignored for processing. The controlled contest is processed if the applicable rule is fulfilled. In traditional use, the controlling contest is in a referendum format. If the controlling contest outcome fails, then the controlled contest becomes moot.
b.	**Constraints:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **Plain Language:** At most one selection of Yes (Y) or No (N) is allowed by each voter for the controlling contest. **Mathematical logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ `(selectedBallotContestOptionsCount = 1) } Note the two allowable options may be labeled in a variety of ways on a ballot, for example, 'for' or 'against'. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods If the controlling contest outcome fails, then the controlled contest becomes moot.
c.	**Tabulation Methods Process Set** See Sec. 4.2.5. Interrelated Contests or Connected Contests

d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** VMWG NIST 1500-107-Voting Methods Universe, Baseline and Variations Resources, ln 30 Ca. Const. art. II, § 15 Ca. Elec. Code § 11383 (failure of recall controlling contest) Ca. Elec. Code § 11384 (success of recall controlling contest) Ca. Elec. Code § 11385 (outcome of recall controlled contest) Ca. Elec. Code §§ 11320–11327 (recall ballot design)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.8 Expected Outcomes and Sample Test Sets, Controlling and Controlled Contest Pair

4.2.11.2 Determine Outcome: Recall Controlled Contest

4.2.11.2	**Determine Outcome: Recall Controlled Contest**
a.	**Description** Recall elections are made up of a pair of contests, one controlling and the other controlled. See the Recall Controlling Contest section above for its description. If the controlling contest passes, then the controlled contest outcome is presented to election officials.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **Selection Criteria** **Plain Language:** Case 1: A valid yes vote in the controlling contest is required in order for this contest to be tabulated. **Mathematical Logic DSL:** ∀ contest (tabulateControlled) ⇔ {(∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >=MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1)

	∧ (N = 1) ∧ (contestOptionSelection = Y) } **Plain Language:** Case 2: A valid vote in the controlling contest is required in order for the controlled contest to be tabulated. **Mathematical Logic DSL:** ∀ contest (tabulateControlled) ⇔ {(∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >=MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = 1) ∧ (MinContestSelectionsAllowed = 1) } **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests The counting rules of the controlled contest are the counting rules that correspond to the voting method of the controlled contest. **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods The decision rule set of the controlled contest is the decision rule set that corresponds to the voting method of the controlled contest. **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods The determine outcome rule of the controlled contest is the determine outcome rules of the voting method of the controlled contest.
c.	**Tabulation Methods Process Set** See Sec. 3.2.4, Process N-of-M Baseline Contest and Sec. 4.2.5. Interrelated Contests or Connected Contests
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** VMWG NIST 1500-107-Voting Methods Universe, Baseline and Variations Resources, ln 30 Ca. Const. art. II, § 15

	Ca. Elec. Code § 11383 (failure of recall controlling contest) Ca. Elec. Code § 11384 (success of recall controlling contest) Ca. Elec. Code § 11385 (outcome of recall controlled contest) Ca. Elec. Code §§ 11320–11327 (recall ballot design)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.22.8 Expected Outcomes and Sample Test Sets, Controlling and Controlled Contest Pair

4.2.12 Ranked Choice Voting (RCV)

The RCV variants and counting rules described below are those currently used in governmental jurisdictions in the U.S. The primary categories of variants used include single-winner (IRV, Instant Runoff Voting) and multi-winner proportional (STV, Single Transferable Vote) ranked choice voting.

The counting rules include the varied treatment of overvotes, skipped votes and repeat ranking. These counting rules are used in both IRV and STV contests and may occur in different combinations. The combinations presented in the following modules represent the current (2024) combinations in use in U.S. jurisdictions. Each jurisdiction using RCV implements one counting rule for overvotes, one for skipped rankings and one for repeat rankings. In addition, some jurisdictions use single candidate elimination while others use batch elimination procedures. The choice does not affect the outcome but may influence the number of rounds of tabulation needed to determine the outcome. A further variant that applies in STV elections is the use of whole number or fractional thresholds. Currently no U.S. jurisdiction uses a fractional threshold calculation. Finally, and also applicable only in STV elections, is the choice between whole ballot or fractional surplus transfer. Most modern STV applications employ fractional surplus transfer while Cambridge, MA. still uses whole ballot surplus transfer.

The voting method modules below are organized by overvote rule, then by skipped ranking rule, then by whether jurisdictions use batch elimination, and then, for STV variants, jurisdictions are further organized by whether they use fractional or whole ballot transfer. This ordering is for display purposes only and is not a statement of preference or related to frequency of use.

4.2.12.1 High-Level Requirements Applicable to Most RCV Methods

4.2.12.1	High-Level Requirements Applicable to Most RCV Methods
a.	**Description** Ranked Choice Voting is a set of voting methods which allow each voter to rank contest options in order of the voter's preference, in which votes are counted in rounds using a series of runoff tabulations to defeat contest options with the fewest votes, and which determines a winner with a majority of final-round votes in a single-winner contest and provides proportional representation in multi-winner contests. The processing used in each IRV round, if there is no apparent winner, is as follows. The candidate with the lowest vote total in the round is considered eliminated. Each ballot containing the eliminated candidate as the current highest rank is processed to count for the next-highest-ranked continuing candidate (one that has not been eliminated). If there are no choices left on a given

	ballot or if a disqualifying ranking occurs prior to the next ranked continuing candidate, that ballot is considered exhausted and inactive for any subsequent rounds. The new round tabulates totals and determines whether any candidate now has sufficient votes to win. If not, the process is repeated round-by-round until a candidate wins. In STV elections, candidates who receive more votes than necessary to be elected – those candidates whose vote totals exceed the threshold of election – are deemed to have surplus votes. If a candidate has a surplus, the next step in counting is to transfer that surplus. Surplus votes may be transferred in various ways, but they follow the same logic as elimination transfers: each ballot counting for a candidate with surplus votes transfers a vote value to the next highest-ranked valid continuing candidate on the ballot. Depending on local rules, this vote value can go inactive if there is an overvote, skipped ranking, or repeat ranking. After transfer of a surplus, if there are more continuing candidates than seats to be filled, rounds continue with elimination of the candidate with the lowest vote total until all seats are filled.
b.	**Constraints** At most one selection may be made by the voter in each ranking position. The voter should not rank the same contest option in more than one ranking position. Some jurisdiction rules will ignore selection of a non-continuing candidate in a round containing multiple selections. The vote on the ballot may count toward a different selected candidate in any given round or, in the proportional multi-winner form of RCV, the vote may go toward multiple candidates in fractional parts summing to 1 single vote, i.e., a Single Transferable Vote (STV). The rounds of ballot counting can stop when the election contest tabulation scenario has reached the point in the process where it has resulted in a complete unofficial outcome for the election contest. For RCV (e.g., IRV or STV) the tabulation method produces "snapshots" for every round of a tabulation, and like the final output, the snapshots are stored in the NIST 1500-103 Cast Vote Record Common Data Format (CVR CDF). The full CVR Report (CVRR) final output includes all operations within the round. This requirement to store per-round snapshots and all final output in the CVR CDF does not preclude storing snapshots and all final output in additional formats. **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (N <= M) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1)

	∧ (MaxSelectionsAllowedPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods. See Sec. 4.2.5. Interrelated Contests or Connected Contests **RCV Tabulation Round Rules:** RCV tabulation proceeds in rounds. "Round of counting" or "round" means a step in the counting process during which selections for continuing candidates are tabulated for the purpose of determining whether a candidate has achieved a threshold of the votes cast for a particular office, whether any surplus needs to be transferred and how much, and, absent any candidate reaching the threshold, which candidate or candidates must be eliminated. VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Rounds See the specific Voting Method Variants in Sections 4.2.12.2-4.2.12.8 for application of each counting rule. In practice, one each of the counting rule options for overvotes, skipped ranking and repeat ranking must be applied to each RCV election. **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Undervote Rules Case:** If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)}

	Overvoted Ranking **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRankings (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted Ranking Rules Cases:** In case of an overvoted ranking, one of two unique decision alternatives, as specified by the jurisdiction: **Exhaust ballot XOR Advance to next ranking choice** **Plain Language:** **Exhaust Ballot** If an overvoted ranking occurs in the current ranking, then **exhaust ballot** XOR (exclusive OR) **Advance to next ranking** If an overvoted ranking occurs in the current ranking, then **advance to next ranking** **Mathematical Logic DSL:** ∀ contestRankings ∧ overVotedRanking) {(overVotedRanking => exhaustBallot) ⊻ (overVotedRanking => advanceToNextRanking)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRankings (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Cases:**

	In case of a skipped ranking, the cases are one of three mutually exclusive decision alternatives: The mutually exclusive constraint is denoted by the symbol: XOR, in the plain language. Singular Skip Exhaust ballot XOR Advance to next ranking choice XOR Consecutive Skips Exhaust ballot **Plain Language:** **Skipped Ranking Exhaust ballot** If no selection is made at this ranking, then **exhaust ballot** XOR (exclusive OR) **Skipped Ranking Advance to next ranking choice** If no selection is made at this ranking, then **advance to next ranking** XOR (exclusive OR) **Consecutive Skipped Rankings Exhaust ballot** If no selection is made at this ranking nor the previous ranking, then **exhaust ballot** **Mathematical Logic DSL:** ∀ (contestRankings ∧ skippedRanking) {(skippedRanking => exhaustBallot) ⊻ (skippedRanking => advanceToNextRanking) ⊻ (consecutiveSkippedRankings => exhaustBallot)} **Repeat Ranking** **Plain Language:** A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking. **Mathematical Logic DSL:** ∀ contestRankings (repeatRanking) ⇔ {((∃ rankingSelectionPerSpecificContestOption) ∧ (rankingSelectionsPerSpecificContestOptionCount >= 2))} **Repeat Ranking Rules Cases:** In case of a repeat ranking, the cases are one of two mutually exclusive decision alternatives: **Repeat Ranking Exhaust ballot** If a previously selected contest option is selected again, then the ballot is exhausted XOR (exclusive OR) **Repeat Ranking Advance to next ranking choice** If a previously selected contest option is selected again, then advance to the next ranking **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking)

	{((repeatRanking => exhaustBallot) ⊻ ((repeatRanking => advanceToNextRanking))} **Threshold** **Plain Language:** A "Threshold" sets the number of votes needed for a candidate to ensure the candidate wins a seat. The threshold is calculated based on the number of valid votes and the number of seats to fill. The standard threshold commonly used is the smallest number of votes such that the maximum number of candidates who can win a seat does not exceed the number of seats to fill. The general form is: threshold = (thresholdBase / (N + 1)) + [.0001..1] **Threshold Rule Cases:** In case of a threshold, the cases are one of three mutually exclusive decision alternatives: whole number dynamic, whole number static and fractional. **Whole Number Dynamic** In the case of whole-number dynamic, the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote. **Mathematical Logic DSL:** ∀ (contestRankings ∧ threshold) {(∃ threshold) ∧ (thresholdBaseDynamic = (castVoteCountCurrentRound)) ∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])} **Whole Number Static** In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote. Note: Most single-winner jurisdictions use a threshold based on the total votes counted in the current round, but multi-winner contests and a few single-winner jurisdictions, e.g. Minneapolis, use a threshold based on the total votes counted in the first round of the contest. **Mathematical Logic DSL:** ∀ (contestRankings ∧ threshold) {(∃ threshold) ∧ (thresholdBaseStatic = (castVoteCountFirstRound)) ∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Fractional Number (Static)
The threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill. When using a fractional threshold, the fraction is rounded up to the specified decimal place (typically 4). Applies to multi-winner RCV (STV) only.

Mathematical Logic DSL:

∀ (contestRankings ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = (thresholdBaseStatic / (seatsReqToFill + 1)) + .0001)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie, the cases are one of three mutually exclusive decision alternatives for tiebreaking in RCV.

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR (exclusive OR)
Tiebreaking between candidates who surpass threshold for multiseat contests
Tie may need to be resolved before surplus redistribution can occur depending on jurisdiction rules.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(surplusRedistributionRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs that impacts the contest outcome determination, the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Some specific RCV tiebreaking mechanisms include batch elimination and comparing results in the previous round.

Decision Rule Set:

Elimination of Candidate(s) within a Round:

Sequential Elimination:

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes.
Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ min(contestOptionVoteCount $\in$[j]))
$\wedge$ (potentialEliminateCandidateSet = min(contestOptionVoteCount $\in$[j]))}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).

Batch Elimination:

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes.
Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes.

This specification is only applicable in single-winner RCV. Note that multi-winner batch elimination has never been adopted in any jurisdiction in the United States, and is not known to be in consideration for adoption, and is therefore out of scope.

Mathematical Logic DSL:

$\forall$ contestRound (batchElimination) $\Leftrightarrow$
{(singleWinnerContest)

	∧ (seatsFilledCount = 0) ∧ (continuingCandidatesCount > 3) ∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2})) ∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2})) ∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)} **Transfer Surplus:** Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold. Transfer surplus is not applicable in IRV or Multi-Pass IRV or multi-seat Bottoms-Up RCV. **Mathematical Logic DSL:** ∀ contestRound (transferSurplus) ⇔ {(contestOptionVoteCount > threshold)} **Determine Outcome Rule:** VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine Outcome **Plain Language** If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold. **Mathematical Logic DSL:** ∀ contest (win) ⇔ { (∃ max (contestOptionVoteCount ∈[j])) ∧ ((contestOptionVoteCount >= threshold) ⊻ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1, Tabulation Scenarios, Central Tabulation See Sec. 4.1, Tabulation Scenarios, Multi-Contest-Multi Voting Method per Ballot

e.	**Authoritative References** See the specific Voting Method Variants for jurisdictions using the variant. **Round-by-round count:** Minneapolis Charter, 167.60(c) Minneapolis Charter, 167.70(c) **Batch elimination:** Minneapolis Charter, 167.20 **Links to Legislative Sources** Threshold Rule: VMWG-NIST-1500-107-Voting Methods and Tabulation Universe, Rules_Determine Threshold Elimination Rule: VMWG-NIST-1500-107-Voting Methods and Tabulation Universe, Rules_Elimination of Candidates **Tiebreaking:** **Decided by lot:** Me. Stat., tit. 21 § 723-A(3) Minneapolis Charter, 167.60(C)(1)(c)–(d) Portland, OR Code § 2.080.030(A)(6) **Check previous rounds:** Takoma Park Charter, § 606(h)
f.	**Test Conditions & Expected Outcomes For This Variant** See the specific Voting Method Variants for applicable Test Conditions & Expected Outcomes.

4.2.12.2 Determine Outcome: Colorado Local Option (IRV)

4.2.12.2	**Determine Outcome:** Colorado Local Option (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by CO: Basalt, Broomfield, Boulder MA: Easthampton

	It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: ballot is exhausted Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: single candidate
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:**

	∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Undervote Rules Case:** If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)} **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **OverVote rules case:** If an overvoted ranking occurs in the current ranking, then **exhaust ballot** **Mathematical Logic DSL:** ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => exhaustBallot)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)}

Skipped Ranking rules case:
If no selection is made at this ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRanking ∧ skippedRanking)
{(skippedRanking => exhaustBallot)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat ranking rules case:
If an overvoted ranking occurs in the current ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ contestRankings ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)
⊻ (overVotedRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)

	{(∃ threshold) ∧ (thresholdBaseDynamic = (castVoteCountCurrentRound)) ∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])} **Surplus Calculation Method** N/A **Candidate Elimination Method** Single Candidate **Plain Language:** If the seat is not filled, eliminate the candidate with the fewest votes. **Mathematical Logic DSL:** **Sequential Elimination:** ∀ contestRound (elimination) ⇔ {(seatsFilledCount < seatsReqToFillCount) ∧ (seatsToFillCount < continuingCandidatesCount) ∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$) ∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn) **Tiebreaking** **RCV Tiebreaking Cases:** In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking: **Tiebreaking between candidates being considered for elimination** Tie must be resolved between last place candidates in an elimination round. XOR **Tiebreaking between candidates for the (final) seats** Tie must be resolved to establish an outcome between tied candidates to be elected. **Mathematical Logic DSL:** ∀ tiedRanking (tieBreak) ⇔ {(eliminationRound ∧ tiedRanking)

	⊻(outcomeDetermination ∧ tiedRanking)} When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows: ○ Flipping a coin ○ Drawing straws ○ Rolling of dice either physically or digitally ○ Picking the high card from a deck of shuffled cards ○ High/low number in random number generation (Pseudo-Random) **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** ∀ singleWinnerContest (win) ⇔ { (∃ threshold) ∧ (∃ $\max_{\text{contestOptionVoteCount} \in [j]}$ ∧ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9)

	Skipped ranking: ballot is exhausted CA SB 1288, 22001(a); Colo. R. 26.8.2 **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.3 Determine Outcome: Statewide IRV_1

4.2.12.3	**Determine Outcome:** Statewide IRV_1
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by Alaska and Maine It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking

	Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: single candidate
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) }

Undervote Rules Case:
If a contest is undervoted, the ballot is exhausted for that contest.

Mathematical Logic DSL:

∀ (contest ∧ underVote)
{(underVote => exhaustBallot)}

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

∀ contestRanking (overVotedRanking) ⇔
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted ranking rules case:
If an overvoted ranking occurs in the current ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then if Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking

Mathematical Logic DSL:

∀ (contestRanking ∧ skippedRanking)
{(consecutiveSkippedRankings => exhaustBallot)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{⊻ (repeatRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseDynamic = (castVoteCountCurrentRound))
∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])}

Surplus Calculation Method
N/A

Candidate Elimination Method
Single Candidate

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws

	○ Rolling of dice either physically or digitally ○ Picking the high card from a deck of shuffled cards ○ High/low number in random number generation (Pseudo-Random) **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** ∀ singleWinnerContest (win) ⇔ { (∃ threshold) ∧ (∃ $\max_{\text{contestOptionVoteCount} \in [j]}$ ∧ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** 1 Va. Admin Code § 20-100-40(2)(b); Maine Rev. Stat. 21-723-A(D). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h),

	Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.4 Determine Outcome: CA-NM-OR-PortlandME (IRV)

4.2.12.4	**Determine Outcome:** CA-NM-OR-PortlandME (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by CA: Berkeley, San Francisco, San Leandro, Albany, Eureka ME: Portland NM: Las Cruces, Santa Fe OR: Benton County, Corvallis It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking

	Threshold: Dynamic, Whole number Candidate elimination: single candidate
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) }

	Undervote Rules Case: If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)} **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted ranking rules case:** If an overvoted ranking occurs in the current ranking, then **exhaust ballot** **Mathematical Logic DSL:** ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => exhaustBallot)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ (contestRanking ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ contestRankings ∧ overVotedRanking)
{(overVotedRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseDynamic = (castVoteCountCurrentRound))
∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])}

Surplus Calculation Method
N/A

Candidate Elimination Method
Single Candidate

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards

	○ High/low number in random number generation (Pseudo-Random) **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** ∀ singleWinnerContest (win) ⇔ { (∃ threshold) ∧ (∃ $\max_{\text{contestOptionVoteCount} \in [j]}$ ∧ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h),

	Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.5 Determine Outcome: Virginia Local Option (IRV)

4.2.12.5	**Determine Outcome:** Virginia Local Option (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by VA: Arlington County It applies the following processing rules: Overvoted ranking: advance to next ranking Skipped ranking: If consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: single candidate
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Selection Criteria

Plain Language: At most one selection may be made by the voter in each ranking position.

Mathematical Logic DSL:

∀ singleWinnerContest (selectionValid) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxSelectionsAllowed = N)
∧ (N <= M)
∧ (MinContestSelectionsAllowed = 1)
∧ (MaxSelectionsPerContestOption = 1)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and
N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Voting Exceptions and Voter Errors

Undervote

Plain Language:
An undervote in an RCV contest occurs when no selections are made in any ranking.

Mathematical Logic DSL:

∀ contest (underVote) ⇔
{ (ballotContestOptionSelectionCount = 0) }

Undervote Rules Case:
If a contest is undervoted, the ballot is exhausted for that contest.

Mathematical Logic DSL:

∀ (contest ∧ underVote)
{(underVote => exhaustBallot)}

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

$\forall$ contestRanking (overVotedRanking) $\Leftrightarrow$
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
$\wedge$ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **ballot advances to next ranking.**

Mathematical Logic DSL:

$\forall$ (contestRanking $\wedge$ overVotedRanking)
{(overVotedRanking => advanceToNextRanking)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

$\forall$ contestRanking (skippedRanking) $\Leftrightarrow$
{($\nexists$ underVote)
$\wedge$ ($\nexists$ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking nor the previous ranking, then **exhaust ballot**

Mathematical Logic DSL:

$\forall$ contestRanking (skippedRanking)
{(consecutiveSkippedRankings => exhaustBallot)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case:
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseDynamic = (castVoteCountCurrentRound))
∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])}

Surplus Calculation Method
N/A

Candidate Elimination Method
Single Candidate

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

	Determine Outcome Rule: See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** $\forall$ singleWinnerContest (win) $\Leftrightarrow$ { ($\exists$ threshold) $\wedge$ ($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$ $\wedge$ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** 1 Va. Admin Code § 20-100-40(2)(b); Maine Rev. Stat. 21-723-A(D). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e),

	HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.6 Determine Outcome: Portland, OR (IRV)

4.2.12.6	**Determine Outcome:** Portland, OR (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by OR: Portland It applies the following processing rules: Overvoted ranking: advance to next ranking Skipped ranking: advance to next ranking Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: single candidate
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed)

∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (N <= M)
∧ (MinContestSelectionsAllowed = 1)
∧ (MaxSelectionsPerContestOption = 1)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and
N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Voting Exceptions and Voter Errors

Undervote

Plain Language:
An undervote in an RCV contest occurs when no selections are made in any ranking.

Mathematical Logic DSL:
∀ contest (underVote) ⇔
{ (ballotContestOptionSelectionCount = 0) }

Undervote Rules Case:
If a contest is undervoted, the ballot is exhausted for that contest.

Mathematical Logic DSL:
∀ (contest ∧ underVote)
{(underVote => exhaustBallot)}

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

∀ contestRanking (overVotedRanking) ⇔
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **ballot advances to next ranking.**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => advanceToNextRanking)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking rules case:
If no selection is made at this ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧

(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case:
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:
∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:
∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseDynamic = (castVoteCountCurrentRound))
∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])}

Surplus Calculation Method
N/A

Candidate Elimination Method
Single Candidate

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:
∀ contestRound (elimination) ⇔
{(seatsFilledCount < seatsReqToFillCount)
∧ (seatsToFillCount < continuingCandidatesCount)
∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$)

∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:
∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language: The prevailing contest option is the contest option that meets or exceeds the threshold.

Mathematical Logic DSL:
∀ singleWinnerContest (win) ⇔

	{ (∃ threshold) ∧ (∃ $\max_{\text{contestOptionVoteCount} \in [j]}$ ∧ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§

	2.080.030(B)(2)(A), 2.080.030(C)(2)(C).
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.7 Determine Outcome: Utah Local Option (IRV)

4.2.12.7	**Determine Outcome:** Utah Local Option (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by Utah County, UT: Payson, Vineyard, Genola, Lehi, Woodland Hills Wasatch County, UT: Heber City Salt Lake County, UT: Kearns, Midvale, South Salt Lake It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: batch elimination
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M)

	∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Undervote Rules Case:** If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)} **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then **if Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking**

Mathematical Logic DSL:

∀ (contestRanking ∧ skippedRanking)
{(consecutiveSkippedRankings => exhaustBallot)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

	Repeat Ranking Rules Case: If a previously selected contest option is selected again, then **advance to the next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking) {(repeatRanking => advanceToNextRanking)} **Threshold Type** Dynamic - Threshold used is calculated from the total votes in the current round **Base Calculation** Whole Number Threshold In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote. **Mathematical Logic DSL:** ∀ (contestRanking ∧ threshold) {(∃ threshold) ∧ (thresholdBaseDynamic = (castVoteCountCurrentRound)) ∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])} **Surplus Calculation Method** N/A **Candidate Elimination Method** Batch elimination **Plain Language:** If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes. Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes. This specification is only applicable in single-winner RCV. Note that multi-winner batch elimination has never been adopted in any jurisdiction in the United States, and is not known to be in consideration for adoption, and is therefore out of scope. **Mathematical Logic DSL:** ∀ contestRound (batchElimination) ⇔

{(singleWinnerContest)
∧ (seatsFilledCount = 0)
∧ (continuingCandidatesCount > 3)
∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2}))
∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2}))
∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:
∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

	Plain Language: The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** $\forall$ singleWinnerContest (win) $\Leftrightarrow$ { ($\exists$ threshold) $\wedge$ ($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$ $\wedge$ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** 1 Va. Admin Code § 20-100-40(2)(b); Maine Rev. Stat. 21-723-A(D). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2).

	Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: batch elimination** CA SB 1288, 22100(d)–(e), Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.8 Determine Outcome: NYC (IRV)

4.2.12.8	**Determine Outcome:** NYC (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by New York City It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: batch elimination
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M)

	∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Undervote Rules Case:** If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)} **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ (contestRanking ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Ballot Advances to next ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case:
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Dynamic - Threshold used is calculated from the total votes in the current round

Base Calculation
Whole Number Threshold
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseDynamic = (castVoteCountCurrentRound))
∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])}

Surplus Calculation Method
N/A

Candidate Elimination Method
Batch elimination

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes.
Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes.
This specification is only applicable in single-winner RCV. Note that multi-winner batch elimination has never been adopted in any jurisdiction in the United States, and is not known to be in consideration for adoption, and is therefore out of scope.

Mathematical Logic DSL:

∀ contestRound (batchElimination) ⇔

	{(singleWinnerContest) ∧ (seatsFilledCount = 0) ∧ (continuingCandidatesCount > 3) ∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2})) ∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2})) ∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)} **Tiebreaking** **RCV Tiebreaking Cases:** In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking: **Tiebreaking between candidates being considered for elimination** Tie must be resolved between last place candidates in an elimination round. XOR **Tiebreaking between candidates for the (final) seats** Tie must be resolved to establish an outcome between tied candidates to be elected. **Mathematical Logic DSL:** ∀ tiedRanking (tieBreak) ⇔ {(eliminationRound ∧ tiedRanking) ⊻(outcomeDetermination ∧ tiedRanking)} When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows: ○ Flipping a coin ○ Drawing straws ○ Rolling of dice either physically or digitally ○ Picking the high card from a deck of shuffled cards ○ High/low number in random number generation (Pseudo-Random) **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

	Plain Language: The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** $\forall$ singleWinnerContest (win) $\Leftrightarrow$ { ($\exists$ threshold) $\wedge$ ($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$ $\wedge$ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Dynamic, Whole number** Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code §

	2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: batch elimination** CA SB 1288, 22100(d)–(e), Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.9 Determine Outcome: Minnesota Cities (IRV)

4.2.12.9	**Determine Outcome:** Minnesota Cities (IRV)
a.	**Description** The generic high-level decision rule specifies that, for a single-winner contest, the candidate with the largest count of votes is the prevailing candidate presented to election officials to be certified as elected. This variant is used by MN: Minneapolis, Minnetonka, Bloomington, St. Louis Park, St. Paul It applies the following processing rules: Overvoted ranking: Advance to next ranking Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: batch elimination
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N)

	∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Undervote Rules Case:** If a contest is undervoted, the ballot is exhausted for that contest. **Mathematical Logic DSL:** ∀ (contest ∧ underVote) {(underVote => exhaustBallot)} **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)

∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **advance to next ranking**

Mathematical Logic DSL:
∀ (contestRankings ∧ overVotedRanking)
{(overVotedRanking => advanceToNextRanking)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:
∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then **advance to next ranking**

Mathematical Logic DSL:
∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:
∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

	Repeat Ranking Rules Case: If a previously selected contest option is selected again, then advance to the next ranking **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking) {(repeatRanking => advanceToNextRanking)} **Threshold Type** Static - Threshold used is calculated from the total votes in the current round **Base Calculation** Whole Number Threshold **Whole Number Static** In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote. **Mathematical Logic DSL:** ∀ (contestRanking ∧ threshold) {(∃ threshold) ∧ (thresholdBaseStatic = (castVoteCountFirstRound)) ∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]} **Surplus Calculation Method** N/A **Candidate Elimination Method** Batch elimination **Plain Language:** If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes. Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes. This specification is only applicable in single-winner RCV. Note that multi-winner batch elimination has never been adopted in any jurisdiction in the United States, and is not known to be in consideration for adoption, and is therefore out of scope.

Mathematical Logic DSL:

∀ contestRound (batchElimination) ⇔
{(singleWinnerContest)
∧ (seatsFilledCount = 0)
∧ (continuingCandidatesCount > 3)
∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2}))
∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2}))
∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.

XOR

Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Determine Outcome Rule:

	See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. **Mathematical Logic DSL:** $\forall$ singleWinnerContest (win) $\Leftrightarrow$ { ($\exists$ threshold) $\wedge$ ($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$ $\wedge$ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20

	Candidate elimination: batch elimination Minneapolis Charter, 167.60(c)(1)(b), St Paul Ordinances, 31.05(d)(1)–(2)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.10 Determine Outcome: Utah Local Option (Multi-Pass IRV/Bloc Preferential Voting)

4.2.12.10	**Determine Outcome:** Utah Local Option (Multi-Pass IRV/Bloc Preferential Voting)
a.	**Description** Multi-Pass IRV (Bloc Preferential Voting), uses multiple iterations of the single-winner IRV tabulation process for multi-winner contests by electing candidates one at a time. After a candidate is determined to be elected, that candidate is removed from the list of continuing candidates before running a new round-by-round tabulation with the original ballots. This method is a majoritarian voting method and is not designed to produce proportional representation, in contrast to STV which was designed with the intent to provide proportional representation. This variant is used by Utah County, UT: Payson, Vineyard, Genola, Lehi, Woodland Hills Wasatch County, UT: Heber City Salt Lake County, UT: Kearns, Magna, Midvale, Millcreek, Salt Lake City, South Salt Lake It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking Repeat ranking: Advance to next ranking Threshold: Dynamic, Whole number Candidate elimination: batch elimination
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position.

Mathematical Logic DSL:

∀ singleWinnerContest (selectionValid) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (N <= M)
∧ (MinContestSelectionsAllowed = 1)
∧ (MaxSelectionsPerContestOption = 1)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and
N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:

See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Voting Exceptions and Voter Errors

Undervote

Plain Language:

An undervote in an RCV contest occurs when no selections are made in any ranking.

Mathematical Logic DSL:

∀ contest (underVote) ⇔
{ (ballotContestOptionSelectionCount = 0) }

Undervote Rules Case:

If a contest is undervoted, the ballot is exhausted for that contest.

Mathematical Logic DSL:

∀ (contest ∧ underVote)
{(underVote => exhaustBallot)}

Overvoted Ranking

Plain Language:

An overvoted ranking occurs when more than one contest option is selected at the ranking

	being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted Ranking Rules Case:** If an overvoted ranking occurs in the current ranking, then **exhaust ballot** **Mathematical Logic DSL:** ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => exhaustBallot)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **if Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ skippedRanking) {(consecutiveSkippedRankings => exhaustBallot)} **Repeat Ranking** **Plain Language:** A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

	Mathematical Logic DSL: ∀ contestRanking (repeatRanking) ⇔ {((∃ rankingSelectionPerSpecificContestOption) ∧ (rankingSelectionsPerSpecificContestOptionCount >= 2))} **Repeat Ranking Rules Case:** If a previously selected contest option is selected again, then **advance to the next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking) {(repeatRanking => advanceToNextRanking)} **Threshold Type** Dynamic - Threshold used is calculated from the total votes in the current round **Base Calculation** Whole Number Threshold In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote. **Mathematical Logic DSL:** ∀ (contestRanking ∧ threshold) {(∃ threshold) ∧ (thresholdBaseDynamic = (castVoteCountCurrentRound)) ∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])} **Surplus Calculation Method** N/A **Candidate Elimination Method** Batch elimination **Plain Language:** If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes. Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes.

This specification is only applicable in single-winner RCV but can be used here since Block Preferential Voting is actually processed as sequential single-winner IRV.

Mathematical Logic DSL:

∀ contestRound (batchElimination) ⇔
{(singleWinnerContest)
∧ (seatsFilledCount = 0)
∧ (continuingCandidatesCount > 3)
∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2}))
∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2}))
∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.

XOR

Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

	Decision Rule Set: See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** The prevailing contest option is the contest option that meets or exceeds the threshold. For multi-pass IRV/Bloc Preferential Voting, the tabulation process is repeated until all seats are filled. **Mathematical Logic DSL:** $\forall$ singleWinnerContest (win) $\Leftrightarrow$ { ($\exists$ threshold) $\wedge$ ($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$ $\wedge$ ($\max_{\text{contestOptionVoteCount} \in [j]}$ >= threshold) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n).
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** 1 Va. Admin Code § 20-100-40(2)(b); Maine Rev. Stat. 21-723-A(D). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2).

	Threshold: Dynamic, Whole number Dynamic: Berkeley Code, tit. 2, 2.14.040(B), Oakland Charter, art. 11, §1105(d)(2), Maine Rev. Statutes, 21-A M.R.S.A. § 723-A(2)(A), San Francisco Charter, art. 13, § 13.102(d), Portland Charter, art. II, § 3(c), San Leandro Code, tit. 1, § 1-11-320(b), Takoma Park Charter, § 606(e), HR 3057 § 331(a)(2)(A), Santa Fe Code 9-1.15(B)(7); Portland, OR City Code § 2.080.030(B)(2). Whole Number: Minneapolis Charter, tit. 85, ch. 167.20; Portland, OR Code 2.08.030(C)(2)(a). **Candidate elimination: batch elimination** CA SB 1288, 22100(d)–(e), Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f) **Round-by-round count** Utah Code 20A-4-603(7)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.11 Determine Outcome: Colorado Local Option (STV)

4.2.12.11	**Determine Outcome:** Colorado Local Option (STV)
a.	**Description** Single Transferrable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus

	redistribution or transfer, which may be a fractional value. This variant is authorized by 1-7-1003 CO Revised Statutes It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: ballot is exhausted Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: single candidate Surplus calculation method: Fractional ballot surplus transfer
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors**

	Undervote **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted Ranking Rules Case:** If an overvoted ranking occurs in the current ranking, then **exhaust ballot** **Mathematical Logic DSL:** ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => exhaustBallot)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => exhaustBallot)

Repeat Ranking

Plain Language:

A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case

If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type

Static - Threshold used is calculated from the total votes in the current round

Base Calculation

Whole Number Threshold

In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus
Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold.

Mathematical Logic DSL:

∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method
Fractional ballot surplus transfer

Plain Language
For each candidate that is designated as elected and has a surplus, the surplus of that candidate shall be redistributed or transferred as follows:

The surplusFraction for the candidate shall be calculated as the quotient, rounded down to a set number, e.g., four, decimal places, of the candidate's surplus votes divided by the total number of votes for the candidate.

Each ballot counted for the candidate shall be redistributed or transferred to, and counted for, the highest-ranked continuing candidate on that ballot using a surplusTransferValue, calculated as the product, rounded down to the set number of decimal places, of the current RCVVoteValue for the elected candidate times the candidate's surplusFraction.

"Residual surplus" means the portion of surplus votes not transferred pursuant to the above paragraphs due to rounding. The residual surplus for the transfer equals the surplus for the elected candidate minus the sum of the surplusTransferValues for every ballot transferred from the elected candidate. The residual surplus shall not be counted for any candidate nor as part of other exhausted votes.

After the candidate's surplus is transferred and their residual surplus is calculated, the candidate no longer has a surplus, and the candidate's vote total for the remainder of the tabulation shall be equal to the threshold. An elected candidate no longer receives transferred or redistributed votes in subsequent rounds.

Mathematical Logic DSL:

∀ contestRound (transferFractionalBalanceSurplus) ⇔
{(contestOptionVoteCount > threshold)
∧ (surplusFraction = (surplus / contestOptionVoteCount)
∧ (surplusTransferValue = RCVVoteValue * surplusFraction)}

Candidate Elimination Method
Single Candidate

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Candidate Elimination Method
Single Candidate

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language:
If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold.

Mathematical Logic DSL:

$\forall$ contest (win) $\Leftrightarrow$
{($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$
((contestOptionVoteCount >= threshold)
$\wedge$ (winnerCount = seatsToFillCount)
$\veebar$
(seatsToFillCount = continuingCandidatesCount))}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9) **Skipped ranking: ballot is exhausted** CA SB 1288, 22001(a); Colo. R. 26.8.2 **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C). **Surplus calculation method: Fractional ballot surplus transfer** Threshold determination: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(a) Surplus fraction: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Surplus: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Transfer value: Minneapolis Code tit. 8.5, ch. 167.20

	Portland, OR Code 2.08.030(C)(2)(b)(1)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.12 Determine Outcome: Albany, CA (STV)

4.2.12.12	**Determine Outcome:** Albany, CA (STV)
a.	**Description** Single Transferrable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus redistribution or transfer, which may be a fractional value. This variant is used by CA: Albany It applies the following processing rules: Overvoted ranking: ballot is exhausted Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: single candidate Surplus calculation method: Fractional ballot surplus transfer
b.	**Constraints**

See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Selection Criteria

Plain Language: At most one selection may be made by the voter in each ranking position.

Mathematical Logic DSL:

∀ singleWinnerContest (selectionValid) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (N <= M)
∧ (MinContestSelectionsAllowed = 1)
∧ (MaxSelectionsPerContestOption = 1)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and
N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Voting Exceptions and Voter Errors

Undervote

Plain Language:
An undervote in an RCV contest occurs when no selections are made in any ranking.

Mathematical Logic DSL:

∀ contest (underVote) ⇔
{ (ballotContestOptionSelectionCount = 0) }

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking

being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

∀ contestRanking (overVotedRanking) ⇔
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **exhaust ballot**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔

{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:
∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Static - Threshold used is calculated from the total votes in the first round

Base Calculation
Whole Number Threshold
In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:
∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus
Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold.

Mathematical Logic DSL:
∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method
Fractional ballot surplus transfer

Plain Language:
For each candidate that is designated as elected and has a surplus, the surplus of that candidate shall be redistributed or transferred as follows:

The surplusFraction for the candidate shall be calculated as the quotient, rounded down to a set number, e.g., four, decimal places, of the candidate's surplus votes divided by the total number of votes for the candidate.

Each ballot counted for the candidate shall be redistributed or transferred to, and counted for, the highest-ranked continuing candidate on that ballot using a surplusTransferValue, calculated as the product, rounded down to the set number of decimal places, of the current RCVVoteValue for the elected candidate times the candidate's surplusFraction.

"Residual surplus" means the portion of surplus votes not transferred pursuant to the above paragraphs due to rounding. The residual surplus for the transfer equals the surplus for the elected candidate minus the sum of the surplusTransferValues for every ballot transferred from the elected candidate. The residual surplus shall not be counted for any candidate nor as part of other exhausted votes.

After the candidate's surplus is transferred and their residual surplus is calculated, the candidate no longer has a surplus, and the candidate's vote total for the remainder of the tabulation shall be equal to the threshold. An elected candidate no longer receives transferred or redistributed votes in subsequent rounds.

Mathematical Logic DSL:

∀ contestRound (transferFractionalBalanceSurplus) ⇔
{(contestOptionVoteCount > threshold)
∧ (surplusFraction = (surplus / contestOptionVoteCount)
∧ (surplusTransferValue = RCVVoteValue * surplusFraction)}

Candidate Elimination Method

Single Candidate

Plain Language:

If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

∀ contestRound (elimination) ⇔
{(seatsFilledCount < seatsReqToFillCount)
∧ (seatsToFillCount < continuingCandidatesCount)
∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$)
∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:
∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Candidate Elimination Method
Single Candidate

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate with the fewest votes.

Mathematical Logic DSL:
∀ contestRound (elimination) ⇔
{(seatsFilledCount < seatsReqToFillCount)

	$\wedge$ (seatsToFillCount < continuingCandidatesCount) $\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$) $\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn) **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language** If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold. **Mathematical Logic DSL:** $\forall$ contest (win) $\Leftrightarrow$ {($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$) $\wedge$ ((contestOptionVoteCount >= threshold) $\wedge$ (winnerCount = seatsToFillCount) $\veebar$ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9).

	Skipped ranking: Advance to next ranking Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C). **Surplus calculation method: Fractional ballot surplus transfer** Threshold determination: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(a) Surplus fraction: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Surplus: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Transfer value: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.13 Determine Outcome: Virginia Local Option (STV)

4.2.12.13	**Determine Outcome:** Virginia Local Option (STV)
a.	**Description** Single Transferrable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects

	multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus redistribution or transfer, which may be a fractional value. This variant is used by VA: Arlington County It applies the following processing rules: Overvoted ranking: Advance to next ranking Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: single candidate Surplus calculation method: Fractional ballot surplus transfer
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)

	∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted Ranking Rules Case:** If an overvoted ranking occurs in the current ranking, then **advance to next ranking**

	Mathematical Logic DSL: ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => advanceToNextRanking)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **if consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking.** **Mathematical Logic DSL:** ∀ (contestRankings ∧ skippedRanking) {(consecutiveSkippedRankings => exhaustBallot)} **Repeat Ranking** **Plain Language:** A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking. **Mathematical Logic DSL:** ∀ contestRanking (repeatRanking) ⇔ {((∃ rankingSelectionPerSpecificContestOption) ∧ (rankingSelectionsPerSpecificContestOptionCount >= 2))} **Repeat Ranking Rules Case** If a previously selected contest option is selected again, then **advance to the next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking) {(repeatRanking => advanceToNextRanking)}

Threshold Type
Static - Threshold used is calculated from the total votes in the first round

Base Calculation
Whole Number Threshold
In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus
Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold.

Mathematical Logic DSL:

∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method
Fractional ballot surplus transfer

Plain Language
For each candidate that is designated as elected and has a surplus, the surplus of that candidate shall be redistributed or transferred as follows:

The surplusFraction for the candidate shall be calculated as the quotient, rounded down to a set number, e.g., four, decimal places, of the candidate's surplus votes divided by the total number of votes for the candidate.

Each ballot counted for the candidate shall be redistributed or transferred to, and counted for, the highest-ranked continuing candidate on that ballot using a surplusTransferValue, calculated as the product, rounded down to the set number of decimal places, of the current RCVVoteValue for the elected candidate times the candidate's surplusFraction.

"Residual surplus" means the portion of surplus votes not transferred pursuant to the above

	paragraphs due to rounding. The residual surplus for the transfer equals the surplus for the elected candidate minus the sum of the surplusTransferValues for every ballot transferred from the elected candidate. The residual surplus shall not be counted for any candidate nor as part of other exhausted votes. After the candidate's surplus is transferred and their residual surplus is calculated, the candidate no longer has a surplus, and the candidate's vote total for the remainder of the tabulation shall be equal to the threshold. An elected candidate no longer receives transferred or redistributed votes in subsequent rounds. **Mathematical Logic DSL:** ∀ contestRound (transferFractionalBalanceSurplus) ⇔ {(contestOptionVoteCount > threshold) ∧ (surplusFraction = (surplus / contestOptionVoteCount) ∧ (surplusTransferValue = RCVVoteValue * surplusFraction)} **Candidate Elimination Method** Single Candidate **Plain Language:** If the seat is not filled, eliminate the candidate with the fewest votes. **Mathematical Logic DSL:** **Sequential Elimination:** ∀ contestRound (elimination) ⇔ {(seatsFilledCount < seatsReqToFillCount) ∧ (seatsToFillCount < continuingCandidatesCount) ∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$) ∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn) **Tiebreaking**

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Candidate Elimination Method
Single Candidate

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

	Determine Outcome Rule: See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold. **Mathematical Logic DSL:** ∀ contest (win) ⇔ {(∃ $\max_{\text{contestOptionVoteCount} \in [j]}$) ∧ ((contestOptionVoteCount >= threshold) ∧ (winnerCount = seatsToFillCount) ⊻ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: If Consecutive rankings skipped, ballot goes inactive. Otherwise advance to next ranking** 1 Va. Admin Code § 20-100-40(2)(b); Maine Rev. Stat. 21-723-A(D). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco

	Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C). **Surplus calculation method: Fractional ballot surplus transfer** Threshold determination: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(a) Surplus fraction: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Surplus: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Transfer value: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.14 Determine Outcome: Portland, OR (STV)

4.2.12.14	**Determine Outcome:** Portland, OR (STV)
a.	**Description** Single Transferrable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate

	receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus redistribution or transfer, which may be a fractional value. This variant is used by OR: Portland It applies the following processing rules: Overvoted ranking: Advance to next ranking Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: single candidate Surplus calculation method: Fractional ballot surplus transfer
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and

N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Voting Exceptions and Voter Errors

Undervote

Plain Language:
An undervote in an RCV contest occurs when no selections are made in any ranking.

Mathematical Logic DSL:

∀ contest (underVote) ⇔
{ (ballotContestOptionSelectionCount = 0) }

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

∀ contestRanking (overVotedRanking) ⇔
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **advance to next ranking**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => advanceToNextRanking)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has

	no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **advance to the next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ skippedRanking) {(skippedRanking => advanceToNextRanking)} **Repeat Ranking** **Plain Language:** A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking. **Mathematical Logic DSL:** ∀ contestRanking (repeatRanking) ⇔ {((∃ rankingSelectionPerSpecificContestOption) ∧ (rankingSelectionsPerSpecificContestOptionCount >= 2))} **Repeat Ranking Rules Case** If a previously selected contest option is selected again, then **advance to the next ranking** **Mathematical Logic DSL:** ∀ (contestRankings ∧ repeatRanking) {(repeatRanking => advanceToNextRanking)} **Threshold Type** Static - Threshold used is calculated from the total votes in the first round **Base Calculation** Whole Number Threshold In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the

nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus

Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold.

Mathematical Logic DSL:

∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method

Fractional ballot surplus transfer

Plain Language

For each candidate that is designated as elected and has a surplus, the surplus of that candidate shall be redistributed or transferred as follows:

The surplusFraction for the candidate shall be calculated as the quotient, rounded down to a set number, e.g., four, decimal places, of the candidate's surplus votes divided by the total number of votes for the candidate.

Each ballot counted for the candidate shall be redistributed or transferred to, and counted for, the highest-ranked continuing candidate on that ballot using a surplusTransferValue, calculated as the product, rounded down to the set number of decimal places, of the current RCVVoteValue for the elected candidate times the candidate's surplusFraction.

"Residual surplus" means the portion of surplus votes not transferred pursuant to the above paragraphs due to rounding. The residual surplus for the transfer equals the surplus for the elected candidate minus the sum of the surplusTransferValues for every ballot transferred from the elected candidate. The residual surplus shall not be counted for any candidate nor as part of other exhausted votes.

After the candidate's surplus is transferred and their residual surplus is calculated, the candidate no longer has a surplus, and the candidate's vote total for the remainder of the tabulation shall be equal to the threshold. An elected candidate no longer receives transferred or redistributed

votes in subsequent rounds.

Mathematical Logic DSL:

∀ contestRound (transferFractionalBalanceSurplus) ⇔
{(contestOptionVoteCount > threshold)
∧ (surplusFraction = (surplus / contestOptionVoteCount)
∧ (surplusTransferValue = RCVVoteValue * surplusFraction)}

Candidate Elimination Method

Single Candidate

Plain Language:

If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

Sequential Elimination:

∀ contestRound (elimination) ⇔
{(seatsFilledCount < seatsReqToFillCount)
∧ (seatsToFillCount < continuingCandidatesCount)
∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$)
∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Tiebreaking

RCV Tiebreaking Cases:

In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination

Tie must be resolved between last place candidates in an elimination round.

XOR

Tiebreaking between candidates for the (final) seats

Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)

	⊻(outcomeDetermination ∧ tiedRanking)} When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows: ○ Flipping a coin ○ Drawing straws ○ Rolling of dice either physically or digitally ○ Picking the high card from a deck of shuffled cards ○ High/low number in random number generation (Pseudo-Random) **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Candidate Elimination Method** Single Candidate **Plain Language:** If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate with the fewest votes. **Mathematical Logic DSL:** ∀ contestRound (elimination) ⇔ {(seatsFilledCount < seatsReqToFillCount) ∧ (seatsToFillCount < continuingCandidatesCount) ∧ (∃ $\min_{\text{contestOptionVoteCount} \in [j]}$) ∧ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn) **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold. **Mathematical Logic DSL:** ∀ contest (win) ⇔

	$\{(\exists \max_{contestOptionVoteCount \in [j]})$ $\wedge$ ((contestOptionVoteCount >= threshold) $\wedge$ (winnerCount = seatsToFillCount) $\underline{\vee}$ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).

	Surplus calculation method: Fractional ballot surplus transfer Threshold determination: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(a) Surplus fraction: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Surplus: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Transfer value: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.15 Determine Outcome: Minnesota Cities (STV)

4.2.12.15	**Determine Outcome:** Minnesota Cities (STV)
a.	**Description** Single Transferrable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus redistribution or transfer, which may be a fractional value.

	This variant is used by MN: Minneapolis It applies the following processing rules: Overvoted ranking: Advance to next ranking Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate elimination: batch Surplus calculation method: Fractional ballot surplus transfer
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote**

	Plain Language: An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (ballotContestOptionSelectionCount = 0) } **Overvoted Ranking** **Plain Language:** An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods. **Mathematical Logic DSL:** ∀ contestRanking (overVotedRanking) ⇔ {(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking) ∧ (MaxSelectionsAllowedPerRanking = 1)} **Overvoted Ranking Rules Case:** If an overvoted ranking occurs in the current ranking, then **advance to next ranking** **Mathematical Logic DSL:** ∀ (contestRanking ∧ overVotedRanking) {(overVotedRanking => advanceToNextRanking)} **Skipped Ranking** **Plain Language:** A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection. **Mathematical Logic DSL:** ∀ contestRanking (skippedRanking) ⇔ {(∄ underVote) ∧ (∄ rankingSpecificSelection)} **Skipped Ranking Rules Case:** If no selection is made at this ranking, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Static - Threshold used is calculated from the total votes in the first round

Base Calculation
Whole Number Threshold
In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus
Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold. (Minneapolis actually delays the implementation of surplus transfer until/unless it is necessary

for determination of the final outcome.)

Mathematical Logic DSL:

∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method

Fractional ballot surplus transfer

Plain Language:

For each candidate that is designated as elected and has a surplus, the surplus of that candidate shall be redistributed or transferred as follows:

The surplusFraction for the candidate shall be calculated as the quotient, rounded down to a set number, e.g., four, decimal places, of the candidate's surplus votes divided by the total number of votes for the candidate.

Each ballot counted for the candidate shall be redistributed or transferred to, and counted for, the highest-ranked continuing candidate on that ballot using a surplusTransferValue, calculated as the product, rounded down to the set number of decimal places, of the current RCVVoteValue for the elected candidate times the candidate's surplusFraction.

"Residual surplus" means the portion of surplus votes not transferred pursuant to the above paragraphs due to rounding. The residual surplus for the transfer equals the surplus for the elected candidate minus the sum of the surplusTransferValues for every ballot transferred from the elected candidate. The residual surplus shall not be counted for any candidate nor as part of other exhausted votes.

After the candidate's surplus is transferred and their residual surplus is calculated, the candidate no longer has a surplus, and the candidate's vote total for the remainder of the tabulation shall be equal to the threshold. An elected candidate no longer receives transferred or redistributed votes in subsequent rounds.

Mathematical Logic DSL:

∀ contestRound (transferFractionalBalanceSurplus) ⇔
{(contestOptionVoteCount > threshold)
∧ (surplusFraction = (surplus / contestOptionVoteCount)
∧ (surplusTransferValue = RCVVoteValue * surplusFraction)}

Candidate Elimination Method
Batch

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate or candidates with the fewest votes.
Batch elimination, the simultaneous elimination of multiple candidates, may be used, subject to jurisdiction rules, if the combined vote total of a group of candidates is less than the candidate with the next higher vote total. Thus there must exist a continuing candidate whose vote count exceeds at least two candidates with fewer votes.

This specification is designed only for single-winner RCV. Minneapolis has its own unique batch elimination process not modeled here.

Mathematical Logic DSL:

∀ contestRound (batchElimination) ⇔
{(singleWinnerContest)
∧ (seatsFilledCount = 0)
∧ (continuingCandidatesCount > 3)
∧ (∃(potentialEliminateCandidateSet {x | x ∈ potentialEliminateCandidateSet, x >= 2}))
∧ (∃(potentialContinuingCandidateSet {x | x ∈ potentialContinuingCandidateSet, x >= 2}))
∧ (∑ potentialEliminateCandidateSetVoteCount) < nextHighestContinuingCandidateVoteCount)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

∀ tiedRanking (tieBreak) ⇔
{(eliminationRound ∧ tiedRanking)
⊻(outcomeDetermination ∧ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Candidate Elimination Method
Single Candidate

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language:
If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold.

Mathematical Logic DSL:

$\forall$ contest (win) $\Leftrightarrow$
{($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$

	((contestOptionVoteCount >= threshold) ∧ (winnerCount = seatsToFillCount) ∨ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), St Paul Ordinances, 31.05(d)(4)(b); Portland, OR City Code § 2.08.030(A)(4). **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: batch elimination** Minneapolis Charter, 167.60(c)(1)(b), St Paul Ordinances, 31.05(d)(1)–(2) **Surplus calculation method: Fractional ballot surplus transfer** Threshold determination: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(a)

	Surplus fraction: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Surplus: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1) Transfer value: Minneapolis Code tit. 8.5, ch. 167.20 Portland, OR Code 2.08.030(C)(2)(b)(1)
f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.12.16 Determine Outcome: Cambridge, MA (STV)

4.2.12.16	**Determine Outcome:** Cambridge, MA (STV)
a.	**Description** Single Transferable Vote (STV) is the most common name for the variant of multi-winner RCV we illustrate here. Multi-winner RCV is a Ranked Choice Voting variant that elects multiple candidates in a single contest and redistributes any surplus votes above the threshold to lower-ranked candidates to ensure proportional representation. Multi-winner RCV elections occur in a series of rounds. Those rounds can consist of counting votes, eliminating candidates, transferring votes from eliminated candidates, electing candidates, and redistributing surplus votes from elected candidates. Counting multi-winner RCV elections proceeds as follows: all first-choice rankings are counted first. If no candidate receives a number of votes necessary for election (typically known as the threshold of election), then the candidate with the fewest votes is eliminated. Ballots counting for that candidate then transfer to the next highest-ranked continuing candidate on each ballot unless there is an intervening selection that disqualifies the ballot. These rounds of counting, elimination, and redistribution recur until all seats up for election are filled. **Threshold** For this STV variant's threshold of election, e.g., the number of votes needed for any candidate to be elected, the whole number threshold rule is used. This is separate from the surplus redistribution or transfer, which may be a fractional value. This variant is used by MA: Cambridge It applies the following processing rules: Overvoted ranking: Ballot is exhausted

	Skipped ranking: Advance to next ranking Repeat ranking: Advance to next ranking Threshold: Static, Whole number Candidate: single elimination (first round: batch for candidates with < 50 votes) Surplus calculation method: Whole ballot surplus transfer
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. **Mathematical Logic DSL:** ∀ singleWinnerContest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (N <= M) ∧ (MinContestSelectionsAllowed = 1) ∧ (MaxSelectionsPerContestOption = 1) ∧ (MaxSelectionsAllowedPerRanking = 1)} Where M is the number of contest options in the list available to the voter, the ith contest option (m_i,m_{i+1}…, m_m) and N is the number of selections allowed to be allocated by the voter in the contest. **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Voting Exceptions and Voter Errors** **Undervote** **Plain Language:** An undervote in an RCV contest occurs when no selections are made in any ranking. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔

{ (ballotContestOptionSelectionCount = 0) }

Overvoted Ranking

Plain Language:
An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.

Mathematical Logic DSL:

∀ contestRanking (overVotedRanking) ⇔
{(rankingSpecificSelectionCount > MaxSelectionsAllowedPerRanking)
∧ (MaxSelectionsAllowedPerRanking = 1)}

Overvoted Ranking Rules Case:
If an overvoted ranking occurs in the current ranking, then **exhaust ballot.**

Mathematical Logic DSL:

∀ (contestRanking ∧ overVotedRanking)
{(overVotedRanking => exhaustBallot)}

Skipped Ranking

Plain Language:
A skipped ranking occurs when a contest is not undervoted and the ranking being processed has no selection.

Mathematical Logic DSL:

∀ contestRanking (skippedRanking) ⇔
{(∄ underVote)
∧ (∄ rankingSpecificSelection)}

Skipped Ranking Rules Case:
If no selection is made at this ranking, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ skippedRanking)
{(skippedRanking => advanceToNextRanking)}

Repeat Ranking

Plain Language:
A repeat ranking occurs when a contest option is selected, when it had been selected at a previous ranking.

Mathematical Logic DSL:

∀ contestRanking (repeatRanking) ⇔
{((∃ rankingSelectionPerSpecificContestOption) ∧
(rankingSelectionsPerSpecificContestOptionCount >= 2))}

Repeat Ranking Rules Case
If a previously selected contest option is selected again, then **advance to the next ranking**

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Threshold Type
Static - Threshold used is calculated from the total votes in the first round

Base Calculation
Whole Number Threshold
In the case of whole number static, the threshold is equal to the number of valid votes in the first round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Mathematical Logic DSL:

∀ (contestRanking ∧ threshold)
{(∃ threshold)
∧ (thresholdBaseStatic = (castVoteCountFirstRound))
∧ (threshold = Rnddown0[(thresholdBaseStatic / (seatsReqToFill + 1)) + 1)]}

Transfer Surplus
Transfer surplus is applicable in STV when votes for a contest option exceeds the threshold.

Mathematical Logic DSL:

∀ contestRound (transferSurplus) ⇔
{(contestOptionVoteCount > threshold)}

Surplus Calculation Method
Whole ballot surplus transfer

Plain Language
Cambridge processes ballots from precincts in a randomly chosen order and selects every Nth ballot where N is the total votes for the elected candidate divided by the excess votes (rounded to the nearest whole number) and transfers the full vote of the selected CVRs to continuing candidates.

Candidate Elimination Method
Single candidate

Cambridge has an additional provision that provides for elimination of any candidate with fewer than 50 votes in the first round.

Plain Language:
If the seat is not filled, eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Tiebreaking

RCV Tiebreaking Cases:
In case of a tie in a single winner RCV contest, the cases are one of two mutually exclusive decision alternatives for tiebreaking:

Tiebreaking between candidates being considered for elimination
Tie must be resolved between last place candidates in an elimination round.
XOR
Tiebreaking between candidates for the (final) seats
Tie must be resolved to establish an outcome between tied candidates to be elected.

Mathematical Logic DSL:

$\forall$ tiedRanking (tieBreak) $\Leftrightarrow$
{(eliminationRound $\wedge$ tiedRanking)
$\veebar$(outcomeDetermination $\wedge$ tiedRanking)}

When a tie occurs the mechanism for breaking ties is a process that most commonly involves chance. Examples of tiebreaking methods involving chance that have been used are as follows:

- Flipping a coin
- Drawing straws
- Rolling of dice either physically or digitally
- Picking the high card from a deck of shuffled cards
- High/low number in random number generation (Pseudo-Random)

Decision Rule Set:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Candidate Elimination Method
Single Candidate

Plain Language:
If not all seats are filled and there are more continuing candidates than seats to fill, , eliminate the candidate with the fewest votes.

Mathematical Logic DSL:

$\forall$ contestRound (elimination) $\Leftrightarrow$
{(seatsFilledCount < seatsReqToFillCount)
$\wedge$ (seatsToFillCount < continuingCandidatesCount)
$\wedge$ ($\exists$ $\min_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$ (potentialEliminateCandidateSet = $\min_{\text{contestOptionVoteCount} \in [j]}$)}

Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)

Determine Outcome Rule:
See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language
If a contest option's vote total equals or exceeds the threshold, then the contest option wins. If the number of continuing contest options equals the number of seats left to fill, then the continuing contest options win even if their vote totals do not meet the threshold.

Mathematical Logic DSL:

$\forall$ contest (win) $\Leftrightarrow$
{($\exists$ $\max_{\text{contestOptionVoteCount} \in [j]}$)
$\wedge$

	((contestOptionVoteCount >= threshold) ∧ (winnerCount = seatsToFillCount) ⊻ (seatsToFillCount = continuingCandidatesCount))} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (nj, nj+1, … , nn)
c.	**Tabulation Methods Process Set** See Sec. 3.2.17, Process Step 6, "Process a staged CVR"
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** See VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Row 76 See FINAL_OCTOBER2024_RCV Profiles for VMWG **Overvoted ranking: ballot is exhausted** San Francisco Charter, § 13.102, CA SB 1288, 22002 (2016), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), Santa Fe Code, 9-1.15(B)(9). **Skipped ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Oakland Charter, 1105(g)–(h), Berkeley Code, 2.14.070, St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), NCGS 163-329(b1)(2), Santa Fe Code, 9-1.15(B)(12); Portland, OR City Code § 2.08.030(A)(3). **Repeat ranking: Advance to next ranking** Minneapolis Charter, 167.60(c)(2), Berkeley Code, 2.14.080, Oakland Charter, 1105(g)–(h), Maine Rev. Statutes, tit. 21 § 723-A(2), St Paul Ordinances, 31.05(d)(4)(b), San Francisco Charter, § 13.102(a), Takoma Park Charter, § 606(g), NCGS 163-329(b1)(2), Santa Fe Code 9-1.15(B)(4), Maine Rev. Statutes, tit. 21 § 723-A(2). **Threshold: Static, Whole number** Minneapolis Charter, tit. 85, ch. 167.20 **Candidate elimination: single candidate** Maine Rev. Statutes, tit. 21 §§ 723-A(2)(B); 723-A(4)(B), St Paul Ordinances, 31.05(d)(1)–(2), Takoma Park Charter, § 606(f), Santa Fe Code 9-1.15(E)(3); Portland, OR Code §§ 2.080.030(B)(2)(A), 2.080.030(C)(2)(C).

	Surplus calculation method: Whole ballot surplus transfer
	Mass. Gen. Laws 54A,
	Mass. Gen. Laws 54A,
	http://www.cambridgema.gov/election/~/link.aspx?_id=D58142E0FBC64BEDB4B0F92D957FD453&_z=z;http://www.cambridgema.gov/election/~/link.aspx?_id=D58142E0FBC64BEDB4B0F92D957FD453&_z=z;
	Mass. Gen. Laws 54A,
	http://www.cambridgema.gov/election/~/link.aspx?_id=D58142E0FBC64BEDB4B0F92D957FD453&_z=z;

f.	**Test Conditions & Expected Outcomes** See **Sec. 3.2.23.2 Test Conditions and Expected Outcomes**

4.2.13 Top-Two IRV Contingent Vote Contest

4.2.13.1 Determine Outcome: Top-Two IRV Contingent Vote Contest, Candidates, Single Seat

4.2.13.1	**Determine Outcome: Top-Two IRV Contingent Vote Contest, Candidates, Single Seat**
a.	**Description** Top-2 IRV, also known as Contingent Vote, is a form of instant runoff voting where all but the top two 1st round ballot options are eliminated after the 1st round of tabulation. The traditional IRV tabulation process is a multi-round process that is either always used or is used if no candidate receives sufficient votes in tabulated first choices to meet the requirements to be elected. If no candidate receives the threshold in the initial tabulation of marked first choices, the two candidates with the greatest number of first choice votes advance to an RCV round of counting. In this round, each ballot counts as a vote for whichever of the two final candidates is ranked highest by the voter. The candidate with the most votes in the RCV runoff round is determined to be elected.
b.	**Constraints** In addition to the High-Level Decision Rule Constraints Applicable to Voting Methods, Sec. 4.2.2 and the constraints in High-Level Requirements Applicable to Most RCV Methods, Sec. 4.2.12.1 the specific constraint for this voting method variant is as follows. **Selection Criteria** **Plain Language:** At most one selection may be made by the voter in each ranking position. Candidate selections cannot be repeated at more than one ranking / contest option. I.e. Candidate 'A' cannot be selected at ranking X and also ranking Y.

Mathematical Logic DSL:

∀ singleWinnerContest (selectionValid) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (MaxSelectionsAllowedPerRanking = 1)
∧ (MaxSelectionsPerContestOption = 1)
∧ (N <= M)}

Repeat Ranking_ Advance to Next Ranking:

Plain Language:
If a previously selected contest option is selected again, then advance to the next ranking

Mathematical Logic DSL:

∀ (contestRankings ∧ repeatRanking)
{(repeatRanking => advanceToNextRanking)}

Decision Rule Set:
See Sec 4.2.12.1.

Threshold rule:

Plain Language Whole-Number Threshold:
Whole number threshold applies to all IRV variants. A majority of votes from the continuing ballots. A "Threshold" sets the number of votes needed to guarantee a win and is calculated based on the number of ballots cast and the number of seats to fill.

Whole Number Dynamic
In the case of whole number dynamic the threshold is equal to the number of valid votes in the current round divided by one more than the number of seats to fill and rounded down to the nearest whole number plus one vote.

Whole Number Threshold Dynamic
Threshold = ((Total votes counted in Round)/(Seats to be elected + 1)) +1, with any fractions disregarded.

Mathematical Logic DSL:

∀ (contestRankings ∧ threshold)

	{(∃ threshold) ∧ (thresholdBaseDynamic = (castVoteCountCurrentRound)) ∧ (threshold = Rnddown0[(thresholdBaseDynamic / (seatsReqToFill + 1)) + 1])} **Transfer Surplus:** N/A **Plain Language: Elimination of Candidates After Counting in the First Round** After tabulation of marked first choices, eliminate all but top two candidates. **Mathematical Logic DSL:** **Sequential Elimination:** ∀ contestRound (elimination) ⇔ {(seatsFilledCount < seatsReqToFillCount) ∧ (continuingCandidatesCount >= 3) ∧ (∃ min(contestOptionVoteCount ∈[j])) ∧ (potentialEliminateCandidateSet = min(contestOptionVoteCount ∈[j])) ∧ (potentialContinuingCandidateSetCount = 2) } Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n). **Determine Outcome Rule:** See Sec 4.2.12.1. **Plain Language:** The contest option with the most votes wins, **Mathematical Logic DSL:** ∀ contest (win) ⇔ {(∃ max (contestOptionVoteCount ∈[j])} Where N is the number of selections allowed to be allocated by the voter in the contest, and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n). VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine Outcome
c.	**Tabulation Methods Process Set** See Sec. 3.2.18, Process Step 6, "Process a staged CVR"

d.	**Tabulation Scenarios Set** See Sec. 4.1, Tabulation Scenarios, Central Tabulation See Sec. 4.1, Tabulation Scenarios, Multi-Contest-Multi Voting Method per Ballot
e.	**Authoritative References** N.C. Gen. Stat. § 163-329(b1)(2) (2010) VMWG-NIST-1500-107-Voting Methods and Tabulation Universe, Links to RCV Resources Tab **Links to Legislative Sources** VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Determine Outcome(Top Two Advance)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.9.5, Top-Two IRV, Test Conditions and Expected Outcomes, Table 17

4.2.14 Presidential Preference Primary

4.2.14.1 Presidential Preference Primary Election

4.2.14.1	**Presidential Preference Primary Election**
a.	**Description** This variant is distinct from most commonly implemented presidential primary contests, which are baseline N-of-M contests. In this variant, the Presidential Preference contest refers to a controlling contest on a Presidential Preference Primary partisan ballot that is linked to a set of Presidential Delegate candidates which may be grouped in delegate contests affiliated with the presidential candidate or in contests with candidates that are affiliated with different presidential candidates. The election selects presidential delegates that will go to the nominating convention for each party's candidates. Each delegate candidate is affiliated with their party's specific presidential candidate. A valid vote selection in the Presidential Preference contest allows vote selections in the Presidential Delegate contest(s) to be tabulated and only those candidates affiliated with the Presidential Preference contest selection are eligible to be tabulated.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods See Sec. 4.2.5. Interrelated Contests or Connected Contests **Selection Criteria** **Plain Language:** A valid contest option selection must be made in the presidential primary contest to allow selections in the Presidential Delegate Contest(s).

	Mathematical Logic DSL: ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = 1)} Where N is the number of selections allowed to be allocated by the voter in the contest **Counting Rules:** Each individual interrelated contest has a tabulation process set that is equal to the tabulation process set for its voting method. See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Decision Rule Set:** The decision rule set is the decision rule set for each individual contest that is qualified given application of the Interrelated Contests filter and the contest's voting method, as described in that module. See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods There exists a relationship between Presidential Preference and Presidential Delegate. If we view this as a decision tree, Presidential Preference execution determines a subset of candidates that are available, i.e., the subset of delegates. **Determine Outcome Rule:** See the outcome rule for each individual contest that is qualified given application of the Interrelated Contests filter and the contest's voting method, as described in that module: See Sec. 4.2.5. Interrelated Contests or Connected Contests Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods See 4.2.15 Presidential Delegate Contest.
c.	**Tabulation Methods Process Set** See Sec. 3.2.3, Process N-of-M, Baseline Contest and Sec. 4.2.5. Interrelated Contests or Connected Contests
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** Ala. Code § 17-13-43

	AL GOP Bylaws Art. XI (2020) 17 R.I. Gen. Laws §§ 12.1-1; 12.1-2 R.I. GOP Del. Selection Plan Section IV, R. 1.0.1 (2020) VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Links to Presidential Preference Primary Resources Tab

4.2.14.1.1 Determine Outcome: Presidential Preference Contest, Controller

4.2.14.1.1	**Determine Outcome: Presidential Preference Contest, Controller**
a.	**Description** Tabulation of selections in a Presidential Delegate contest is only enabled if there is a valid selection in the Presidential Preference contest. The ballot may contain multiple Presidential Delegate contests, each affiliated with a different presidential candidate, or one or more delegate contests with candidates affiliated with different presidential candidates. Only the candidates on the ballot as delegates for the associated presidential candidate in the Presidential Preference contest will be eligible for tabulation. Selections of any other candidates will be ignored.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods See Sec. 4.2.5. Interrelated Contests or Connected Contests **Selection Criteria** **Plain Language:** 1. A candidate must be selected in the presidential primary contest to allow selections in the Presidential Delegate Contest(s). 2. Only candidates affiliated with the selected presidential candidate are eligible to be selected in the Delegate contest. All other selections will be ignored. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = 1)} **Counting Rules:** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3,

	High-Level Decision Rule Constraints Applicable to Voting Methods **Plain Language:** Tabulated using N-of-M counting rules. M only includes the list of candidates on the ballot as delegates for the associated presidential candidate in the Presidential Preference contest. **Mathematical Logic DSL:** ∀ contest (validBallotContestCount) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (contestCounterInitialValue = 0) ∧ (contestCounterIncrement = 1) ∧ (contestCounterValue = (contestCounterValue + contestCounterIncrement)) ∧ (contestCounterValue <= ballotsCastCount) } Where N is the number of selections allowed to be allocated by the voter in the contest **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods
c.	**Tabulation Methods Process Set** See Sec. 3.2.4, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Links to Presidential Preference Primary Resources Tab Ala. Code § 17-13-43 AL GOP Bylaws Art. XI (2020) 17 R.I. Gen. Laws §§ 12.1-1; 12.1-2 R.I. GOP Del. Selection Plan Section IV, R. 1.0.1 (2020)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.9.6, Expected Outcomes and Sample Test Sets, Presidential Preference

	Contest, Table 18

4.2.14.1.2 Determine Outcome: Presidential Delegate Contest, Controlled

4.2.14.1.2	**Determine Outcome: Presidential Delegate Contest, Controlled**
a.	**Description** Tabulation of selections in a Presidential Delegate contest is only enabled if there is a valid selection in the Presidential Preference contest. The ballot may contain multiple Presidential Delegate contests, each affiliated with a presidential candidate or one or more delegate contests containing candidates affiliated with different presidential candidates. Only the candidates on the ballot as delegates for the associated presidential candidate in the Presidential Preference contest will be eligible for tabulation. Selections of any other candidates will be ignored.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods See Sec. 4.2.5. Interrelated Contests or Connected Contests **Selection Criteria** **Plain Language:** A valid candidate selection must have been made in the controlling Presidential Delegate Controlling Contest to allow selections in the Presidential Delegate Contest(s). Only candidates affiliated with the selected presidential candidate are eligible to be selected in the Delegate contest. All other selections will be ignored. The selection criteria is the selection criteria of the voting variant of the controlled contest, which may be any voting method in use for the jurisdiction in the United States, as determined by the jurisdiction. Most commonly the voting method is N-of-M also known as plurality. **Mathematical Logic DSL:** ∀ contest (selectionValid) ⇔ { (∃ contestOptionSelection) ∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed) ∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed) ∧ (MaxContestSelectionsAllowed = N) ∧ (MinContestSelectionsAllowed = 1) ∧ (N = 1) }

Where N is the number of selections allowed to be allocated by the voter in the contest

Plain Language:
A valid vote in the controlling contest is required in order for the controlled contest to be tabulated.

Mathematical Logic DSL:

∀ contest (tabulateControlled) ⇔
{(∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >=MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (N = 1) }

Where N is the number of selections allowed to be allocated by the voter in the contest.

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods

Plain Language:
Tabulated using N-of-M counting rules. M only includes the list of candidates on the ballot as delegates for the associated presidential candidate in the Presidential Preference contest.

Mathematical Logic DSL:

∀ contest (validBallotContestCount) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (contestCounterInitialValue = 0)
∧ (contestCounterIncrement = 1)
∧ (contestCounterValue = (contestCounterValue + contestCounterIncrement))
∧ (contestCounterValue <= ballotsCastCount) }

Where N is the number of selections allowed to be allocated by the voter in the

	contest **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods
c.	**Tabulation Methods Process Set** See Sec. 3.2.4, Process N-of-M, Baseline Contest
d.	**Tabulation Scenarios Set** See Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** VMWG-NIST-1500-107-VotingMethodAndTabulationUniverse, Links to Presidential Preference Primary Resources Tab Ala. Code § 17-13-43 AL GOP Bylaws Art. XI (2020) 17 R.I. Gen. Laws §§ 12.1-1; 12.1-2 R.I. GOP Del. Selection Plan Section IV, R. 1.0.1 (2020)
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.9.6, Expected Outcomes and Sample Test Sets, Presidential Preference Contest, Table 18

4.2.15 Massachusetts Group Vote Contest

4.2.15.1 Determine Outcome: Massachusetts Group Vote Contest

4.2.15.1	**Determine Outcome: Massachusetts Group Vote Contest**
a.	**Description** Massachusetts Group Voting is a voting method that has always coincided with the Massachusetts Presidential Primary ballot and is used to elect members of a Town, Village or Ward (political party) Committee. Due to the large number of candidates that can be selected (e.g., 35), the contest format using this method has one or more group selections as a contest option, with individual contest options listed after the group selections. Candidates can either be associated with a group or be independent. The contest may also contain write-in selections which would be independent of any group. A group selection acts as if all candidates in the group are individually selected. Direct selection of a candidate in a selected group is redundant.
b.	**Constraints** See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3,

High-Level Decision Rule Constraints Applicable to Voting Methods

Selection Criteria

Plain Language:
The total number of candidates that may be selected, whether directly selected and/or by group selection, cannot exceed the allowed number of selections.
Candidates may be selected directly or by selecting a candidate group. A candidate is considered selected either by selecting a group associated with the candidate and/or by direct selection.

Mathematical Logic DSL:

∀ contest (castVote) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)}

Where N is the number of selections allowed to be allocated by the voter in the contest

Counting Rules:
See Sec. 4.2.2, Constraints that generally apply to most Voting Methods and Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods
Counting rules follow N-of-M contest counting rules. Candidates are considered selected whether individually selected or selected via a group selection.

Plain Language: Overvote
As in an N-of-M contest, an overvote exists if there are more candidates selected than the number allowed. If a group is selected that is full (has as many candidates as seats to fill), any additional selections (candidates outside the group) will result in an overvote.

Mathematical Logic DSL:

∀ contest (validBallotContestCount) ⇔
{ (∃ contestOptionSelection)
∧ (selectedBallotContestOptionsCount <= MaxContestSelectionsAllowed)
∧ (selectedBallotContestOptionsCount >= MinContestSelectionsAllowed)
∧ (MaxContestSelectionsAllowed = N)
∧ (MinContestSelectionsAllowed = 1)
∧ (contestCounterInitialValue = 0)
∧ (contestCounterIncrement = 1)
∧ (contestCounterValue = (contestCounterValue + contestCounterIncrement))

	∧ (contestCounterValue <= ballotsCastCount) } Where N is the number of selections allowed to be allocated by the voter in the contest **Plain Language: Undervote** If a group has fewer candidates than allowed, additional individual or group selections may be made as long as the total number of candidates does not exceed the allowed number. Undervotes will be tallied if the total number of candidates selected is less than the allowed number. **Mathematical Logic DSL:** ∀ contest (underVote) ⇔ { (selectedBallotContestOptionsCount < MinContestSelectionsAllowed) } **Decision Rule Set:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods **Determine Outcome Rule:** See Sec. 4.2.3, High-Level Decision Rule Constraints Applicable to Voting Methods
c.	**Tabulation Methods Process Set** See Sec. 3.2.22, Process Massachusetts Group Vote Contest
d.	**Tabulation Scenarios Set** VMWG-NIST-1500-107_2020_Voting Methods and Tabulation Models, Sec. 4.1.1, Tabulation Scenarios
e.	**Authoritative References** Mass. Gen Laws ch. 52 § 9; ch. 53 §§ 34, 35, 70E
f.	**Test Conditions & Expected Outcomes For This Variant** See Sec. 3.2.23.9.7, Expected Outcomes and Sample Test Sets, Massachusetts Group Vote Contest, Table 19 , Table 1

4.3 Domain Models

Election Business Process Model

The NIST-EAC Election Modeling Process Model outlines a set of activities commonly carried out by election administrators within the United States, spanning the entire lifecycle from voter registration to vote tallying and the certification of results. It uses a standard Business Process Model and Notation (BPMN) approach to provide, as much as possible, an unambiguous understanding of processes. BPMN's ability to compose very

complex processes allows for elections to be viewed at 10,000-foot or closeup, that is, from a process map down to a logical set of tasks.

The model may serve as a reference to election officials, system developers, and other stakeholders.

Election Business Process Model BPMN Diagram
Voting Methods and Tabulation Methods at…

VMWG Voting Methods Universe

As part of its work modeling voting and tabulation methods used in the United States, the VVSG Voting Methods Working Group collected example statutes or other sources of law representing the voting methods modeled in this standard. Those sources of law were then broken down into their various rules, requirements, processes, and other components and cataloged in the Voting Method and Tabulation Universe spreadsheet at this link.

The Working Group also conducted extensive research into the laws governing two detailed/complex forms of tabulation and voting used in the united states: Post-Election Auditing and ranked choice voting. Each available ranked choice voting amendment, ordinance, statute, or regulation, and each available state-level auditing statute or regulation is included as an authoritative resource that was used to produce this body of work. The folder includes additional auditing or testing laws, regulations, and guides when their contents arose in research.

While examples were researched and cataloged, there was no such comprehensive research of other voting or tabulation methods, such as pre-election logic & accuracy testing, voting system certification, or baseline plurality contests.

The RCV folder includes a table of contents laying out what laws are or are not included in the folder, whether because they could not be found online or because they do not exist in the relevant jurisdiction.

A list of post-election audit laws not available online at the time of the research follows.

- Kansas § 25-3009
- Maryland election law § 11-309
- North Carolina § 163-182.1
- Utah Audit policy (as required by the law included in the folder)

Appendix A: DSL Dictionary and Glossary

VMWG Domain Specific Language Dictionary

The purpose of the VMWG Domain Specific Language (DSL) is to provide unambiguous labels with definitions for defining and specifying the VMWG Voting and Tabulation systems domain model - also known as the VMWG Universe. The unambiguous human-readable labels are designed so that the objects, things, processes, enumerations, etc. that are components of the voting and election systems universe can be specified precisely. The VMWG DSL is designed to support domain experts' active participation in writing precise specifications of voting and tabulation methods - without being mathematicians or computer scientists, and similarly be humanly readable for various stakeholders to understand and use the specifications. We have drawn upon well-established standards for developing human-readable software code and applied research on commonly understood election domain terms of art.

The VMWG DSL consists of the labels defined in the VMWG DSL Dictionary, mathematical logic, and other math symbols as connectives. In this specification, we have used the VMWG DSL to create precise voting method specifications as collections of clauses that represent the constraint logic in conjunctive standard form to define the necessary and sufficient clauses that must be true for a voting method to be correctly implemented. The VMWG DSL Dictionary serves the second purpose of consistently labeling components in the VMWG UML model for the voting systems universe.

Voting Methods Working Group (VMWG), NIST Interoperability Working Group

Version: November 2024 Authors: VMWG
NIST SP 1500-107 Voting Methods and Tabulation Models
UML Model and Constraint DSL Dictionary

All contest configurations per election must be immutable once configured via configuration management.
Boolean Type means: Range of values is True/False (T/F, 0/1).
Configure Y/N indicates whether the element is to be configured through configuration management.
Counter indicates the element is a counter, subject to constraints on counters such as: All counters must be initially set to zero and monotonically increment-only during the operation of the election.
Artifact indicates the element is a physical or digital work product.
Constant means that the element is immutable, typically required to be configured, also a type of Configure = T.

Label	**Category/ Type/ Configure Y/N**	**Purpose/Description** *** = VVSG 2.0 definition**
advanceToNextRanking	Boolean	After determining the current ranking can no longer be counted in a given round of a contest, this label directs the user/program to consider the next ranking on the ballot.

Label	**Category/ Type/ Configure Y/N**	**Purpose/Description** *** = VVSG 2.0 definition**
ballot	Artifact, digital (or paper)	"Presentation of the contest options for a particular voter." - Voluntary Voting System Guidelines 2.0, PP 266 https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf
ballotContestOptionSelectionCount	Numeric	A count of the selections in a contest on a single ballot, whether valid or invalid. In the case of a ranked ballot, the counter associated with a distinct contest across the specific contest ranking(s).
ballotCount	Numeric	The total count value held within any ballotCounter.
ballotCounter	Counters Config: Initialization	Any counter that aggregates a count of ballots of different categories, such as ballots cast total, ballot cast by style, ballots cast by party, blank ballots, etc. The count value must be zero at initialization.
ballotsCastCount	Numeric	The total count value held within any ballotCounter counting Cast ballots, where a Cast ballot is valid, i.e., no adjudication is needed or the ballot already went through adjudication.
batchElimination	Process	In ranked choice voting, elimination of multiple candidates simultaneously based on the criterion that none of them could mathematically prevail.
blankBallot	Boolean	"An issued ballot without any selections made. Synonyms: unmarked ballot" -VVSG 2.0 Glossary
candidateSelection	Boolean	A detected mark or selection of a contest option in an electable office seat contest, whether valid or invalid.
candidateSelectionInvalidated	Boolean	One or more selections were made that are not valid.
castVote	Boolean	One or more votes Cast in a single contest on a ballot, where a Cast ballot is valid, i.e., no adjudication is needed or the ballot already went through adjudication.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
castVoteCount	Numeric	A count of all votes Cast for a contest,where a Cast ballot is valid, i.e., no adjudication is needed or the ballot already went through adjudication.
castVoteCountCurrentRound	Numeric	A count of all valid votes cast in a round of ranked choice voting.
castVoteCountFirstRound	Numeric	A count of all valid votes cast in the first round of ranked choice voting.
castVoteRecord	Artifact, digital (or paper) Config: No	One or more (contest option selections recorded on a ballot) Cast Vote Records or an aggregation of castVoteRecords in the form of CastVoteRecord, in the NIST CVR CDF may satisfy the castVoteRecord format requirement
castVoteRecordReport	Artifact, digital (or paper) Config: No	A complete record as required by the jurisdictions reporting of one or more CastVoteRecords or an aggregation of CastVoteRecords in the form of CastVoteRecordReports, in the NIST CVR CDF may satisfy the CastVoteRecord format requirement
castVoteRecordSnapshot	Artifact, digital (or paper) Config: No	The artifact stored for every round of RCV tabulation. Like the final output, the snapshots are stored in the NIST 1500-103 Cast Vote Record Common Data Format (CVR CDF). The full CVR Record (CVRR) final output includes all of the inter-round and per round RCV snapshots.
configuredContest	Boolean Config: No	Indicates a contest within an election cycle has been or has not yet been configured.
connectedContest	Data Object Config: Initialization	A contest whose tabulation is affected by a selection in another contest. Examples are a partisan contest in a general election affected by a straight party contest and delegate contests affected by a valid selection in a presidential contest in a presidential primary election.
consecutiveSkippedRankings	Boolean	Consecutive skipped rankings occur if no selection is made at two sequential skipped rankings before its highest continuing ranking.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
constant	Config: Initialization	An immutable object that is configured with a specific value or that is explicitly specified to be equal to a particular value; the value does not change.
contest	Data Object Config: Initialization	An entity on a ballot with one or more selectable contest options and a specified number of allowed selections. Includes referenda/questions and electable office seats.
contestCounterIncrement	Numeric	The number of votes added to the contestOptionVoteCount for a valid selection of a contest option of a specific contest on each tabulated ballot.
contestCounterInitialValue	Numeric Config: Initialization	The initial vote count in a contest as initialized in the configuration. The count value must be zero at initialization.
contestOption	Data Object	A selectable choice in a contest. Includes candidates and write-ins in an office seat contest and yes/for and no/against in a referendum/question.
contestOptionSelection	Boolean	An explicit selection made by a voter in a specific contest on a given ballot, whether valid or invalid, and captured in the CVR.
contestOptionVoteCount	Numeric Config: Initialization	The value at any point in time of a counter that counts the number of valid selections of a specific contest option. The count value must be zero at initialization.
contestRankings	Data Object	The set of all ranking selections for a single contest.
contestRound	Process	In an RCV contest,"round" identifies a step in the counting process during which selections for continuing candidates are tabulated for the purpose of determining whether a candidate has achieved a threshold of the votes cast for a particular office.
contestVoteCounterValue	Value	Any value accumulation of a counter associated with a specific contest. Includes vote counter, undervote counter,

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
	Config: Initialization	overvote counter. Aggregated total at a point in time for the jurisdiction applicable to the contest. The count value must be zero at initialization.
continuingCandidate(s)	Value	A candidate (or candidates) who has neither been elected nor eliminated
continuingCandidatesCount	Numeric	The number of candidates who have neither been elected nor eliminated
controlledContest	Config: Initialization	A contest whose tabulation is influenced by a valid selection in a controllingContest. Examples are a partisan contest in a general election controlled by a valid selection in a straight party contest and a partisan contest in an open primary election controlled by a valid selection in a party preference contest.
controllingContest	Config: Initialization	A contest where a valid option selection influences the tabulation of one or more controlledContests. Examples are a straight party contest in a General election, a party preference contest in an Open Primary election and the controlling contest in a retention contest pair.
controllingSelection	Boolean	A selection in a controlling contest that is valid according to its contest tabulation/decision rules.
counter	Counters Config: Initialization	Any counter used in tabulation. Includes contest counters, contest statistic counters and ballot counters The count value must be zero at initialization.
counterIncrement	Numeric	A positive number.
decisionRuleSet	Config: Initialization	The set of rules associated with contest option selection processing for the defined contest type that defines validity, vote tabulation and effect on any connected contests.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
directSelection	Boolean	Any explicit selection of a contest option in a specific contest. This is opposed to an indirectSelection of a contest option in a specific controlledContest resulting from a valid directSelection in a controlling contest.
elected	Boolean	The candidate determined to have been awarded the seat in a given contest based on the aggregated accumulated totals tabulated according to the contest decision rules. This is not official until certified by the designated canvassing board.
eliminatedCandidate	Boolean	The term used to identify a candidate/contest option that has been eliminated in a previous RCV round or one that is to identify a candidate/contest option that will be eliminated in the current RCV round. An eliminated candidate is no longer considered in future rounds.
eliminationRound	Boolean	The name for a RCV round in which one or more lowest vote getters are eliminated and votes for those eliminated contest options are redistributed to the next highest ranked continuing contest options.
exclusiveRule	Boolean	Valid votes in an individual contest connected to the Straight Party question are counted and the Straight Party selections are not counted in that contest. Any selection invalidates the straight party.
exhaustBallot	Boolean	Applies to an RCV contest and occurs during a round of processing when either the current selection is invalid and rules indicate the ballot is no longer eligible for further processing or the rules require processing the next selection but there are no further selections of active candidates
inclusiveRule	Boolean	Valid votes in an individual contest connected to the Straight Party are counted. If the Straight Party selections in combination with the contest selection do not result in an overvote in the contest, then the Straight Party votes are also counted.
indirectSelection	Boolean	The selection of a contest option in a controlled contest as a result of a valid option selection in a controllingContest.

Label	**Category/ Type/ Configure Y/N**	**Purpose/Description** *** = VVSG 2.0 definition**
MaxContestSelectionsAllowed	Constant Numeric Config: Yes	The maximum number of voter 'contest choices' or voter selections allowed for a particular contest on a ballot. Immutable constant once configured.
MaxContestSetSelectionsAllowed		The maximum number of voter 'contest choices' or voter selections allowed for a set of partisan contests for a specific party on a ballot. Immutable constant once configured.
MaxRoundVoteSelectionsAllowed	Constant Numeric Config: Yes	The maximum number of voter 'contest choices' or voter selections allowed for a particular contest on a ballot - per round where the voting method variant has multiple rounds in which voter selections are allocated to a set of one or more Continuing Candidates. Most commonly the constant value is configured to equal 1. Immutable constant once configured.
MaxSelectionsAllowedPerContestOption	Constant Numeric Config: Yes	The maximum number of voter 'contest choices' or voter selections allowed for a particular contest option for a contest on a ballot. Immutable constant once configured.
MaxSelectionsAllowedPerRanking	Constant Numeric Config: Yes	The maximum number of voter selections allowed for a particular ranking for an RCV contest. Immutable constant once configured.
maxWinners	Numeric	Maximum number of candidates who can reach threshold potentially. Must be less than or equal to the number of seats required to be filled.
MinContestSelectionsAllowed	Constant Config: Yes	The minimum number of voter 'contest choices' or voter selections allowed for a particular contest on a ballot. Most commonly configured to equal zero. Immutable constant once configured.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
multiWinnerContest	Boolean	Any contest for which there are multiple seats. Multi-winner: The candidates, equal to the number of seats to be filled, with the most votes, after all votes are counted, are elected
nextHighestContinuingCandidateVoteCount	Numeric	The number of votes for a continuing candidate with the next highest vote count above another candidate or set of candidates to which it is being compared.
outcome	Boolean	A contest outcome is either the election of a candidate or candidates to fill a contest's seat(s) or the result of a referenda/question contest.
outcomeDetermination	Process	The application of the Decision Rule Set and the Determine Outcome Rule to establish one or more prevailing contest options.
outcomeDetermined	Boolean	True when the application of the Decision Rule Set and the Determine Outcome Rule to establish one or more prevailing contest options has occurred and therefore an outcome has been determined.
overVote	Boolean	In a given contest, an overvote exists when the number of contest option selections is more than the allowed number on a specific ballot.
overVoteCounter	Counters Config: Initialization	A counter that tabulates overvotes for a specific contest. It can either aggregate votes lost due to overvote or ballots containing an overvote for the contest. The count value must be zero at initialization.
overVotedRanking	Boolean	An overvoted ranking occurs when more than one contest option is selected at the ranking being processed. Contest Overvote does not apply in RCV voting methods.
partyPreferenceSelection	Boolean	The selection of a party contest option in a Party Preference contest in an Open Primary election or a presidential candidate option in a Presidential contest in a Presidential/Delegate Primary. Both restrict tabulation to

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
		contests connected to the option selection.
partySelection	Boolean	The selection of the preferred party option in either the straight party contest in a General election or the party preference contest in an Open Primary election.
passed	Boolean	The successful result of a question/referenda contest based on the aggregated accumulated totals tabulated according to the contest decision rules. This is not official until certified by the designated canvassing board. Must be configured to False / zero at initialization.
potentialContinuingCandidateSet	Data set	All continuing candidates less the potentialEliminateCandidateSet
potentialContinuingCandidateSetCount	Numeric	The count of candidates who have not been eliminated or identified in the potentialEliminateCandidateSet
potentialEliminateCandidateSet	Boolean	The set of candidates/contest options that may be eliminated in the current RCV round.
potentialEliminateCandidateSet	Data set	In an RCV contest when batch elimination rules apply, this is the set of candidates whose vote totals are such that there is no possibility of any of them winning.
potentialEliminateCandidateSetCount	Numeric	In an RCV contest when batch elimination rules apply, it is the number of candidates in the set of candidates whose vote totals are such that there is no possibility of winning for any of them and will concurrently be eliminated in the processing of the current round
potentialEliminateCandidateSetVoteCount	Numeric	The number of votes in the set of candidates/contest options that may be eliminated in the current RCV round.
rankingSelection	Boolean	Selection of an RCV contest option at a specific ranking.
rankingSelectionCounter	Counters Config: Initialization	Increments by one for each selection within a ranking. The count value must be zero at initialization.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
rankingSelectionCounterValue	Numeric	The value accumulated at any point in time and stored in a rankingPositionSelectionCounter.
rankingSelectionPerSpecificContestOption	Boolean	A ranking selection within a single ballot and specific contest option.
rankingSelectionsPerSpecificContestOptionCount	Numeric	A count of ranking selections within a single ballot for a specific contest option.
rankingSpecificSelection	Boolean	In an RCV contest, the selection of a specific contest option at a specific ranking.
rankingSpecificSelectionCount	Numeric	Number of selections made within a specific ranking within an RCV contest.
RCVVoteValue	Numeric	Applies to a Ranked Choice Voting multi-seat contest where proportional transfer occurs of excess votes from a winning candidate who exceeds the threshold. The RCVVoteValue at the beginning of tabulation of each ballot is 1. If a candidate earns more votes than needed to win, each RCVVoteValue counting for that candidate is multiplied by the surplusFraction to generate a surplusTransferValue. That surplusTransferValue is subtracted from the original RCVVoteValue to determine a new (reweighted) RCVVoteValue, which stays with the winning candidate. The calculated SurplusTransferValue transfers to the next candidate on each ballot. Once all transfers are complete, each surpusTransferValue that a candidate receives is treated as a distinct RCVVoteValue in future rounds. When a candidate is eliminated, each RCVVoteValue transfers at its current value to the next candidate ranked on that ballot. The sum of all RCVVoteValues counting for a candidate is equal to the total number of votes a candidate has at the end of a round of counting. See Concrete Example [footnote 01] The count value must be configured to equal one (1) at

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
		initialization.
repeatRanking	Boolean	In an RCV contest, the selection of a contest option in a specific ranking that has also been selected in a previous ranking in the same contest.
report	Artifact	The electronic or hard copy presentation of any aspect of a specific election. Examples include initial, current or final election summary results for a precinct or precinct set, a list of precincts in the election and a spreadsheet of overall election results by precinct or precinct subset showing candidate totals as well as overall totals.
reportCounterSet	Results Report Element Config: Yes	The set of associated counters of aggregated tabulated ballot results that appear on a designated report. Reports include machine totals, precinct totals, district totals, district total, jurisdiction totals, etc. It can all or a subset of selected contests,
residualValue	Numeric	Occurs when using multi-winner RCVwith fractional transfer of surplus votes. It is the fraction of votes that are not transferred due to rounding of fractional transfer such that total votes total to the number of ballots cast.
seatsFilledCount	Numeric	The total number of contest options winners that have been determined.
seatsFilledCountRCV	Numeric	The number of winners that have been determined in an RCV contest following any round.
seatsReqToFillCount	Numeric	The total number of seats required to fill.
seatsToFillCount	Numeric	The remaining number of seats to fill following any round in RCV tabulation.
selectedBallotContestOptionsCount	Numeric	A count of a contest options selections in a specific contest on a specific ballot.
selectionValid	Boolean	A contest option selection in a given contest on a specific ballot that can be counted as a vote according to tabulation rules,

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
singleWinnerContest	Boolean	Any contest for which there is exactly one seat or one outcome, such as a ballot measure.
skippedRanking	Boolean	A skipped ranking occurs when a voter does not make a selection in a given ranking of an RCV contest.
snapshot	Artifact	For RCV (i.e., IRV or STV) the tabulation method produces "snapshots" at every step within a round for every round of a tabulation, and like the final output, the snapshots are stored in the NIST 1500-103 Cast Vote Record Common Data Format (CVR CDF).
straightPartyIndication	Boolean	A valid selection of a contest option in a Straight Party contest on a partisan general election ballot
superMajority	Boolean	A contest that requires a larger percentage of votes to be elected than the > 50% required in a majority contest. (usually 60%?)
surplus	Numeric	In the case of a multi-seat RCV contest, votes in excess of the threshold that a candidate receives during a round of processing
surplusFraction	Numeric	In the case of a Ranked Choice Voting multi-seat contest where proportional transfer of excess votes from a winning candidate who exceeds the threshold is reached. The surplusFraction is the number of excess votes for the winning candidate, above threshold, divided by the total vote. It is normally the decimal equivalent of the fraction truncated to four digits.
surplusRedistributionRound	Process	In the case of a RCV multi-seat contest, the proportional transfer of excess votes from a winning candidate or candidates who exceeds the threshold.

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
surplusTransferValue	Numeric	Is used in a Ranked Choice Voting multi-seat contest where excess votes from candidates that exceed the threshold are proportionally redistributed. The surplusTransferValue is calculated by multiplying a vote's pre-existing RCVVoteValue by the surplusFraction. The transferred vote to the candidate from each ballot is normally the decimal equivalent of the percentage truncated to four decimal places.
tabulateControlled	Process	Applies to a Recall/Replace contest pair where the Recall contest is controlling and must contain a selection in order for Replace, controlled contest, to be tabulated. Tabulation rules determine whether the selection must be valid.
tabulationMethod	Process	A tabulation method is a set of partially ordered process steps and operations for performing counting, accumulation, aggregation, and other data and math operations on vote selection data sets. Each tabulation method supports at least one tabulation scenario.
tabulationScenario	Process	A tabulation scenario is a situation and sequence of events or activities intended by election administrators to be sufficient to achieve a tabulation goal.
threshold	Numeric	A "Threshold" sets the number of votes needed to determine the outcome and is calculated based on the number of valid votes and the number of seats to fill.
thresholdBase	Numeric	The count of votes cast that is used in determining a threshold in an RCV contest round.
thresholdBaseCurrentRound	Numeric	The thresholdBaseCurrentRound is the count of active votes cast in the current round in a RCV contest. It is the numerator in determining a contest threshold in some RCV contests. Typically used in a single-winner (IRV) contest. Denominator is the number of seats + 1.
thresholdBaseDynamic	Numeric	The count of votes cast in the current round that is used when updating the threshold in each round of an RCV contest.
thresholdBaseFirstRound	Numeric	The thresholdBaseFirstRound is the count of votes cast in the

Label	Category/ Type/ Configure Y/N	Purpose/Description * = VVSG 2.0 definition
		first round in an RCV contest. It is the numerator in determining a contest threshold in some RCV contests. Typically used in multi-winner RCV (STV) Denominator is the number of seats + 1.
thresholdBaseStatic	Numeric	The count of votes cast in the first round that is used in determining the threshold in an RCV contest where the threshold is constant across all rounds.
tie	Boolean	Occurs when the outcome of a contest cannot be determined due to the aggregated totals associated with the contested contest options being equal.
tieBreak	Boolean	True when a tie is resolved between two or more candidates or measures, such as to determine contest outcome or RCV ranking.
tiedRanking	Boolean	In RCV tabulation, a tiedRanking exists when the eliminated candidate in a round cannot be determined due to candidates with the least votes having an equal number of votes, when candidates exceed the threshold with an equal number of votes in a multi-seat contest and the order of surplus redistribution must be determined, and when a winner cannot be determined due to candidates having a equal number of votes in a final round.
transferSurplus	Boolean	A transfer surplus exists if the contestOptionVoteCount is greater than the threshold in a multi winner (STV) RCV contest.
underVote	Boolean	In a given non-RCV contest, one or more undervotes exist when the number of contest option selections is fewer than the allowed number on a specific ballot. An undervote in an RCV contest occurs when no selections are made in any ranking.
underVoteCounter	Counters Config: Initialization	A counter that tabulates undervotes for a specific contest. The count value must be zero at initialization.

Label	**Category/ Type/ Configure Y/N**	**Purpose/Description** *** = VVSG 2.0 definition**
validBallotContestCount	Boolean	Identifies a selected contest option that is valid and will be used to increment the counter that accumulates votes for that option by the number of associated votes (normally one).
validContestOptionSelections	Boolean	The set of selected contest options in a given contest that have been validated as countable.
vote	Boolean	Indication of support for a particular contest option.*
voteCounter	Counters Config: Initialization	Any counter that aggregates selections for a specific contest option. The count value must be zero at initialization.
voteSelectionDataSet	Data set	A vote selection data set representing from 1 to n ballots is required as a prerequisite to tabulation.
voteValue	Numeric	The numeric value of a valid contest option selection based on contest tabulation rules.
votingMethod	Enumeration	A voting method encompasses a set of partially ordered process steps, tabulation methods, and decision rules. Each voting method has at least one associated tabulation method and at least one election outcome decision rule set consisting of at least one decision rule. Executing the appropriate tabulation method and election contest decision rule set is in the scope of the voting method.
win	Boolean	There exists a prevailing outcome for a contest option.
winner	Boolean	Prevailing contest option/candidate, *subject to certification by the certifying authority.* Wi = the ith winning candidate. W1 is the first candidate to win. W2 is the second candidate to win.
winnerCount	Numeric	The number of contest options/candidates who have met the criteria for being declared a winner

Domain Specific Language (DSL) Notation

Symbol	Meaning
∀	For all; for any; for each; universal quantification
∃	There exists; existential quantification
∃!	There exists precisely one; singleton existential quantification
∄	There does not exist;
∧	Logical AND Logical conjunction
∨	Logical OR Logical inclusive disjunction
⊻	Logical XOR; exclusive OR
⇒	implication; implies; if…then
⇔	if and only if; iff
¬	negation
T (or True)	True
F (or False)	False
()	Precedence grouping
≡, :=	Equivalence, Equal by definition
s.t., |	Such that
+. -, *, /, <, >, =	Usual meanings in arithmetic
+. -, *, /, <, >, ≥ ≤ ≪ ≫ =, ≈ ~≠ ± ± ÷ % mod Δ	Usual meanings in arithmetic

$\cap \cup \subseteq \subset \not\subset \supseteq \supset \not\supset \{ \} \emptyset$ $\in \notin \mid \mathbb{U}$	**Usual meanings in set theory**
$\sum \sum\sum \prod$	**Usual meanings, summation and multiplicative**
min, max, argmax	**The usual meanings**

Symbols, notation representing numerical values

Term	**Definition**
M m, m_i	M: Available Contest Options m: Number of contest options in the list available to the voter. Discussion A contest option may be Yes or No or may be For or Against in a Referendum/Question or a specific candidate running for an elected office.
N n, n_j	N: Number of Allowed Contest Option Selections n: The number of selections allowed to be allocated by the voter in the contest and j is the jth contest option selected from j=1 to j=n: (n_j, n_{j+1}, … , n_n). Discussion: For a Yes/No Referendum/Question or a plurality single-seat contest, N would be 1. In a plurality multi-seat contest, N would equal the number of seats. In Approval Voting N would be the same as M. In Limited Voting, N would be less than the number of seats to fill.
V v, v	V_{Mi} = number of votes for the ith listing of candidate M or the ith contest option for candidate M. V_{Cii} = number of votes for the ith listing of contest option C or the ith contest option for contest option C, and S = sum of V_C for all the contest options.

Glossary

Introduction

In general, the Voting Methods Working Group Glossary and use of terms seek to preserve the terms in use and in practice in the elections field, using legislation and other authoritative election official and election administrator documents including RFPs, RFIs, manuals, and documentation as well as in use in the discourse in the field to determine and define those terms.

For the Voting Methods Working Group (VMWG) and this standard, terms in use terms of art in the field must be preserved to achieve the purpose, goals, and benefits of this standard for our stakeholders.

In summary, our purpose includes creating a more precise model of the universe of elections code (elections law) using legislation governing elections and voting, local jurisdiction rules, RFPs, RFIs and other documentation of practice that we identify as authoritative source material in elections operated by state and local government election administrators in the United States. Specifically, our purpose is to create a common and more precise model than is possible in the inherently less precise plain language algorithmic specifications that exist in the authoritative source materials.

Expected benefits of preserving terms of art and in practice include but are not limited to disambiguation and increased precision. Disambiguation and increased precision have a direct and measurable impact on metrics and Key Performance Indicators (KPI) of the efficient operations of elections processes. Costs may be reduced through the reduction of unnecessary overhead and resource use, reduction in error, and increases in efficiency and accuracy, for election administrators and for voting system certifiers. Additional examples of metrics that may be positively impacted include reduced errors in acceptance testing, decreased incidents of faults discovered in logic and accuracy testing, more accurate capacity planning. Further, adopting a standard for voting methods and tabulation methods may provide a means for greater accountability for compliance to governing law and local rules, and greater accountability for compliance with RFPs and resultant contracts for elections officials, manufacturers and service providers.

We may differ in our glossary terms or labels for objects, entities, or things in the NIST SP 1500-107 model, in a few cases, as compared to other elections and voting glossaries. We seek to minimize any difference in use of terms from the NIST VVSG Glossary. We also seek to match, as much as possible, use of terms and labels in our models with the NIST Interoperability Common Data Format (CDF) including those NIST standards or specifications that NIST SP 1500-107 adopts. To that end, standards and specifications that we adopt include NIST Cast Vote Record (CVR) CDF and NIST Election Results Reporting CDF, Election Modeling working group's elections business process model, NIST VVSG Cybersecurity working group's requirements, and enumerations that are part of the NIST Interoperability CDFs.

The adopter of NIST SP 1500-107 should assume that any elections and voting terms that are used in this specification and not defined in this Glossary are defined in the NIST VVSG Glossary.

We ask that users of this glossary credit this specification, NIST SP 1500-107, and the VMWG in any publications. Please post to the VWMG listserv with any questions or comments: voting-methods@list.nist.gov. Thank you.

Term	Definition
Batch	A collection of ballots gathered as a group for tabulation or other processing. **Technical Definition** A collection of ballots gathered as a group for tabulation processing or other operations on vote selection data sets including but not limited to post-election auditing, ballot accounting, acceptance testing, and logic and accuracy testing. **Discussion** Tabulation may be performed on a batch of ballots or ballots represented as Cast Vote Record as defined in NIST Special Publication 1500-103 Cast Vote Records Common Data Format V1.0
Batch elimination	A simultaneous defeat of multiple continuing contest options for which it is mathematically impossible to be elected or to prevail. **Discussion** If a specified candidate satisfies both of the following conditions, then all candidates with fewer votes may be designated as defeated: (1) At least one other candidate has at least as many votes as the specified candidate. (2) The specified candidate has more votes than the total votes for all candidates with fewer votes.

Contest Options	The candidates or issues on a ballot among which the voters may express a preference or support by their votes. **Technical Definition** The candidates or issues or selection options on a ballot among which the voters are to express a vote preference by marking (or deciding not to mark) their ballot. Positions on a ballot presentation, ballot layout or ballot intended to be the locations for marks to be made by a voter on that ballot. Positions on a ballot intended to be the locations for marks to be detected by an election system such as an optical scanner, tabulator, or to be adjudicated by an election official or judge.
Continuing Ballot	A ballot that will be processed in the current contest RCV round.
Continuing Candidate	See Continuing Contest Option
Continuing Contest Option	A qualified candidate, measure, issue or other contest option that has not yet been elected, approved or eliminated in the current round or instance of RCV contest processing.
Decision Rule	A decision rule is a constraint or a criterion that is used to determine an election contest outcome. **Technical Definition** A decision rule is a non-null constraint or a criterion that is used to determine an election contest outcome. A decision rule is a required element or attribute of a decision rule set.

Decision Rule Set	A decision rule set is a set of one or more decision rules. **Technical Definition** A decision rule set is a non-null and non-empty set of one or more decision rules. A decision rule set is a required element of a voting method. **Discussion** We intentionally exclude a null rule and/or null set or empty set from this definition. Clearly, this requirement does not imply that a systems implementation would be coded to be unable to properly handle the edge case of errors including a null decision rule data element. The purpose of these definitions is to define what should be so in correct and intended operation.
Exhausted Ballot	A ballot encountered in a round of RCV processing of a contest that has no further valid ranking of continuing contest options or that contains a condition in a subsequent choice that invalidates further consideration of the ballot. Proposed Equivalent term: Inactive Ballot **Discussion** The term 'exhausted ballot' is the term in use and in practice, in legislation and other authoritative elections official and election administrator documents including RFPs, RFIs, manuals, and documentation as well as in use in the discourse in the field. Therefore, this term is preserved to achieve the purpose, goals, and benefits of this standard. In Multi-Pass IRV all ballots, including exhausted ballots, are revived when beginning the count to fill a second or later seat.
Highest-ranked continuing contest option	The next preferred continuing contest option on a given ranked ballot.
Highest Continuing Ranking	The ranking on a voter's ballot with the lowest numerical value or the next highest position in sequence for a continuing RCV contest option.

Inactive Ballot	Equivalent to Exhausted Ballot. **Discussion** See also: Discussion for Exhausted Ballot. The term "Exhausted Ballot" is used extensively as a term of art in the RCV legislation, local jurisdiction artifacts, contracts, RFPs and in the field. Therefore, for the purpose of NIST SP 1500-107 we preserve this as a key existing label for the same concept.
Majority of Votes	Greater than 50-percent of the votes counted for a contest, or greater than 50-percent of the votes counted for a contest for all continuing contest options in an RCV round. **Technical Definition** majorityOfVotes = (sum(aggregate(votesContinuing)) /2) +1 **Discussion** "Majority of votes" shall mean fifty percent (50%) plus one of the votes cast on continuing ballots. (San Leandro)
Mathematically Impossible to be Elected or Prevail	A contest option in an RCV contest where the current vote total plus all votes that could possibly be transferred to it in future rounds would not be sufficient to surpass the contest option with the next higher current vote total. **Discussion** See Batch Elimination for operation.
Overvote	Occurs when the number of selections made by a voter in a contest is more than the maximum number allowed. Synonym: Over-vote **Discussion** For RCV an overvote occurs when a given ranking is selected for more than one contest option.
Ranking, RCV	The number or position selected for a contest option to indicate a voter's ranked preference for that option.

Repeated ranking	Selection of more than one ranking for the same contest option for the contest being counted. **Discussion** Sometimes referred to as Duplicate Ranking. Not to be confused with "duplicate ranking" when that term is used to refer to an overvote.
Round of counting or round	A round is a subprocess in the tabulation process during which current votes for all continuing contest options are counted in accordance with the applicable tabulation method counting rules. **Technical Definition** A round is a subprocess in the tabulation counting process during which current votes for all continuing contest options are counted in accordance with the applicable tabulation method counting rules. A round is for the purpose of determining the following: ● whether a contest option has achieved a majority or a threshold, ● whether and which contest option or contest options are to be eliminated, ● redistribution of surplus in a multi-winner contest. **Discussion** A round is for the purpose of determining the following: ● whether a contest option has achieved a majority or a threshold, ● whether and which contest option or contest options are to be eliminated, ● redistribution of surplus in a multi-winner contest.
Skipped ranking, RCV	When a voter omits a ranking and ranks a contest option at a subsequent ranking.
Surplus	The number of votes cast for a contest option in excess of the number required to meet or exceed the applicable threshold rule. **Discussion** Applies to multi-winner RCV, aka STV

Surplus fraction, Surplus factor	The proportion of each vote to be transferred when a surplus is transferred. **Technical Definition** The quotient, rounded down to n decimal places, of a contest option's surplus divided by the total number of votes for the contest option in the round in which the surplus occurs. **Discussion** Applies to multi-winner RCV, aka STV
Tabulation Method	A tabulation method is a set of process flow steps and operations for performing counting, accumulation, and other data and math operations on a vote selection data set. **Technical Definition** Tabulation methods are sets of partially ordered process steps and operations for performing counting, accumulation, and other data and math operations. Each tabulation method supports, or is a part of at least one tabulation scenario .A tabulation method is used by a voting method to produce an election contest outcome. **Discussion** See Section 3.1.3 Tabulation Methods for more information on tabulation methods and tabulation process flow diagrams NIST_SP_1500-10X_VotingMethods_3-1-3_TabulationMethods_ContestProcessingHighLevel-HD-LLEditorsDraft20180510.docx
Tabulation Scenario	A situation and sequence of events or activities intended by Elections Officials and Elections Administrators to be sufficient to achieve a tabulation goal. **Technical Definition** A tabulation scenario contains a situation and sequence of events, steps in process, and tabulation state changes. Sets of ballots in the form of Cast Vote Record data sets, proceed in steps to be operated upon according to the defined process of tabulation that is executing, with the objective of producing an output count and related statistics and reporting for a particular instance of tabulation.

	Discussion A tabulation scenario may be intended to count only a subset of the total number of ballots needed to count for an election contest result that is output by a voting method. Examples of tabulation scenario purposes include but are not limited to configuration, logic and accuracy and acceptance testing, batch, precinct tabulation, central tabulation, statewide 'rollups' or aggregations, recounts, and audits. This specification currently supports tabulation scenarios as follows. **Tabulation Scenarios** ● Batch ● Batch with Accumulation ● Election Contest ● Reporting Unit ● Aggregation ● Compositions of Batch and Contest variants with Accumulation and Aggregation ● Multi-Contest Multi Voting Method per Ballot ● Distributed variants of this list of tabulation scenarios ● Precinct ● Central tabulation ● Tabulation scenarios for the auditing use case ● Tabulation scenarios for the recount use case ● Tabulation for configuration testing ● Tabulation for logic and accuracy and acceptance testing
Threshold/Quota, RCV	The number of votes that are sufficient for a contest option to prevail. In STV, it is also the point at or above which additional votes for a contest option are considered to be surplus.
Transfer value	The transfer value of a ballot is the one vote or portion of a vote that the ballot will contribute to the vote total for the ballot's highest-ranked continuing contest option after a surplus transfer or elimination of one or more contest options. **Technical Definition** When used in surplus transfer, the product, rounded down to n decimal places, of a ballot's value multiplied by a contest option's surplus fraction.
Undervote	Occurs when the number of voter selections in a contest is less than the maximum number allowed for that contest or when no selection is made. The number of undervotes is equal to the number of votes lost, for example, if no selection is made in a vote for two contest the number of votes lost is two.

	Synonyms: under-vote https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf
Undervote, RCV	An undervote occurs when a voter does not rank any contest options in a ranked choice voting contest.
Voting Method	A Voting Method encompasses a set of process flow or algorithmic steps, tabulation methods, and a decision rule set. Each voting method contains at least one tabulation method and at least one decision rule set containing at least one decision rule. The decision rule set is used by the voting method to produce an election contest outcome. **Discussion** Executing the appropriate election contest decision rule set is in the scope of the voting method. The labels by which voting methods are known are elements of the NIST Common Data Format (CDF) enumeration vote variation. Voting methods supported by this specification include the following. ● Plurality ● Majority ● Super-Majority ● N-of-M (Includes the case of 1-of-M) ● Ranked Choice Voting (RCV) ● Approval ● Cumulative ● Proportional ● Range See Section 3.1.3 Voting Methods for more information on voting methods and process flow diagrams.
Voting Method Variant	A voting method variant is a particular, more specific variety of a baseline voting method where a particular tabulation method, and/or a more specifically or differently constraining decision rule set are applied to produce the election contest outcome.
Vote Variation or VoteVariation	Vote variation is an enumeration defined in NIST SP 1500-103 Cast Vote Record (CVR) Version 1.0, Section 4.26, CDF, that contains String labels by which voting methods are known as its elements.

	Discussion **Not to be confused with the term "Voting Variation"in the NIST Glossary** **https://collaborate.nist.gov/voting/bin/view/Voting/VotingVariation** The VoteVariation enumeration is an element of the UML model that specifies the CVR Common Data Format (CDF). **See** NIST SP 1500-103 Cast Vote Record Version 1.0, Section 4.26 (page 61) **https://github.com/usnistgov/CastVoteRecords/blob/master/CVR%20Specification-2018-07-17.pdf**
Vote	A ballot choice that is counted toward the outcome of a contest. **Discussion** [See also: vvsg Contest Option Vote, previously was called Valid Vote https://github.com/HiltonRoscoe/GlossaryMD/blob/master/vvsg_living_glossary.md#contest-option-vote] As applied to RCV, the selection(s) may or may not contribute to the official vote total of a contest option in the final round of tabulation. If all selected contest options are eliminated, the ballot would be considered "inactive' with respect to the contest being counted. Never-the-less, an inactive ballot is still considered a "vote," having been counted for a contest option during the tabulation process. This is no different that a "vote" cast for a losing contest option on any valid ballot.

Appendix B: Reference Implementation, RCV Tabulator

RCTab f/k/a the Universal RCV Tabulator

RCTab (formerly the Universal RCV Tabulator) is offered by the Ranked Choice Voting Resource Center[1] as a reference implementation of the RCV tabulation methods variants specified in NIST 1500-107. NIST 1500-107, Voting and Tabulation Methods, adopts the NIST Common Data Format (CDF) standard NIST 1500-103, Cast Vote Records (CVR). This RCV Tabulator reference implementation thereby adopts and implements NIST 1500-103 CVR.

RCTab is a free, open-source application designed to efficiently and accurately tabulate ranked choice voting elections using a variety of rules commonly in use in North America. It is available under the GNU Affero General Public License v. 3.0 (GNU AGPL v3.0). The principal authors are Hylton Edingfield, Louis Eisenberg, and Jon Moldover, three members of a group called Bright Spots based in San Francisco.

In July, 2019, the Ranked Choice Voting Resource Center submitted the RCV Tabulator, v. 1.0.1, for testing to a Voting Systems Testing Laboratory (VSTL). On August 30, 2019, the VSTL provided a letter-report of compliance stating: "Based on the testing performed and the results obtained, the Universal RCV Tabulator solution identified in this letter-report is believed to meet the requirements set forth by the VVSG 1.0." A copy of the VSTL report follows the description of the application.

In May, 2020, the Ranked Choice Voting Resource Center submitted an update of the RCV Tabulator to a VSTL. On May 20, 2020, the VSTL provided a letter-report of compliance stating: "Cast Vote Records (CVR) that were generated by the baseline system were imported into the RCV-Tabulator and processed through tabulation to report generation. In addition, ES&S CVR's were successfully converted to CDF and results were accurately tabulated. No system issues were noted during testing." The report states further, "Based on the testing performed and the results obtained, the modified RCV-Tabulator v1.1.0 solution identified in this test report is believed to meet the applicable requirements set forth by the VVSG 1.0." A copy of this VSTL report also follows the description of the application.

In October 2020, the RCVRC submitted version 1.2.0 of the tabulator to a VSTL. On December 4, 2020, the VSTL provided a letter that found version 1.2.0 "meets the requirements set forth by the VVSG 1.0." Version 1.2.0 was later subject to additional VSTL testing before being approved for use in New York City's RCV elections. Following that additional testing, version 1.2.0 was certified for use in New York on May 25, 2021. Version 1.2.0 has also been approved for use in Utah, Virginia, and Maine.

In 2023, Clear Ballot incorporated version 1.3.1 of RCTab into their election management system. Clear Ballot then submitted ClearVote version 2.5 (including RCTab for RCV tabulation) to VSTLs for EAC certification testing and for Oregon certification. In August 2024, ClearVote 2.5 was certified for use in Oregon elections, including RCTab for use in RCV elections.

Also in 2023, Hart submitted their voting system to the California Secretary of State for approval. That submission included RCTab 1.3.1 running on a standalone workstation. Following a first round of testing,

[1] The Ranked Choice Voting Resource Center is a business name of the Election Administration Resource Center, a 501(c)(3) nonprofit corporation.

updates were made to RCTab and version 1.3.2 was submitted for approval in November 2023. Version 1.3.2 was then certified as part of Hart's system on April 22, 2024.

RCTab has been used in the following jurisdictions and tabulation scenarios:

Official Elections		
RCTab Ver. 1.2.0		**RCTab Ver. 1.3.1**
Utah County, UT	State of Maine (beginning Nov. 2024 - previously used ES&S software)	Portland, OR/Multnomah County, OR
New York City, NY	Portland, ME	**RCTab Ver. 1.3.2**
Arlington County, VA	Westbrook, ME	Redondo Beach, CA (upcoming March 2025)
Private/Party-Run Primaries		
Arlington County Democrats "Firehouse" Primary May 2020	University of Houston Student Elections	
County/district-level delegate apportionment		
Alaska Democratic Presidential Primary 2020	Kansas Democratic Presidential Primary 2020	Wyoming Democratic Presidential Primary 2020
Testing and Tabulation Verification		
Minneapolis, MN	State of Maine (pre-Nov. 2024)	Benton County, OR

The application, source code and all documentation may be accessed at https://github.com/BrightSpots/rcv/.

Input formats accepted:

1. Cast vote record files
    a. NIST 1500-103 CVR CDF
    b. Clear Ballot .csv CVR export files
    c. Dominion .csv and .json CVR export files
    d. ES&S .xlsx CVR export files[2]
    e. Hart .csv CVR export files
2. Configuration files
    a. RCTab .json configuration files

[2] VSTL tested to applicable VVSG 1.0 and VVSG 1.1 standards as of August 30, 2019.

Overview

RCTab is a free, open-source application developed by Bright Spots, a private public interest group based in San Francisco, in conjunction with the Ranked Choice Voting Resource Center (rankedchoicevoting.org) for the purpose of enabling rapid and accurate processing of a wide variety of ranked choice voting (RCV) elections.

It allows users to:

- Create contest configuration files using a graphical user interface (GUI)
- Validate contest configuration files to ensure they will function with the tabulator and that all selections are valid within the confines of the software
- Process the outcome of a ranked choice voting contest

A contest configuration file specifies:

- Which ranked choice voting rule variations to use
- A list of candidates
- File paths to one or more cast vote record (CVR) files
- Output formatting options (contest name, date, jurisdiction, etc.)

The Tabulator produces the following output:

- A summary .csv file containing round-by-round vote totals for each candidate and the apparent winner(s)
- A summary .json file containing information included in the .csv file and additional information which can be used by external tools for visualizing contest results
- A detailed audit .log file describing how every ballot was counted in each round over the course of the tabulation
- A comprehensive post-processing CVR in the NIST 1500-107 and NIST 1500-103 JSON common data format (CDF).

CVR Input files supported formats include the NIST 1500-107 and NIST 1500-103 JSON format for CVRs as well as the Excel format produced by ES&S.

The configuration options available are shown below and include all rule variants currently used or proposed to be used in existing jurisdictions where RCV is used or proposed for use.

Contest Info Tab

Below is what the contest info tab looks like when a user first navigates to it. This is also the first screen a user will see upon successfully launching RCTab. Contest info fields are in the middle of the screen. The black box at the bottom of the screen is called the operation log box. It sends the user messages about the tabulation process, any errors encountered in using a configuration file, any errors in tabulation, and other software errors. Information in this panel is saved to the rcv_0.log file. Users cannot turn this feature off while using the RCTab software. The panel on the right side of the document is the “Hints" tab which includes any interaction with RCTab, including producing configuration files, running tabulations, hashing results files, and transmission of files from RCTab on USB drives should follow transmission procedures required in the jurisdiction, including the use of a team with no less than two trained personnel.

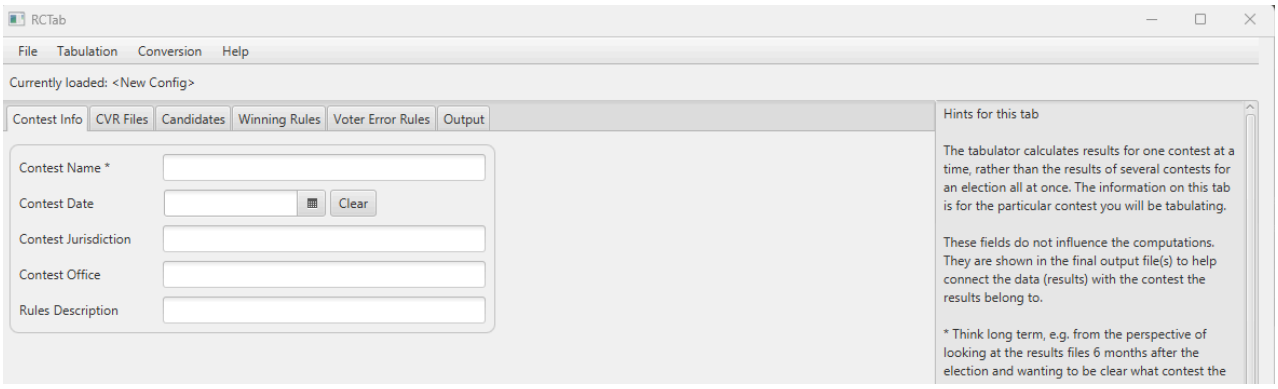


Figure 17. RCTab Tab 1: Contest Info.

These settings appear on the "Contest Info" tab in RCTab.

- Contest Name
- Contest Date
- Contest Jurisdiction
- Contest Office
- Rules Description

The tabulator calculates results for one contest at a time, rather than the results of several contests for an election all at once. The information on this tab is for the particular contest you will be tabulating.

These fields do not influence the computations. Contest Name, Contest Date, Contest Jurisdiction, and Contest Office are shown in the final output file(s) to help connect the data (results) with the contest the results belong to.

- Think long term, e.g. from the perspective of looking at the results files 6 months after the election and wanting to be clear what contest the results belong to.

- You may find it helpful to revisit this tab once you have done a few test runs and see what the output looks like.

Contest Name (required): Enter a name to identify it.

- Examples: City Council 2018, Board of Election Ward 13 2017, Mayor, Referendum 289b

Contest Date (optional): The date on which the election for this contest was run.

- Clear permits user to clear this field
- The calendar button allows a user to navigate through a calendar to select the date of the contest

Contest Jurisdiction (optional): E.g.: Minneapolis, Eastpointe

- Whether this is helpful may depend on what you put into the Contest Name field

Contest Office (optional): E.g.: Mayor, County Clerk

- Whether this is helpful may depend on what you put into the Contest Name field

Rules Description (optional): What short description of this configuration would help you remember in, say, six months what election this specific rule configuration is for? This option's use impacts no other options. It is included in configuration .json files and audit .log files.

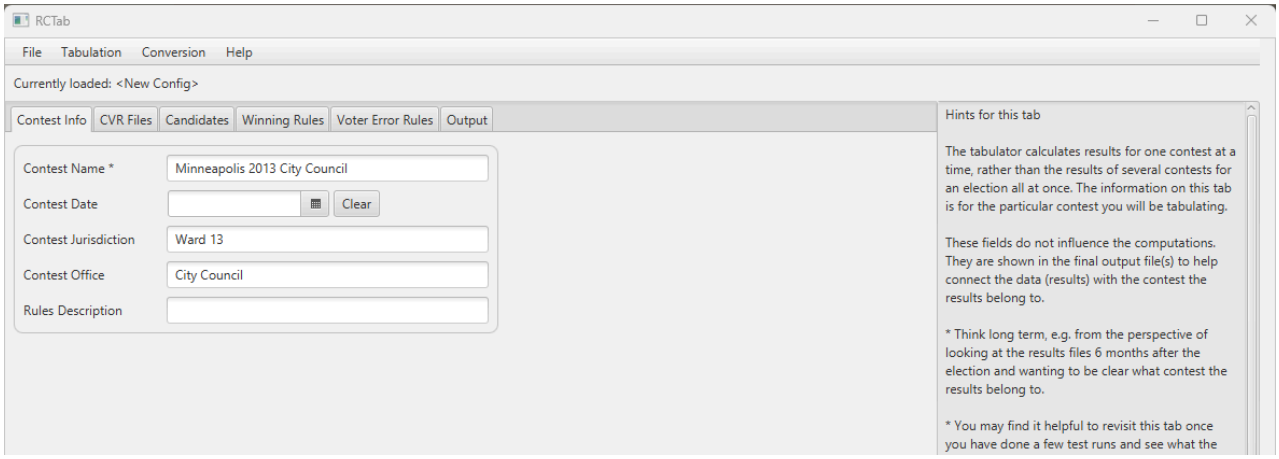


Figure 18: Example Completed Contest Info tab

CVR Files Options

This is what the CVR Files tab looks like when a user first navigates to it:

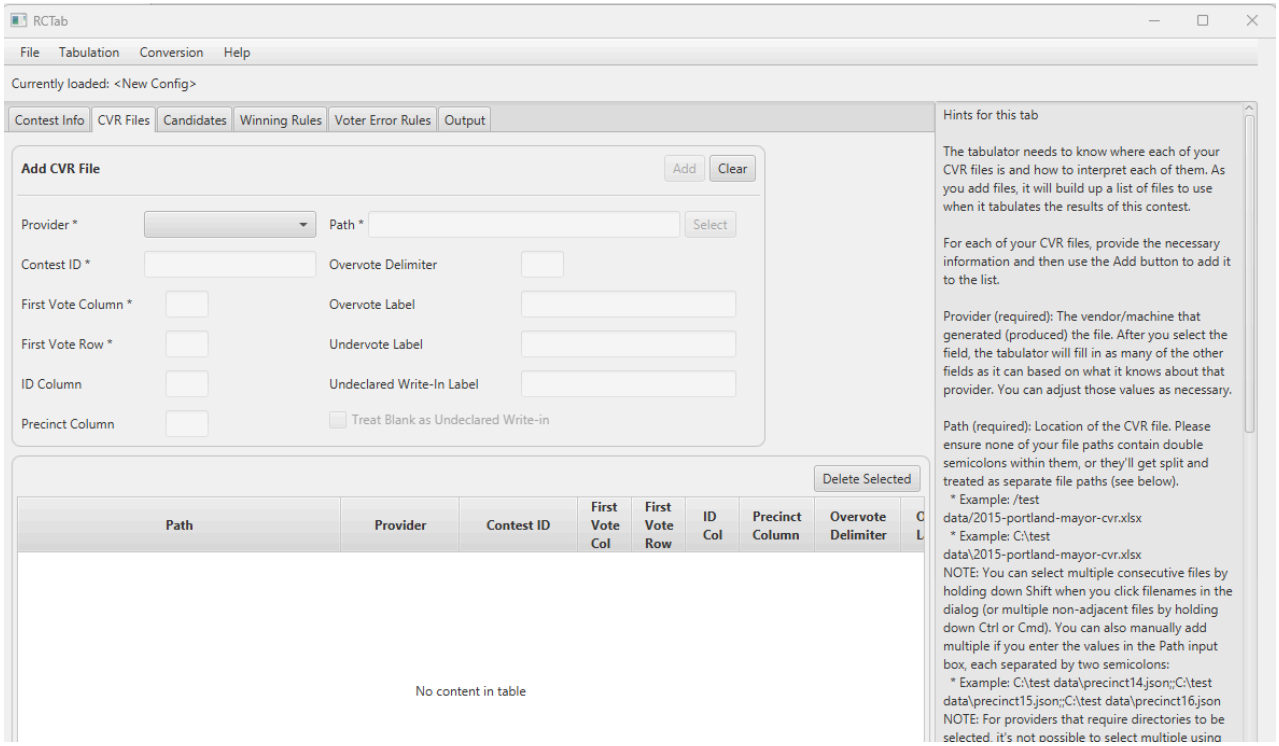


Figure 19: RCTab Tab 2: CVR Files

The tabulator needs a configuration file laying out where each of its user's CVR files for the contest to be tabulated are and how to interpret each of them. Only add files to the configuration that contain data for the contest you are tabulating. CVRs that do not include data for the contest to be tabulated will cause tabulation to fail. Fields used to describe CVRs are on the top half of the tab, while the table on the bottom half of the tab will display each CVR file you have added to your configuration file so far. All information in this tab directs RCTab on where to find files and how to read those files, it does not actually pull in any vote information. Vote information is only used after a user clicks the "Tabulate" button under "Tabulation" as described later in this guide.

For each of your CVR files, provide the necessary information and then use the Add button to add it to the list.

Provider (required): The vendor/machine that generated (produced) the file. After you select the field, the tabulator will fill in as many of the other fields as it can, based on what it knows about that provider. You can adjust those values as necessary. Different options are active for different providers. See below for a break-down of each field and what is required:

Path (required)**:** Location of the CVR file.

- Example: /Users/test data/2015-portland-mayor-cvr.xlsx
- Select allows a user to navigate using Windows Explorer to select a location.

Contest ID: Some CVRs assign an ID label to each contest in the CVR. The tabulator needs to know which contest is being tabulated when multiple contests are included in one CVR. Enter the ID of the contest being tabulated in this field.

First Vote Column: the column where the first vote record is.

First Vote Row: the row where the first vote record is.

ID Column: The column the IDs are in. Not all CVR files contain an ID column.

Precinct Column: The column that contains the precinct.

Overvote Delimiter (optional, but must be blank if "Overvote Label" is provided)**:** If using a CVR in ES&S style, overvotes can be reflected in a CVR by displaying all candidates marked at a ranking. Those candidate names will be differentiated from each other by a delimiter, something like a vertical bar | or a slash /. If your overvotes are delimited like this, enter the delimiter used in this field. Note: that ES&S files may include only the label "overvote" and no additional information, in which case the "Overvote Label" field should be used instead.

Overvote Label (optional, but must be blank if Overvote Rule is)**:** Some CDF and ES&S CVRs use a particular word/phrase to indicate an overvote.

Undervote Label (optional)**:** Some ES&S CVRs use a particular word/phrase to indicate an undervote.

Undeclared Write-in Label (optional)**:** Some CVRs use a particular word/phrase to indicate a write-in.

Treat Blank as Undeclared Write-in (optional for ES&S)**:** When checked, the tabulator will interpret blank cells in this ES&S CVR as votes for undeclared write-ins.

Add: Adds CVR to the configuration file. Impacts no other option. Operation impacted by whether required fields (depending on Provider selected) are filled in.

Clear: Clears all fillable values in CVR Files tab above the CVR Files table.

Delete Selected: Deletes CVR file information listed in the CVR table. Impacted by and impacts no options.

This guide will now briefly show which provider settings permit users to edit which additional settings.

- Provider options:
    - CDF
    - Clear Ballot
    - Dominion
    - ES&S
    - Hart

CDF

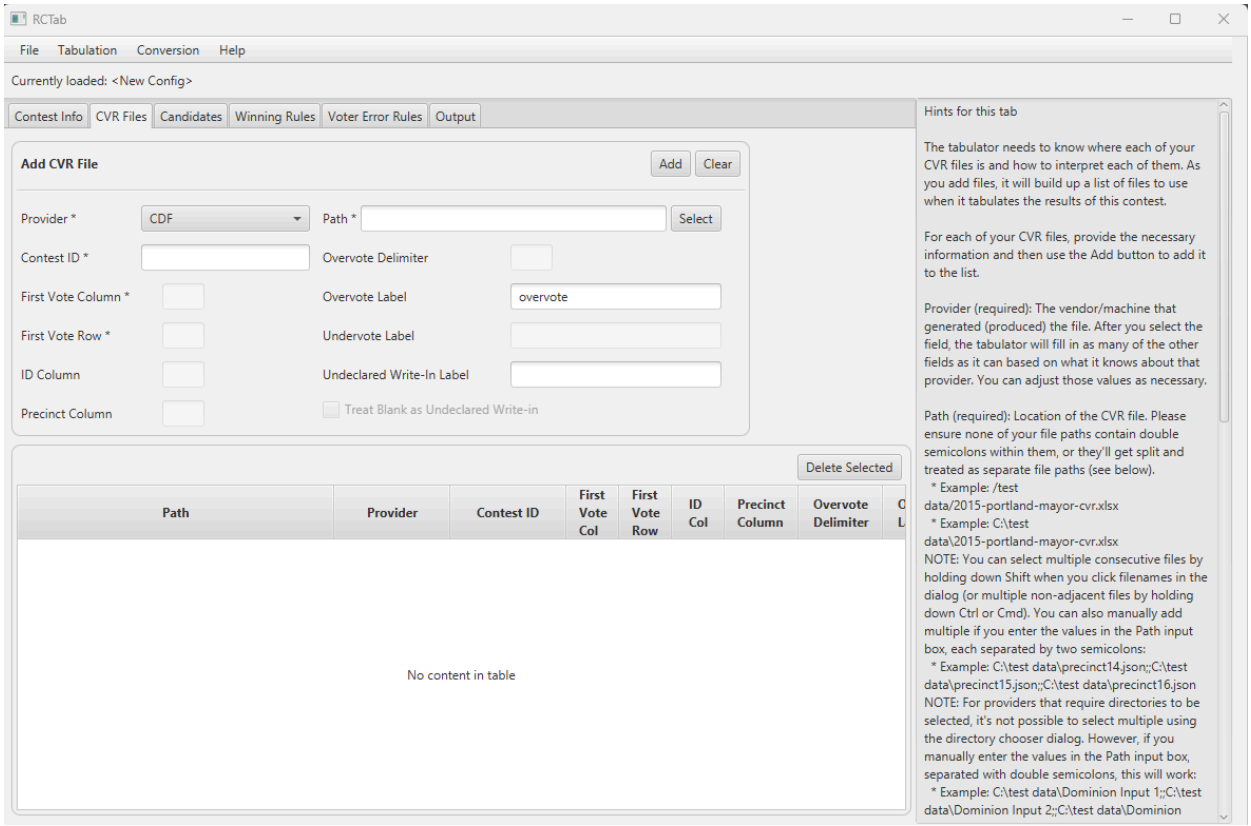


Figure 20: RCTab Tab 2: CVR Files | CDF

Permits user to edit these options:	**Does not permit user to edit these options:**	
Path	First Vote Column	Overvote Delimiter
Contest ID	First Vote Row	Undervote Label
Overvote Label (Default value: overvote)	ID Column	Treat Blank as Undeclared Write-In
Undeclared write-in label	Precinct Column	

Clear Ballot

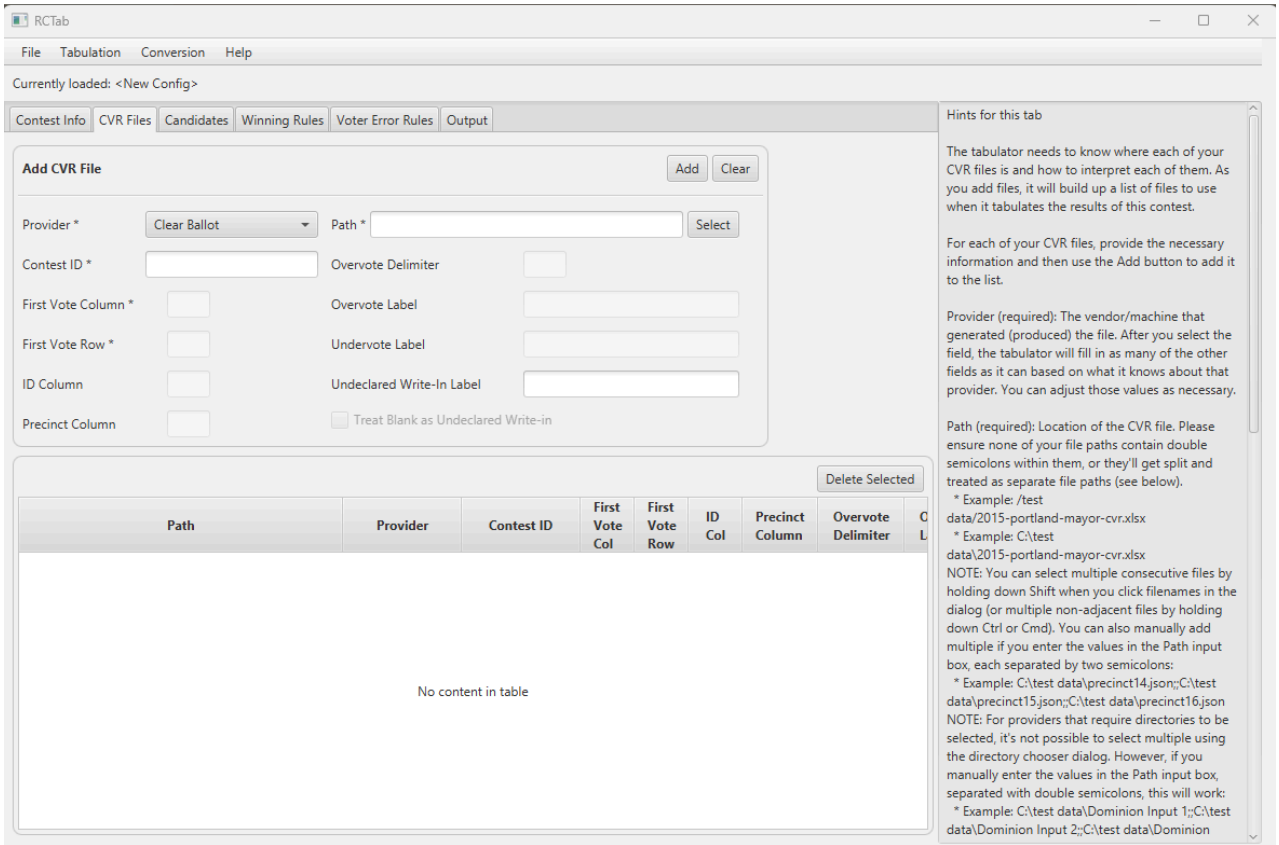


Figure 21: RCTab Tab 2: CVR Files | Clear Ballot

Permits user to edit these options:	**Does not permit user to edit these options:**	
Path	First Vote Column	Overvote Delimiter
Contest ID	First Vote Row	Overvote Label
Undeclared Write-In Label	ID Column	Undervote Label
	Precinct Column	Treat Blank as Undeclared Write-In

Dominion

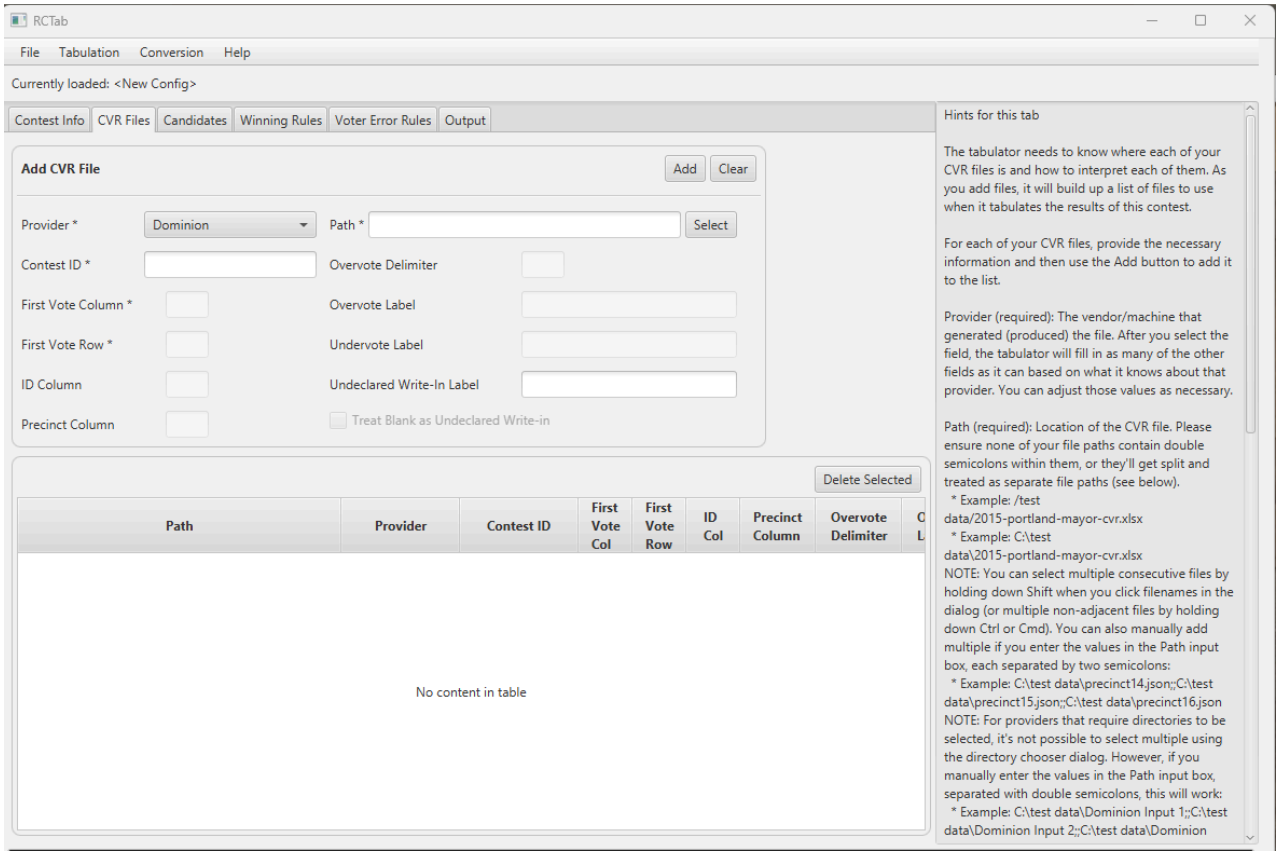


Figure 22: RCTab Tab 2: CVR Files | Dominion

Permits user to edit these options:	**Does not permit user to edit these options:**	
Path	First Vote Column	Overvote Delimiter
Contest ID	First Vote Row	Overvote Label
Undeclared Write-In Label	ID Column	Undervote Label
	Precinct Column	Treat Blank as Undeclared Write-In

ES&S

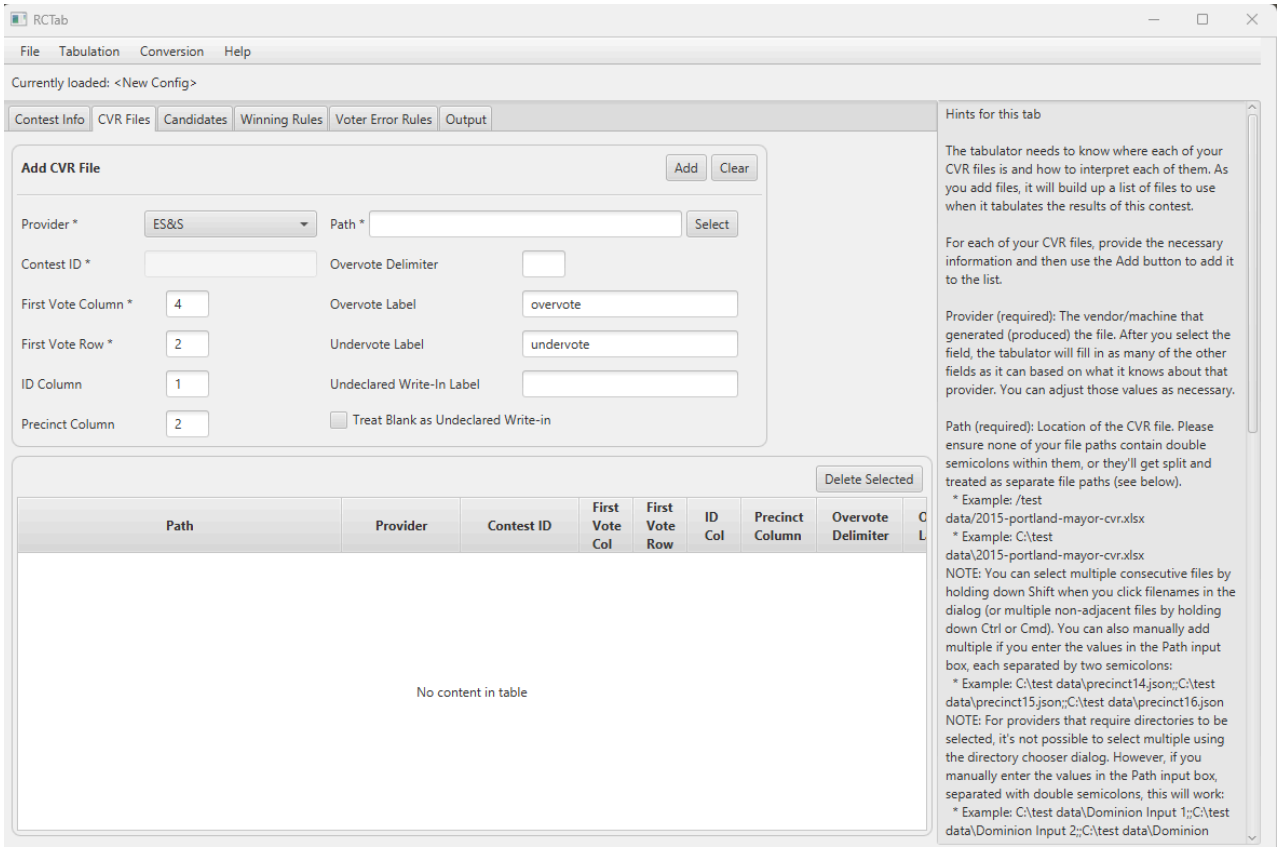


Figure 23: RCTab Tab 2: CVR Files | ES&S

Permits user to edit these options:		**Does not permit user to edit these options:**
Path	Overvote Delimiter	Contest ID
First Vote Column (Default Value: 4)	Overvote Label (Default Value: overvote)	
First Vote Row (Default Value: 2)	Undervote Label (Default Value: undervote)	
ID Column (Default Value: 1)	Undeclared Write-In Label	

Precinct Column (Default Value: 2)	Treat Blank as Undeclared Write-In

It is assumed that the ES&S export uses the default settings so below are the ES&S default settings. The user will only need to enter the path, but all other default settings are included. If the default settings are not used in the ES&S export file, the user must determine the values to be used. If defaults are not being used, users should confirm the proper values for First Vote Column, First Vote Row, ID Column, Precinct Column, Overvote Label, Undervote Label, and Undeclared Write-In Label by referencing the CVRs files to be used in the Tabulation.

The First Vote Column location depends on the information users choose to export in their CVR exports from ES&S's system. ES&S systems can include individual CVR ID information, Precinct information, and ballot style label information in the first three columns of a CVR, as shown in the below screenshot. Column A (aka Column 1) has CVR ID numbers; Column B (Column 2) has Precinct information, Column C (3) has ballot style information, and Column D (4) includes the first ranking in the RCV contest in this CVR. This is the standard CVR layout for ES&S. More information is available from ES&S.

A	B	C	D
Cast Vote Record	Precinct	Ballot Style	Rep. to Congress 1st Choice District 2
1	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
2	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
3	Fayette	CAN Ballot Style 130	DEM Golden, Jared F.
4	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
6	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
8	Fayette	CAN Ballot Style 130	Hoar, William R.S.
10	Fayette	CAN Ballot Style 130	undervote
15	Fayette	CAN Ballot Style 130	DEM Golden, Jared F.
17	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
19	Fayette	CAN Ballot Style 130	REP Poliquin, Bruce
20	Fayette	CAN Ballot Style 130	Hoar, William R.S.

Figure 24: ES&S Example CVR File.

CVR exports from ES&S contain columns for each ranking in an RCV contest. See the above screenshot for an example. This screenshot includes the five rankings voters had in the contest, arranged in sequential order. This CVR export included only data for the one RCV contest to be run. The setting "Maximum Number of Candidates That Can Be Ranked" (covered below in Winning Rules) instructs RCTab how many columns to process when running the round-by-round count. If that setting is set to five and the First Vote Column setting is set to four (Column D), RCTab will process this CVR starting at Column D/4 (column number 4) and continue through columns E/5, F/6, G/7, and H/8 - five columns for the five rankings voters have.

Hart

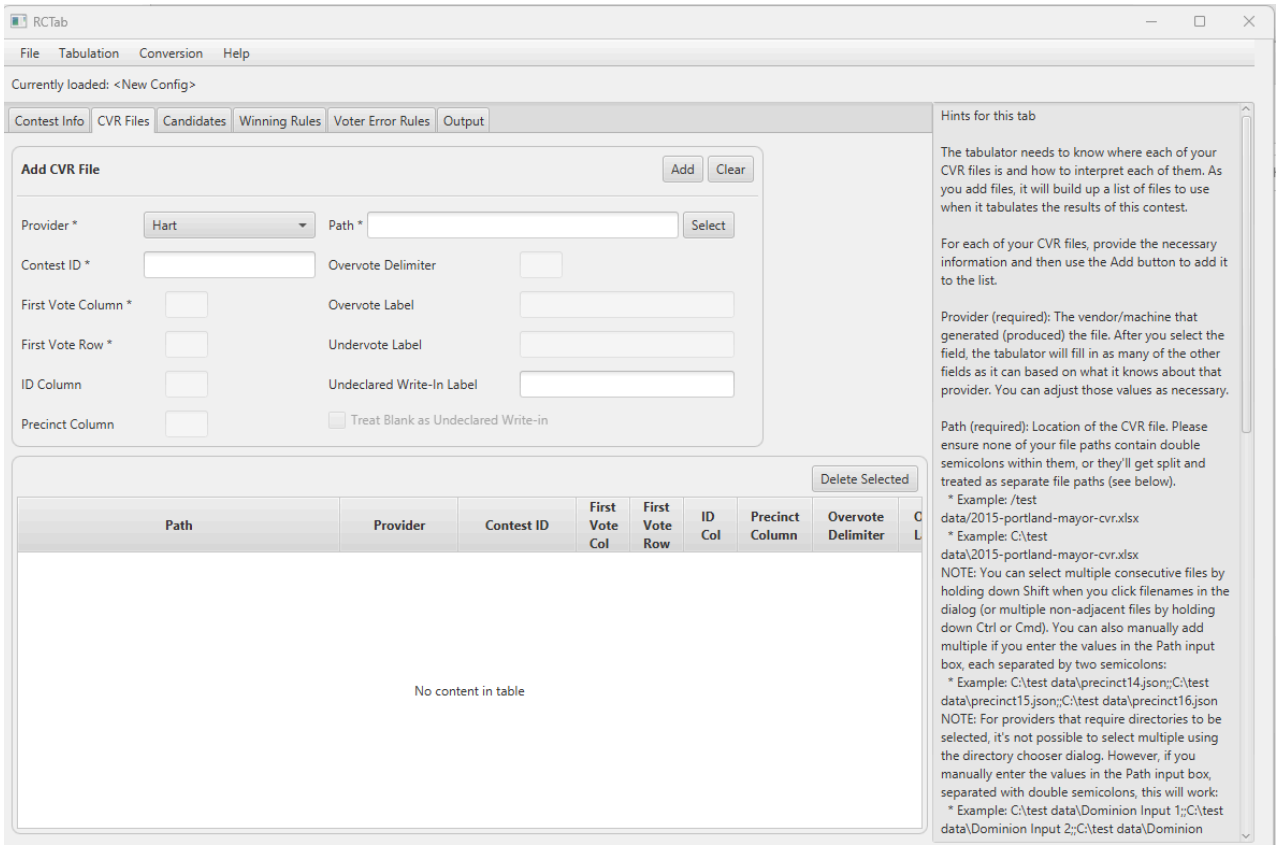


Figure 25: RCTab Tab 2: CVR Files | Hart

Permits user to edit these options:	**Does not permit user to edit these options:**	
Path	First Vote Column	Overvote Delimiter
Contest ID	First Vote Row	Overvote Label
Undeclared Write-In Label	ID Column	Undervote Label
	Precinct Column	Treat Blank as Undeclared Write-In

Path: Select the path of the folder that contains signed Hart CVRs.

Candidates Tab

The candidates tab allows users to put in information about how candidates are referred to in cast-vote record files. These settings impact how CVR files are read. Candidate names entered on this tab will also be used to display candidate names in results files. Below is a screenshot of the Candidates Tab when a user first navigates to it.

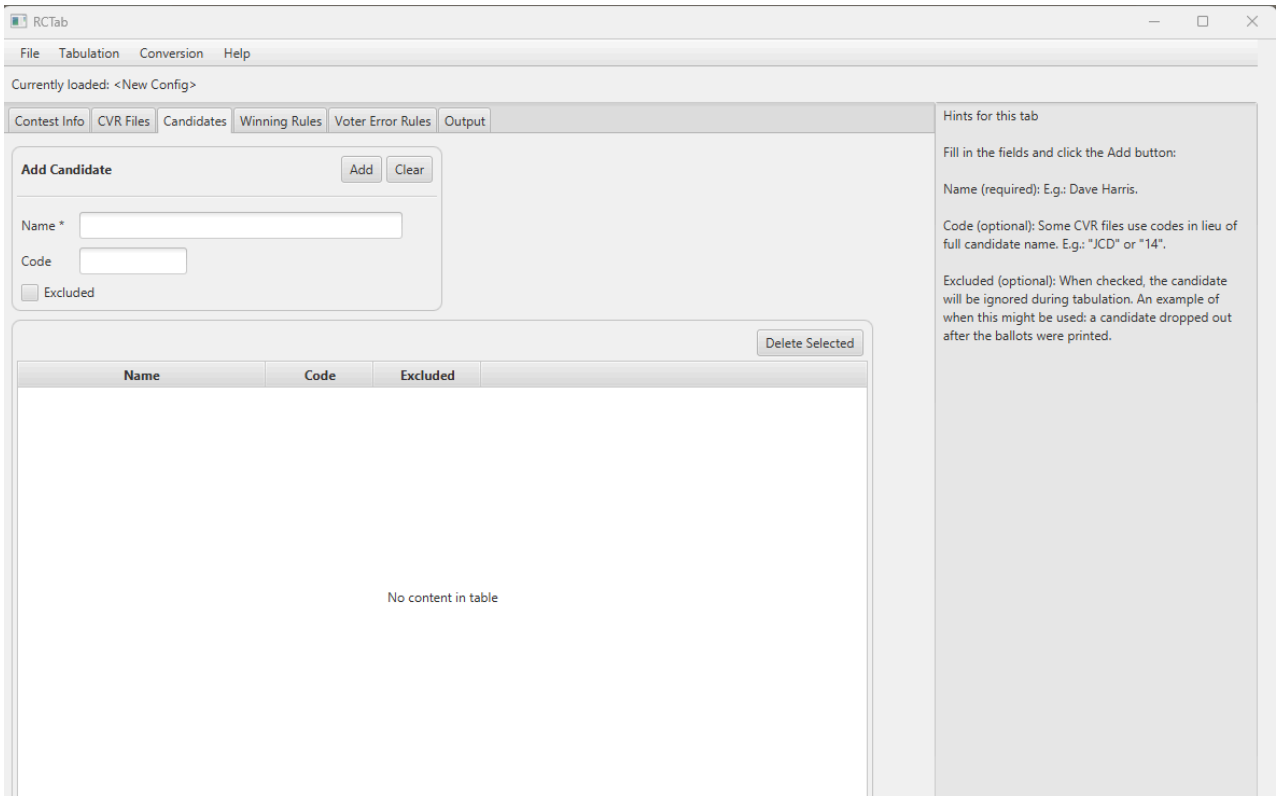


Figure 26: RCTab Tab 3: Candidates

Fill in the fields and click the Add button:

Name (required)**:** E.g.: Dave Harris. This information is used to display candidate names in results files.

Code (optional)**:** Some CVR files use codes in lieu of the full candidate name. e.g.: "JCD" or "14".

Excluded (optional)**:** When checked, the candidate will be ignored during tabulation. An example of when this might be used: a candidate dropped out after the ballots were printed.

Add: Adds Name, code, and/or excluded information to the Candidates Table.

Clear: Clears any information in Name, Code, and Excluded.

Delete Selected: Deletes selected candidate data from the Candidate Table.

Below is a screenshot of an example of a Candidates Tab while being filled out:

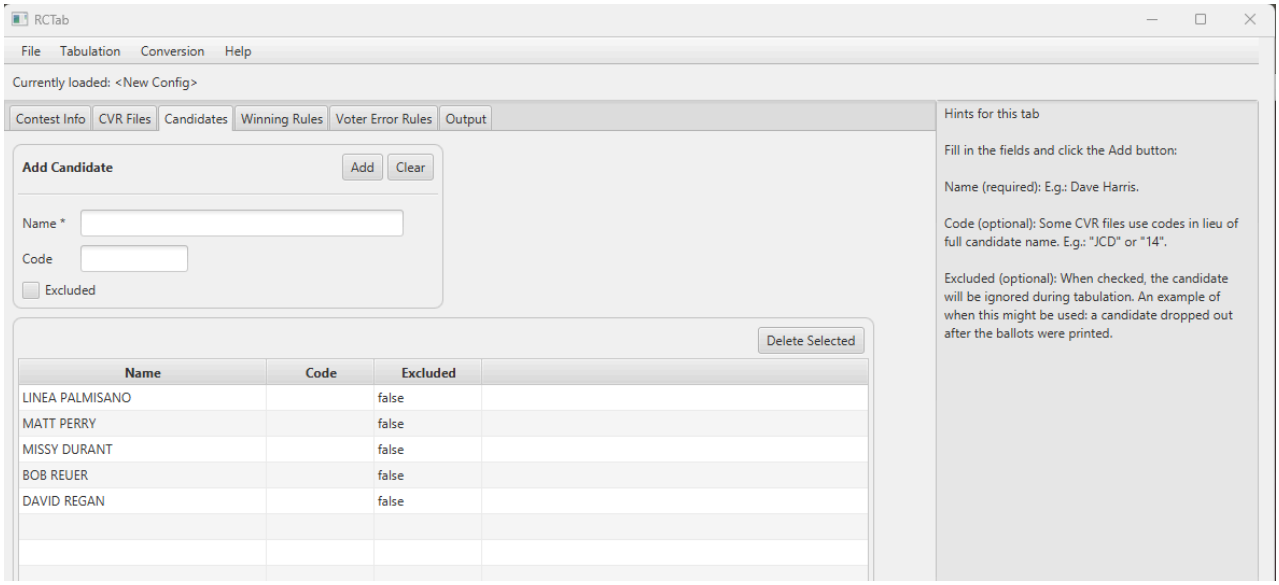


Figure 27: Example Complete Candidates tab

Winning Rules Options

Winning Rules options tell RCTab what kind of ranked choice voting election to run and how to handle details required for each kind of ranked choice voting RCTab can run.

This is a screenshot of what winning rules look like when a user first navigates to them.

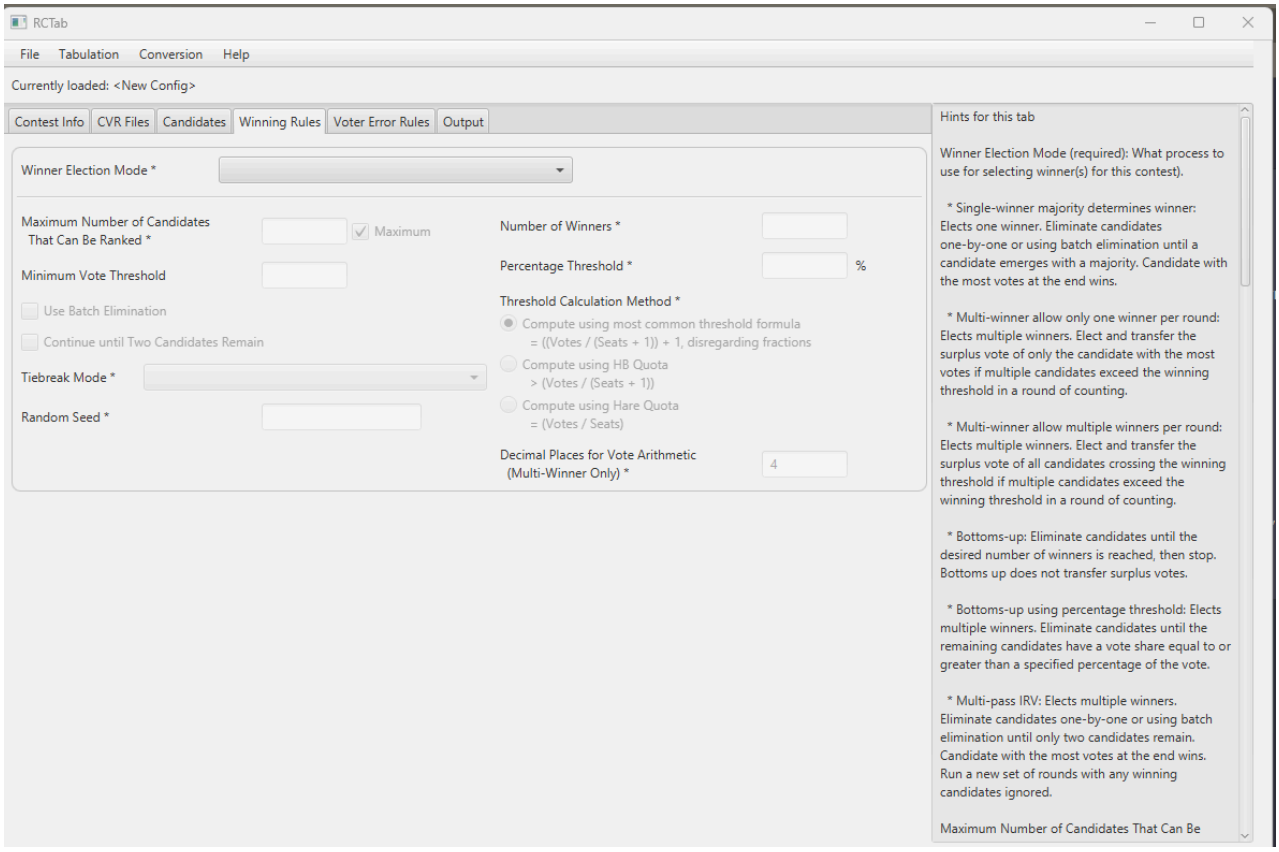


Figure 28: RCTab Tab 4: Winning Rules

There are two main sets of options on the Winning Rules tab: Winner Election Mode options and Tiebreak Mode options.

Winner election mode options have many options that interact in many different ways. Which options are required and which options users can edit ultimately depend on the winner election mode option itself. Options are described below. Screenshots are then provided of each interface for each winner election mode.

Winner Election Mode (required)**:** What process to use for selecting winner(s) for this contest.

- **Single-winner majority determines winner:** Elects one winner. Eliminate candidates one-by-one or using batch elimination until a candidate emerges with a majority. Candidate with the most votes at the end wins.
- **Multi-winner allows only one winner per round:** Elects multiple winners. Elect and transfer the surplus vote of only the candidate with the most votes if multiple candidates exceed the winning threshold in a round of counting.
- **Multi-winner allows multiple winners per round:** Elects multiple winners. Elect and transfer the surplus vote of all candidates crossing the winning threshold if multiple candidates exceed the winning threshold in a round of counting.
- **Bottoms-up:** Eliminate candidates until the desired number of winners is reached, then stop. Bottoms up does not transfer surplus votes.
- **Bottoms-up using percentage threshold:** Elects multiple winners. Eliminate candidates until the remaining candidates have a vote share equal to or greater than a specified percentage of the vote.
- **Multi-pass IRV:** Elects multiple winners. Eliminate candidates one-by-one or using batch elimination until only two candidates remain. Candidate with the most votes at the end wins. Run a new set of rounds with any winning candidates ignored.

Maximum Number of Candidates That Can Be Ranked (required)**:** How many rankings each voter has in this contest. This tells RCTab how many rankings to read from each CVR. This option can be filled out in the text box or by selecting a checkbox. Selecting the checkbox disables the text box option and inserts the value “maximum”. The checkbox can be unselectedDefault value: Maximum. Available in all winner election modes.

Minimum Vote Threshold (optional)**:** The number of first-choice votes a candidate must receive in order to remain in the race. Any candidates falling below the minimum vote threshold are eliminated and have their votes transferred. Most jurisdictions do not set a minimum vote threshold. Note that if no candidate exceeds the minimum vote threshold, vote tabulation will silently fail. Be sure not to set a minimum vote threshold beyond the number of votes candidates have. This option can impact how any winner election mode tabulation runs. Its validity is not impacted by any other option. Available in all winner election modes.

Use Batch Elimination (optional)**:** Batch elimination, or simultaneous elimination of all candidates for whom it is mathematically impossible to be elected, eliminates all candidates who cannot receive enough votes to surpass the candidate with the next highest number of votes. Example: in a six candidate contest with 200 votes, Candidate A has 80 votes, Candidate B has 70, and the other four combined have 50. Because those four candidates can never combine their votes to surpass Candidate B, they can be batch eliminated. This option impacts how single-winner majority determines winner tabulation runs. Its validity is impacted by the winner election mode. It will not impact who wins the election. Available only when Winner Election Mode is "Single-winner majority determines winner".

Continue until Two Candidates Remain (optional)**:** Single-winner ranked choice voting elections can stop as soon as a candidate receives a majority of votes, even though 3 or more candidates may still be in the race. Selecting this option will run the round-by-round count until only two candidates remain, regardless of when a candidate wins a majority of votes. It will not impact who wins the election. Available only when Winner Election Mode is "Single-winner majority determines winner".

Number of Winners (required)**:** The number of seats to be filled in the contest. This option impacts the calculation of the election threshold. Its validity and editability is impacted by the winner election mode. It sets the value for “S” in the calculations described in the UI/configuration file parameters. Available and required for Multi-winner allow only one winner per round, Multi-winner allow multiple winners per round, Bottoms-up, and Multi-pass IRV. Set to 1 by RCTab for Single-winner majority determines winner.

- Note that multi-pass IRV does not use this value to calculate its election threshold. The election threshold in multi-pass IRV is always calculated using an S value of 1. This number is used in multi-pass IRV to determine how many instances of a single-winner majority determines winner tabulation must be run on CVR files. See Configuration Parameters for more details.

Percentage Threshold: The share of votes a candidate must have in order to win. Used to calculate the election threshold in “Bottoms-up using percentage threshold.” Candidates falling below this threshold are eliminated one-by-one beginning with the candidate with the fewest votes. Required and Available only when Winner Election Mode is "Bottoms-up using percentage threshold".

Threshold Calculation Method: The threshold of election is the number of votes a candidate must receive in order to win the election. There are three primary ways to calculate the threshold of election in multi-winner RCV contests. This will be set in law (either by statute or regulation) in your jurisdiction. Required and available only when Winner Election Mode is "Multi-winner allow only one winner per round" or "Multi-winner allow multiple winners per round".

- **Compute using most common threshold formula:** The most common threshold formula is calculated by dividing the number of votes by the number of seats plus one, then adding one to that number. Fractions are disregarded. This is also known as the Droop quota. Candidates must receive this number of votes (or more) to win. This is the default threshold calculation.
- **Compute using HB Quota**: The HB, or Hagenbach-Bischoff, Quota divides the number of votes by the number of seats plus one, leaving fractions. Candidates must receive more than this number of votes to win.
- **Compute using Hare Quota**: The Hare quota divides the number of votes by the number of seats. Fractions are disregarded. It requires candidates to receive that number of votes (or more) to win.

Decimal Places for Vote Arithmetic (Multi-Winner Only): Sets how many decimal places after the decimal point are used in surplus transfers and in calculating the threshold. Its validity and

editability is impacted by the winner election mode. Required and available only when Winner Election Mode is "Multi-winner allow only one winner per round" or "Multi-winner allow multiple winners per round". Default value: 4.

Tiebreaking

There are six tiebreak modes (described in technical detail in **Section 25 - Configuration File Parameters**). Tiebreak modes impact the validity/operation of one other option: random seed. Tiebreak mode is required for all winner election modes.

Tiebreak Mode (required for all winner election modes)**:** Ties in ranked choice voting contests can occur when eliminating candidates or when electing candidates. All winner election modes may have a tie for last place. Tiebreak mode breaks the tie and determines which candidate loses the tie. Multi-winner allows only one winner per round contests can have ties between candidates who have both crossed the threshold of election; in that case, ties are broken to determine whose surplus vote value transfers first. Tiebreak procedures are set in law, either in the ranked choice voting law used in your jurisdiction or in the elections code more generally. Select the option from this list that complies with the law and procedure in your jurisdiction.

- **Random:** Randomly select a tied candidate to eliminate or, in multi-winner allow only one winner per round contests only, elect. Requires a random seed.
- **Stop counting and ask:** Pause count when a tie is reached. The user is prompted to select any tied candidate to eliminate or, in multi-winner allow only one winner per round contests only, elect.
- **Previous round counts (then random):** The tied candidate with the least votes in the previous round loses the tie. If there is a tie in the previous round, the tie is broken randomly. Requires a random seed.
- **Previous round counts (then stop counting and ask):** The tied candidate with the least votes in the previous round loses the tie. If there is a tie in the previous round, the user is prompted to select any tied candidate to eliminate or, in multi-winner allow only one winner per round contests only, elect.
- **Use candidate order in the config file:** Use the order of candidates in the config file to determine tiebreak results. Candidates lower in the list lose the tiebreaker.
- **Generate permutation:** Generate a randomly ordered list of candidates in the contest. Candidates lower in the permutation lose the tiebreaker. Requires a random seed.

Random Seed (required if Tiebreak Mode is "Random", "Previous round counts (then random)", or "Generate permutation"): Enter a positive or negative integer to generate random orders. Impacts operation of randomized functions for tiebreaking. Unavailable if Tiebreak Mode is Stop Counting and Ask, Previous round counts (then stop counting and ask) or Use candidate order in the config file.

Following are screenshots of each of the winning rules interfaces on RCTab.

Single-winner majority determines winner

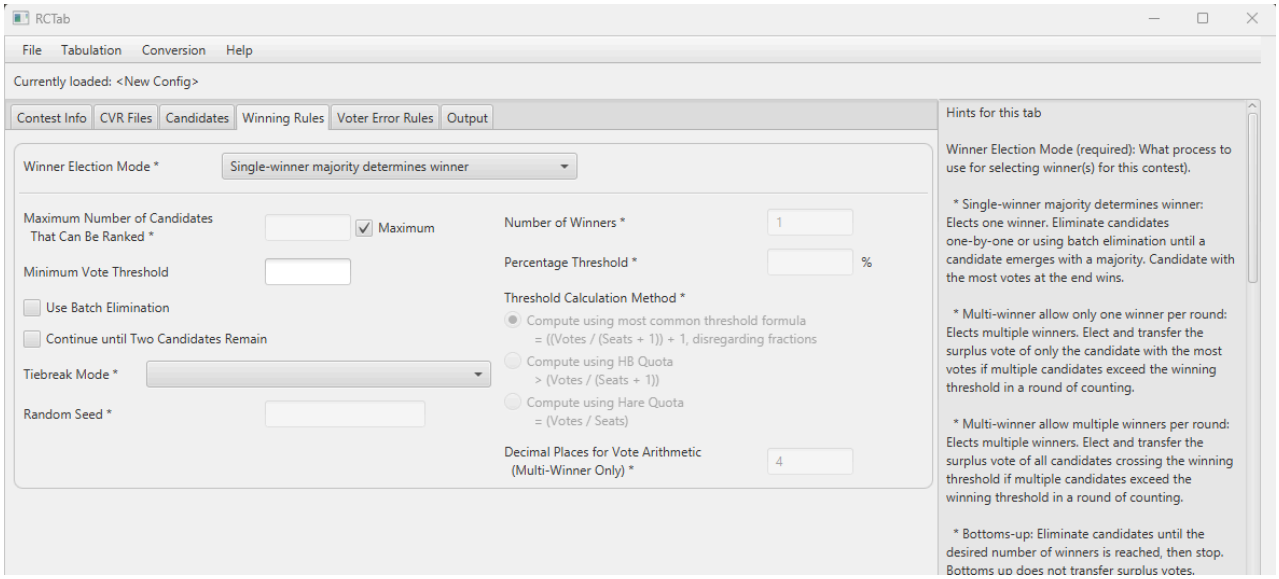


Figure 29: RCTab Tab 4: Winning Rules | Single-winner majority determines winner screen

Permits user to edit these options:	**Automatically fills in and locks this option:**
Maximum Number of Candidates That Can Be Ranked*	Number of Winners*
Minimum Vote Threshold	
Use Batch Elimination	**Invalid options, locked for this mode:**
Continue until Two Candidates Remain	Percentage Threshold
Tiebreak Mode*	Threshold Calculation Method
Random Seed (* if using randomized)	Decimal Places for Vote Arithmetic

Note: Required settings are denoted with an asterisk (*).

Multi-winner allow only one winner per round

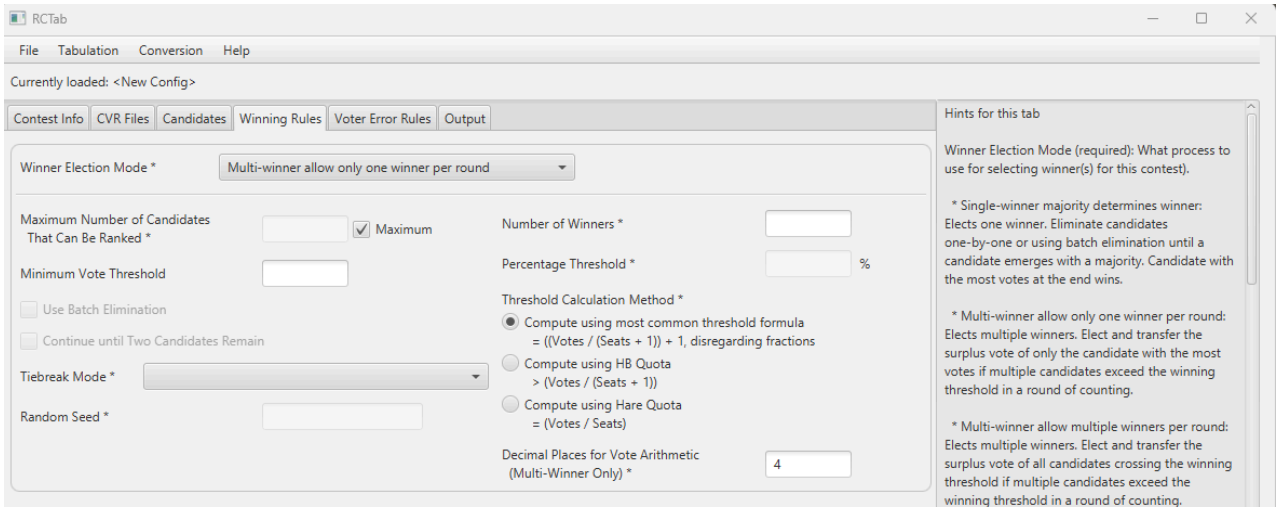


Figure 30: RCTab Tab 4: Winning Rules | Multi-winner allow only one winner per round screen

Permits user to edit these options:

Maximum Number of Candidates That Can Be Ranked*

Minimum Vote Threshold

Tiebreak Mode*

Random Seed (* if using randomized tiebreak mode)

Number of Winners*

Threshold Calculation Method*

Decimal Places for Vote Arithmetic*

Invalid options, locked for this mode:

Percentage Threshold

Use Batch Elimination

Continue until Two Candidates Remain

Note: Required settings are denoted with an asterisk (*).

Multi-winner allow multiple winners per round

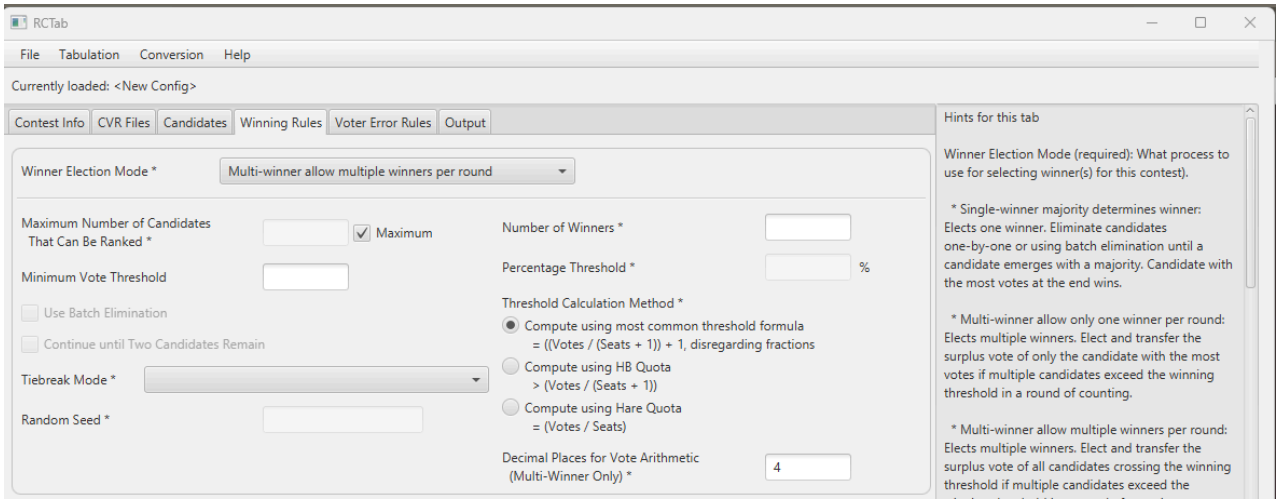


Figure 31: RCTab Tab 4: Winning Rules | Multi-winner allow multiple winners per round screen

Permits user to edit these options:

Maximum Number of Candidates That Can Be Ranked*

Minimum Vote Threshold

Tiebreak Mode*

Random Seed (* if using randomized tiebreak mode)

Number of Winners*

Threshold Calculation Method*

Decimal Places for Vote Arithmetic*

Invalid options, locked for this mode:

Use Batch Elimination

Continue until Two Candidates Remain

Percentage Threshold

Note: Required settings are denoted with an asterisk (*).

Bottoms-Up

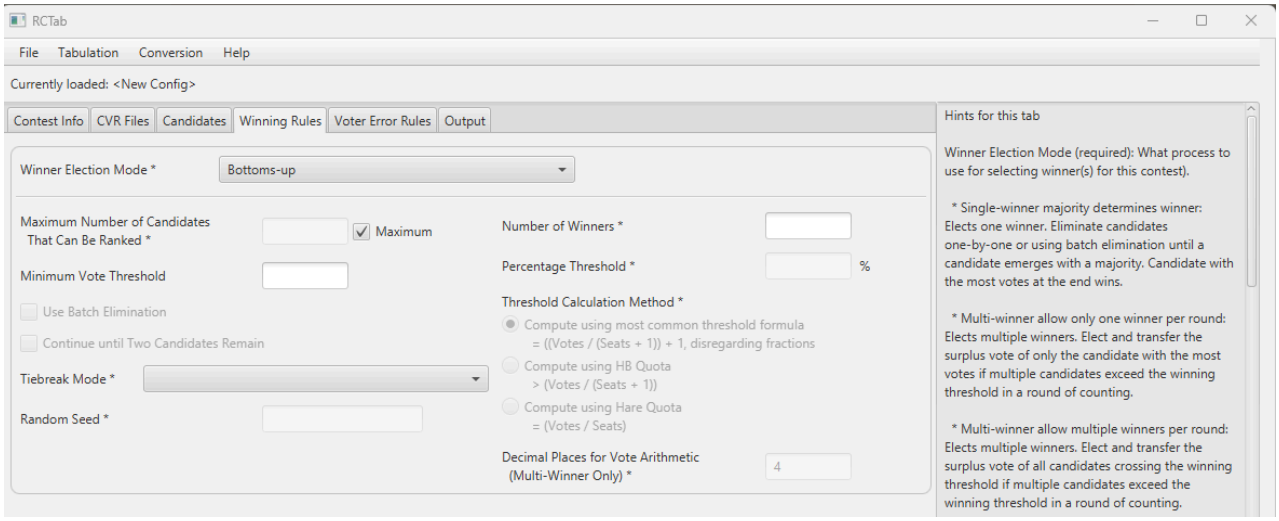


Figure 32: RCTab Tab 4: Winning Rules | Bottoms-up screen

Permits user to edit these options:	**Invalid options, locked for this mode:**
Maximum Number of Candidates That Can Be Ranked*	Use Batch Elimination
Minimum Vote Threshold	Continue until Two Candidates Remain
Tiebreak Mode*	Percentage Threshold
Random Seed (* if using randomized tiebreak mode)	Threshold Calculation Method
Number of Winners*	Decimal Places for Vote Arithmetic

Note: Required settings are denoted with an asterisk (*).

Bottoms-up using percentage threshold

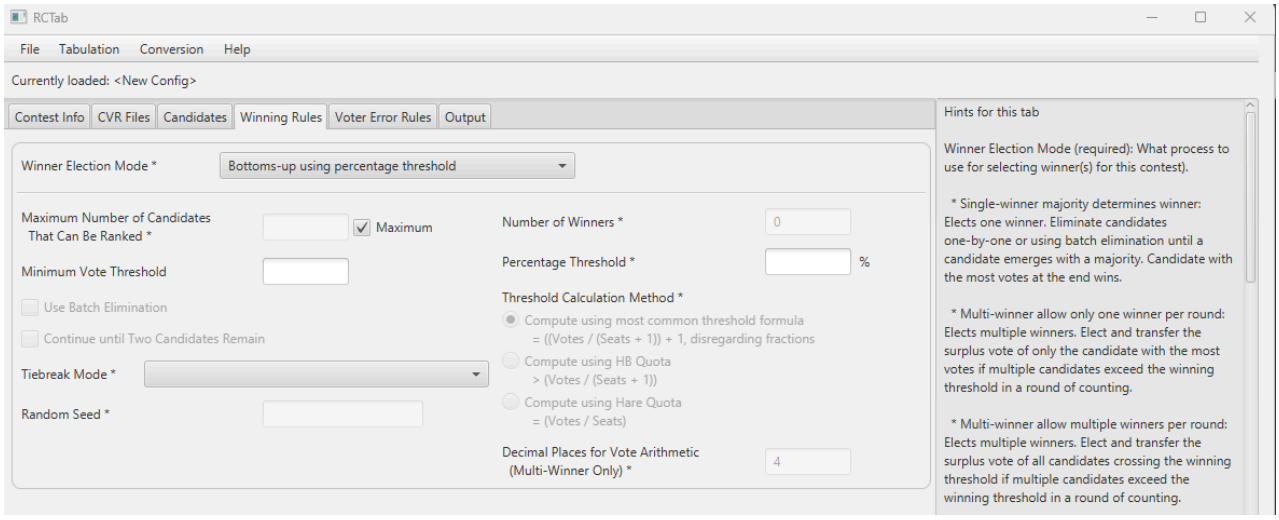


Figure 33: RCTab Tab 4: Winning Rules | Bottoms-up using percentage threshold screen

Permits user to edit these options:	**Automatically fills in and locks this option:**
Maximum Number of Candidates That Can Be Ranked*	Number of Winners*
Minimum Vote Threshold	
Tiebreak Mode*	**Invalid options, locked for this mode:**
Random Seed (* if using randomized tiebreak mode)	Use Batch Elimination
	Continue until Two Candidates Remain
Percentage Threshold*	
	Threshold Calculation Method
	Decimal Places for Vote Arithmetic

Note: Required settings are denoted with an asterisk (*).

Multi-pass IRV

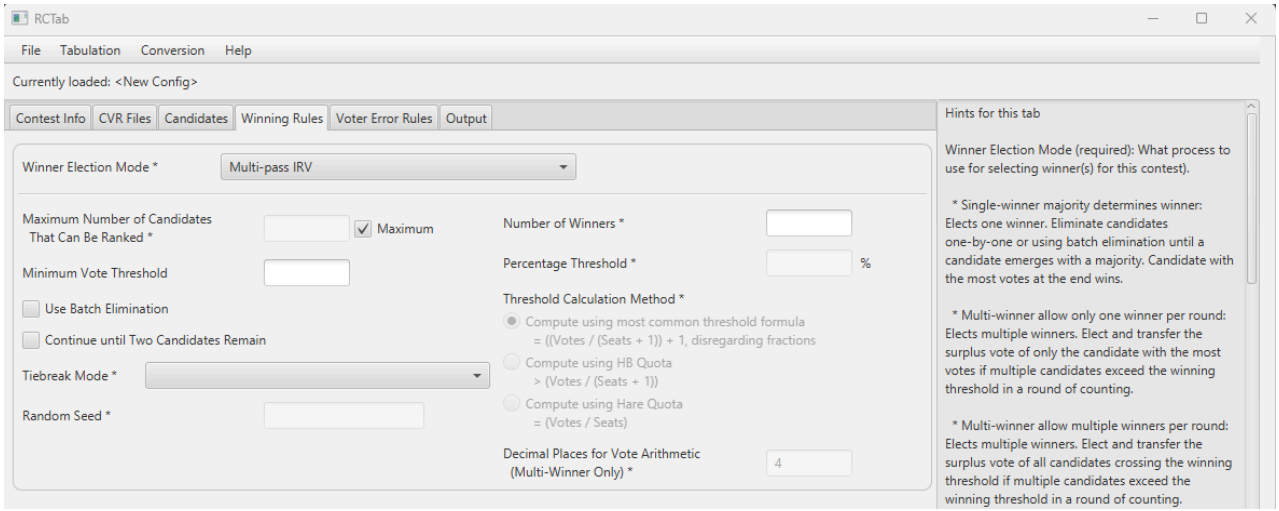


Figure 34: RCTab Tab 4: Winning Rules | Multi-pass IRV screen

Permits user to edit these options (required settings denoted with *):

Maximum Number of Candidates That Can Be Ranked*

Minimum Vote Threshold

Tiebreak Mode*

Random Seed (* if using randomized tiebreak mode)

Number of Winners*

Invalid options, locked for this mode:

Use Batch Elimination

Continue until Two Candidates Remain

Threshold Calculation Method

Decimal Places for Vote Arithmetic

Percentage Threshold

Tiebreak Modes

Random, Generate Permutation, Previous Round Counts (then random)

If users select any of the tiebreaking modes that incorporate randomness (random, generate permutation, previous round counts (then random)) this is how RCTab will appear. Users must fill out the Random Seed setting.

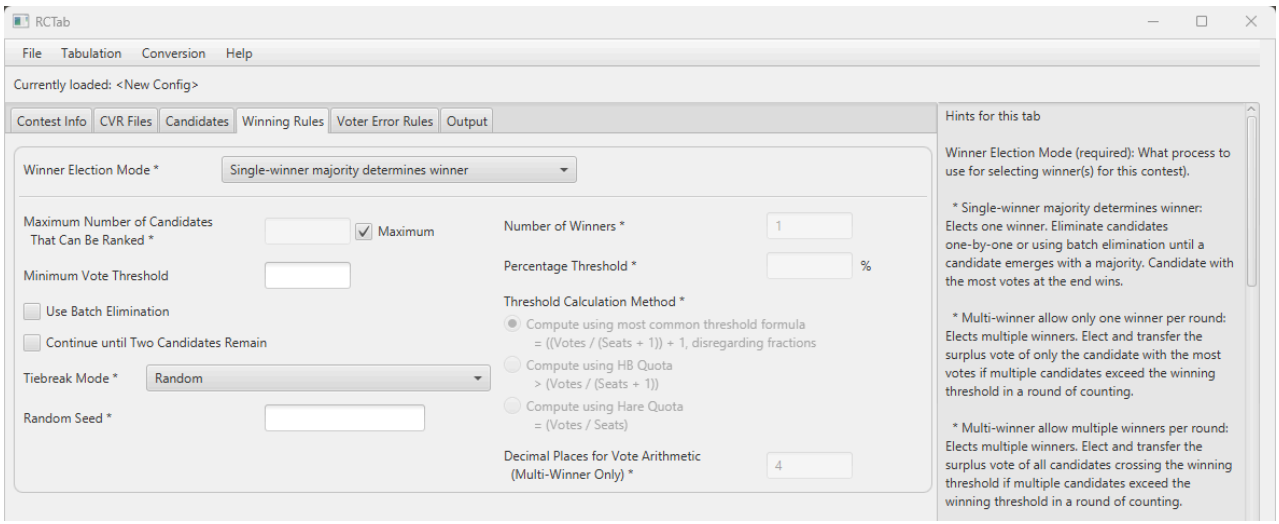


Figure 35: RCTab Tab 4: Winning Rules | Random Tiebreak mode screen

Stop counting and ask, previous round counts (then stop counting and ask), use candidate order in the config file

If users select any of the tiebreaking modes that incorporate user input (Stop counting and ask, previous round counts (then stop counting and ask), use candidate order in the config file) this is how RCTab will appear. The Random Seed field will be locked.

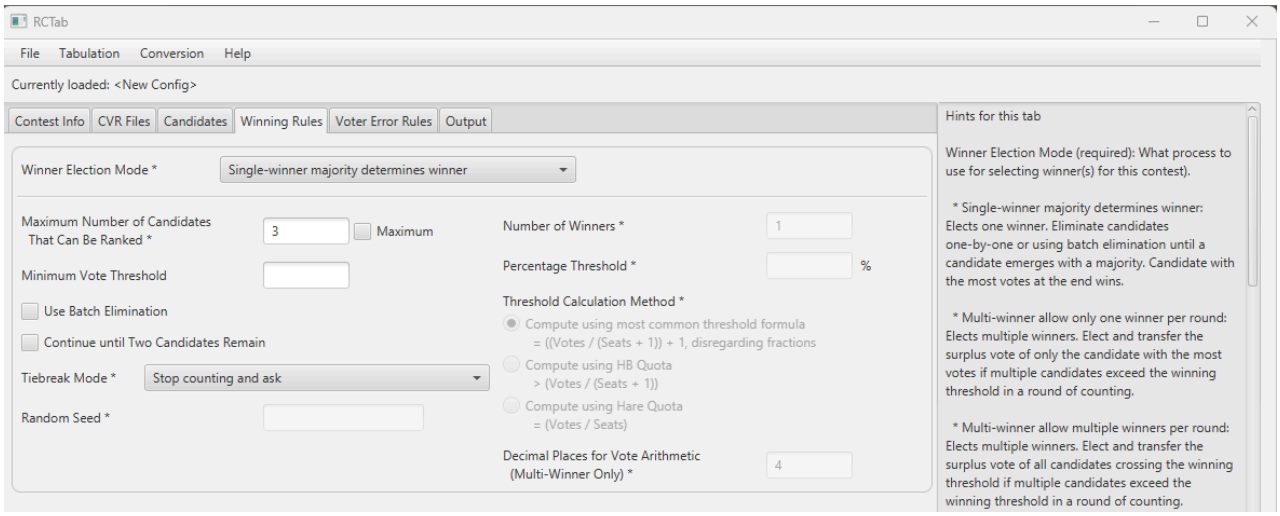


Figure 36: RCTab Tab 4: Winning Rules | User input Tiebreak mode screen

Voter Error Rules

The tabulator needs to know how to handle voter errors in your jurisdiction. These requirements are typically included in statute or regulation. This is a screenshot of what winning rules looks like when a user first navigates to it.

Figure 37: RCTab Tab 5: Voter Error Rules

Overvote Rule (required)**:** How to handle a ballot where a voter has marked multiple candidates at the same ranking when that ballot is encountered in the round-by-round count.

- **Always skip to next rank:** Skips over an overvote and goes to the next validly-marked ranking on a ballot.
    - Requires Overvote Label to be supplied (when overvote label can be edited)
- **Exhaust immediately:** A ballot with an overvote exhausts when that overvote is encountered in the rounds of counting.
    - Requires overvote label to be supplied (when overvote label can be edited)
- **Exhaust if multiple continuing:** If a voter has an overvote but only one candidate at that overvote is still in the race when that overvote is encountered, the ballot counts for that candidate. If multiple candidates at the overvote are still in the race, the ballot exhausts.
    - Requires overvote delimiter to be supplied (when overvote delimiter can be edited)

How Many Consecutive Skipped Ranks Are Allowed (required)**:** How many rankings in a row can a voter skip and still have later rankings count? 0 allows no skipped rankings. 1 allows voters to skip rankings one at a time, but not more than 1 in a row, and so on. Selecting the checkbox disables the text box and inserts the value “unlimited”. Checkbox can be unselected. Default value: 1. Example: A voter could rank their 1st, 3rd and 5th choices and not exhaust the ballot under this rule, for example.

Exhaust on Multiple Ranks for the Same Candidate (optional)**:** When checked, the tabulator will exhaust a ballot that includes multiple rankings for the same candidate when that repeat ranking is reached. When unchecked repeated rankings will not exhaust the ballot. Example: A voter ranks the same candidate 1st and 3rd, a different candidate 2nd, and another candidate 4th. If their original first choice and their second choice are eliminated, the ballot exhausts when it reaches the repeat ranking in rank 3. The ranking in the 4th rank does not count.

Below is an example screenshot of a sample filled-out voter error rules tab. The user has selected Always skip to next rank for Overvote, has selected unlimited for How Many Consecutive Skipped Ranks Are Allowed (making the text box inoperable) and has left Exhaust on Multiple Ranks for the Same Candidate blank.

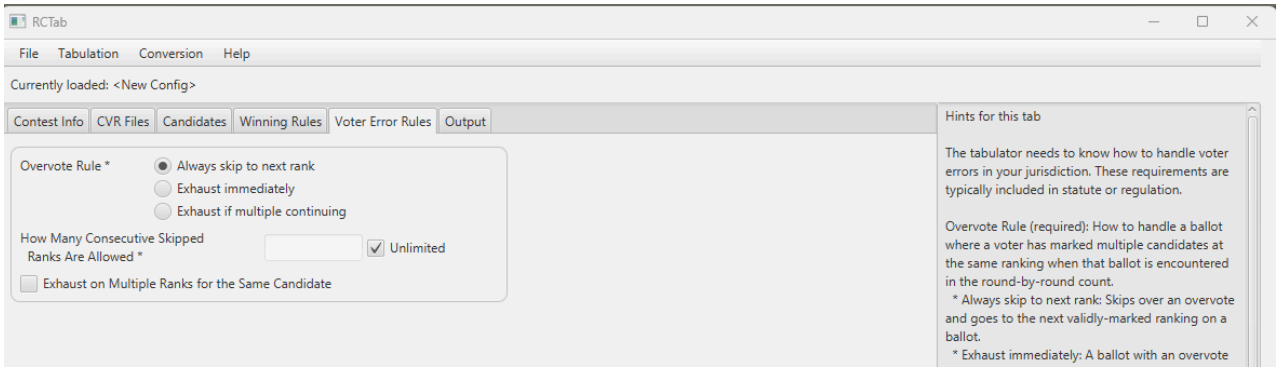


Figure 38: Example Complete Voter Error Rules tab

Output Tab

Tell the tabulator where results files go and what additional results files you want. This is a screenshot of what output looks like when a user first navigates to it.

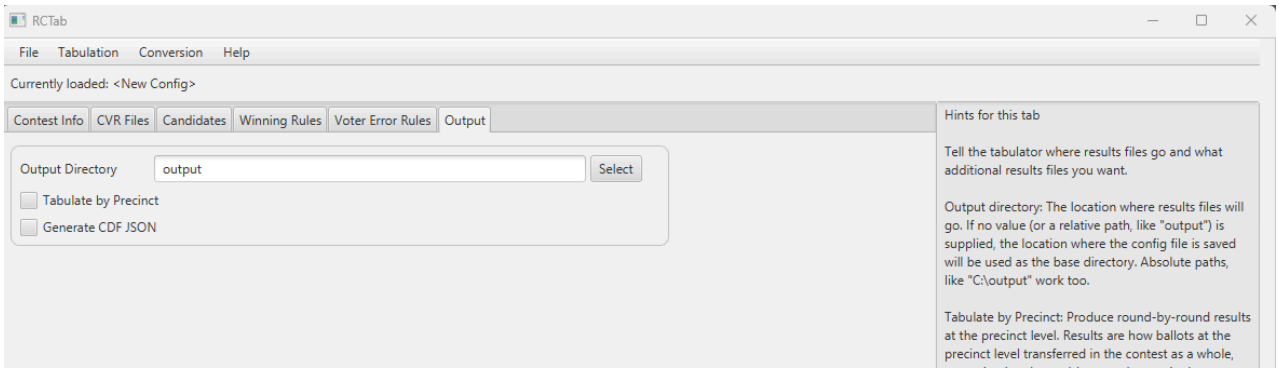


Figure 39: RCTab Tab 6: Output

Output directory: One folder per contest tabulated should be created. Output directory instructs RCTab where to save any output files from successful RCTab tabulations. The location where results files will go. If no value (or a relative path, like output) is supplied, the location where the config file is saved will be used as the base directory. Absolute paths, like C:\output work too. Select allows users to navigate through Explorer to select a location.

RCTab will not allow the output path to be configured to a Windows user account folder (by default anything under the path c:\users\). This requirement ensures read-only output. You **must** follow the instructions in **Section 22 - Installation Instructions for Windows OS** to set permissions on the output folder. Following these instructions ensures that the 'RCTab' user account cannot edit or delete RCTab summary output files or audit logs.

Tabulate by Precinct: Produce round-by-round results at the precinct level. Results are how ballots at the precinct level transferred in the contest as a whole, not a simulated round-by-round count in the precinct. Requires precinct information in CVR Files tab for ES&S CVRs. Impacts which files are written to the output directory.

Generate a CDF JSON: Produce a VVSG common data format JSON file of the CVR. Impacts which files are written to the output directory.

Examples of output files include:

1. summary.csv

	A	B	C	D
1	Contest	Test Set 1		
2	Jurisdiction	TestSets_Single_winner		
3	Office	Exhaust at overvote		
4	Date	7/17/2019		
5	Winner(s)	B		
6	Threshold	7		
7				
8	Rounds	Round 1	Round 2	Round 3
9	Eliminated	D	C	
10	Elected			B
11	B	6	6	8
12	A	5	5	5
13	C	2	3	0
14	D	1	0	0
15	Inactive ballots	1	1	2

Figure 40. RCTab Summary.csv file example.

The summary.csv file contains the following fields, populated with information from the config file used for tabulation:

1) B1: Contest name
2) B2: Jurisdiction name
3) B3: Office name
4) B4: Date of contest
5) B5: Winner names
6) B6: Election threshold
7) Row 8: Round labels
8) Row 9: Candidates eliminated in a round, if any
9) Row 10: Candidates elected in a round, if any
10) A11–A11+ number of total candidates: Candidate names
11) B11–ZXX: Candidate totals in a round (final cell depends on total number of candidates and rounds in a contest)
12) BXX+1–ZXX+1: Inactive ballot totals (cell labels depend on total number of candidates and rounds in a contest)

13) BXX+2–ZXX+2: Residual surplus totals [multi-winner only] (cell labels depend on total number of candidates and rounds in a contest)

2. audit_0.log

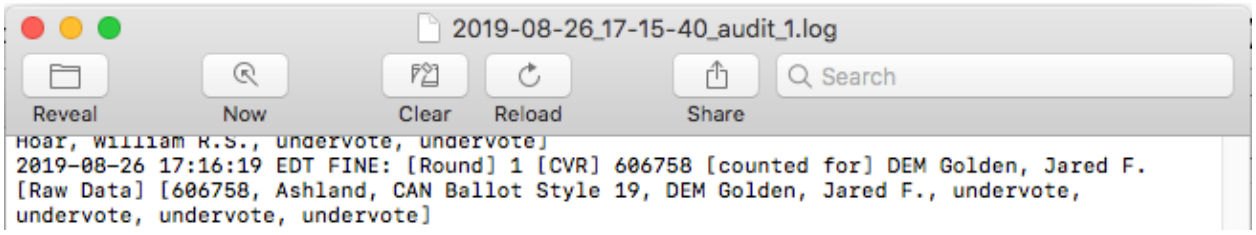


Figure 41. RCTab Audit Log example.

The audit log provides a summary description of how a given ballot counted in each round of counting in an election. Depending on the size of the election, multiple audit logs may be created (the maximum size of an audit log is 50MB, so as soon as an audit log becomes 50MB in size, a new one is created). Ballots are labeled according to their ID number in a CVR, if identified in the CVR tab of the tabulator, or by their row number in the spreadsheet, if no ID number column is specified in the CVR tab.

In the above example, the ballot was at Aug. 26, 2019 at 5:16:19pm EDT. This particular log was for the ballot as counted in Round 1. The CVR label of this ballot was 606758. The ballot counted for DEM Golden, Jared F.

The audit log then includes raw data for the ballot which, generically, is [CVR ID, Precinct label, Ballot Style number, first rank value, second rank value, third rank value, fourth rank value, fifth rank value]. The number of ranking values included is based on the ranking limitation indicated in the config and set using the rules tab of the Tabulator.

3. summary.json

2019-08-26_17-15-40_summary.json


```
{
  "config" : {
    "contest" : "Maine 2018 CD2",
    "date" : "2018-06-12",
    "jurisdiction" : "Maine",
    "office" : "Congressional District 2",
    "threshold" : "140686"
  },
  "results" : [ {
    "round" : 1,
    "tally" : {
      "Bond, Tiffany L." : "16552",
      "DEM Golden, Jared F." : "132013",
      "Hoar, William R.S." : "6875",
      "REP Poliquin, Bruce" : "134184"
    },
    "tallyResults" : [ {
      "eliminated" : "Hoar, William R.S.",
      "transfers" : {
        "DEM Golden, Jared F." : "2565",
        "REP Poliquin, Bruce" : "1630",
        "exhausted" : "2680"
      }
    }, {
      "eliminated" : "Bond, Tiffany L.",
      "transfers" : {
        "DEM Golden, Jared F." : "7862",
        "REP Poliquin, Bruce" : "3117",
        "exhausted" : "5573"
      }
    } ]
  }, {
    "round" : 2,
    "tally" : {
      "DEM Golden, Jared F." : "142440",
      "REP Poliquin, Bruce" : "138931"
    },
    "tallyResults" : [ {
      "elected" : "DEM Golden, Jared F.",
      "transfers" : { }
    } ]
  } ]
}
```


Figure 42. RCTab Summary.json file example.

The summary.json file includes similar data to the summary.csv file in .json format. It can be used with the visualizer application at this link to visualize results from an RCV contest.

The first set of information, “config,” details the contest, date of contest, jurisdiction, office being elected, and threshold of election.

Rounds are then presented, broken out in to vote totals for each candidate at the start of the round and information about each candidate elected or eliminated in a round. Total votes transferred from a candidate to other candidates are then provided.

4. cvr_cdf.json

2019-09-05_15-55-50_cvr_cdf.json


```
      } ],
      "Type" : "interpreted"
    } ],
    "CurrentSnapshotId" : "ballot-20-round-2",
    "ElectionId" : "election-001"
  }, {
    "@type" : "CVR.CVR",
    "BallotPrePrintedId" : "21",
    "CVRSnapshot" : [ {
      "@id" : "ballot-21",
      "@type" : "CVR.CVRSnapshot",
      "CVRContest" : [ {
        "@type" : "CVR.CVRContest",
        "CVRContestSelection" : [ {
          "@type" : "CVR.CVRContestSelection",
          "ContestSelectionId" : "cs-dem_golden__jared_f.",
          "SelectionPosition" : [ {
            "@type" : "CVR.SelectionPosition",
            "HasIndication" : "yes",
            "IsAllocable" : "unknown",
            "NumberVotes" : 1,
            "Rank" : 1
          } ]
        } ],
        "ContestId" : "contest-001"
      } ],
      "Type" : "original"
    }, {
      "@id" : "ballot-21-round-1",
      "@type" : "CVR.CVRSnapshot",
      "CVRContest" : [ {
        "@type" : "CVR.CVRContest",
        "CVRContestSelection" : [ {
          "@type" : "CVR.CVRContestSelection",
          "ContestSelectionId" : "cs-dem_golden__jared_f.",
          "SelectionPosition" : [ {
            "@type" : "CVR.SelectionPosition",
            "HasIndication" : "yes",
            "IsAllocable" : "yes",
            "NumberVotes" : 1,
            "Rank" : 1
          } ]
        } ],
        "ContestId" : "contest-001"
      } ],
      "Type" : "interpreted"
    }, {
      "@id" : "ballot-21-round-2",
      "@type" : "CVR.CVRSnapshot",
      "CVRContest" : [ {
        "@type" : "CVR.CVRContest",
        "CVRContestSelection" : [ {
          "@type" : "CVR.CVRContestSelection",
          "ContestSelectionId" : "cs-dem_golden__jared_f.",
          "SelectionPosition" : [ {
            "@type" : "CVR.SelectionPosition",
            "HasIndication" : "yes",
            "IsAllocable" : "yes",
            "NumberVotes" : 1,
```


Figure 43. RCTab CVR CDF .json example.

“Generate CDF JSON" will instruct the tabulator to generate a JSON of cast vote records (CVR) in the Common Data Format (CDF). The CVR CDF .json file includes information about how each ballot was processed in each round, following the common data format laid out in 1500-103.

Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.0.1

Letter Report

To: The Election Administration Resource Center dba Ranked Choice Voting Resource Center

From: Wendy Owens - Pro V&V, Inc.

CC: Keith Long – RCVRC; Alan Simmons, Michael Walker - Pro V&V, Inc.

Date: August 30, 2019

Subject: Universal RCV Tabulator v1.0.1

Dear RCVRC:

Pro V&V, a VSTL, is providing this letter to report the results of the modification testing performed on the RCVRC's Ranked Choice Voting Tabulator (Universal RCV Tabulator 1.0.1), a post-EMS secondary tabulator, which in this instance is an external software modification-addition to the ES&S's EVS 6.0.4.0 EAC Certified Voting System (ESSEVS6040 – May 03, 2019).

This testing campaign was performed, with the support of ES&S, to verify that the submitted modification-addition meets the certification requirements found in the Voluntary Voting System Guidelines Version 1.0 (VVSG 1.0).

To start, Pro V&V determined that this modification was subject only to limited certification testing as RCVRS was able to establish that this detached modification-addition did not affect the previously demonstrated compliance of the baseline system to the VVSG 1.0. Limited testing, in addition to other stated uses, is intended to facilitate the integration of vote counting software with other systems and election software. The following modifications to the previously certified system were evaluated:

- Addition of the Universal RCV Tabulator

Background

The ES&S EVS 6.0.4.0 voting system is a complete EAC certified voting system composed of EMS software applications, central count devices, and polling location devices.

This Universal RCV Tabulator can process data from voting machines that are capable of exporting cast vote records (CVR's) and tabulate a single-winner or multi-winner ranked choice voting election according to the rules used in current state, county or city election jurisdiction in the United States. The Universal RCV Tabulator outputs the tabulation results in comma separated values (.csv) tabular data format and creates an audit file for the RCV election.

Although the Tabulator was tested using files produced by the EVS 6.0.4.0, ES&S confirmed that the exported files created from all EVS certified systems among EVS 5.0.0.0 thru 6.0.4.0 are identical and would produce the same testing results with the Universal RCV Tabulator as noted in this report

Application Summary

The Universal RCV Tabulator was configured and setup as it would be for normal operations as per the TDP. Both RCVRC and ES&S provided materials and phone support as needed to aid in the testing process.

For testing, the Universal RCV Tabulator resided on three independent test support platforms with the Cast Vote Records (CVR) originating from the ES&&S EVS 6.0.4.0 baseline system. The modified system supported voting variations are:

- Closed primaries
- Open primaries
- Partisan offices
- Non-partisan offices
- Write-in voting
- Ballot rotation
- Straight party voting
- Cross-party endorsement
- Split precincts
- Vote for N of M
- Ranked order voting (w/Universal RCV Tabulator)
- Provisional or challenged ballots

Software

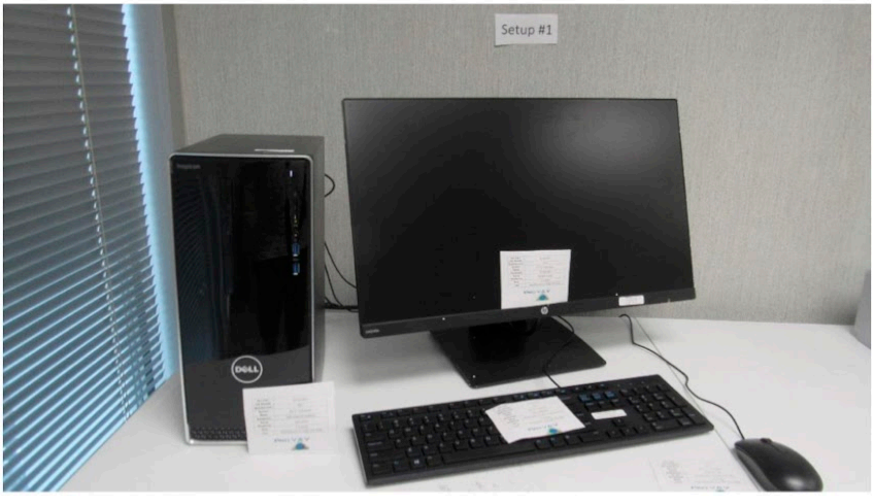
Component Name	Version	Unique Identifier (SHA 512 hash value)
Universal RCV Tabulator	Windows 1.0.1	a83d84e22453ac656d9fd1b78e326e96a034801174b043a4ba1afa5167442050059d74962e6e0ee278063797527aa009d33ea70951e37fc30cc3cc77163800da
Universal RCV Tabulator	OSX 1.0.1	a0ea658cd0d8b685436aec7a240de3f9b0ed84980823a489586c249a79c1da8a4f5bfc99f94c10969ebb821f53dbb0d9c6da606b28adee8cfe4fa7e3beab4f7b
Universal RCV Tabulator	Linux 1.0.1	92c7af9b074929206e196e711c7399c51f8386712b6f1b7b5c31e51be18511c8af7cfca36c77427ed832f89b8b186d59f97f2cdccbc78e990a2dab7da1f2c5b4

Testing Hardware

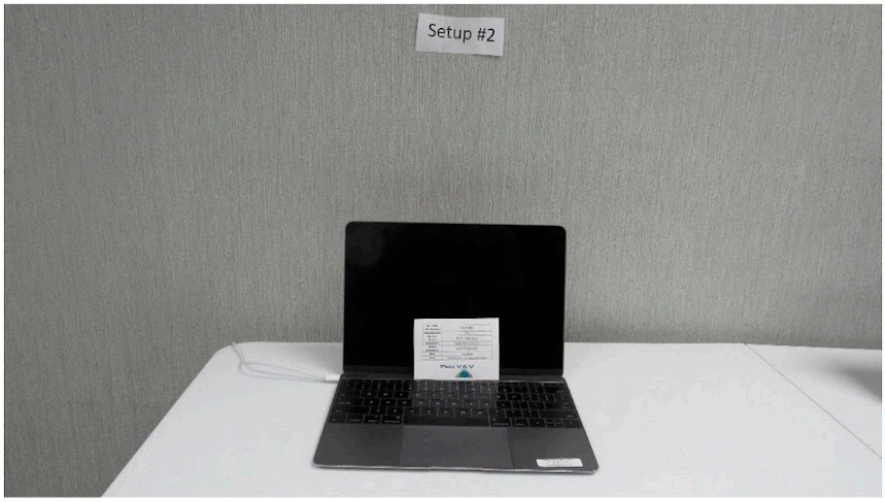
Component Name	Model/Version Number	Serial Number(s)	Description
Universal RCV Tabulator	Dell Inspiron Desktop Windows 10 Pro	1QLJDW2	Universal RCV Tabulator Platform
Universal RCV Tabulator	Apple Macbook 8.1 Mac OS 10.13	CO2PP2BAGCN3	Universal RCV Tabulator Platform
Universal RCV Tabulator	HP Elite Book Laptop 840 Mint 19.1	SCG6284VLW	Universal RCV Tabulator Platform
Printer	Samsung Xpress M2020W Printer	CNB3M47DJ8	Universal RCV Tabulator Printer
Uninterruptable Power Supply	APC Backup UPS 600	4B191P18483	Universal RCV Tabulator External Power Supply

Testing Support Materials

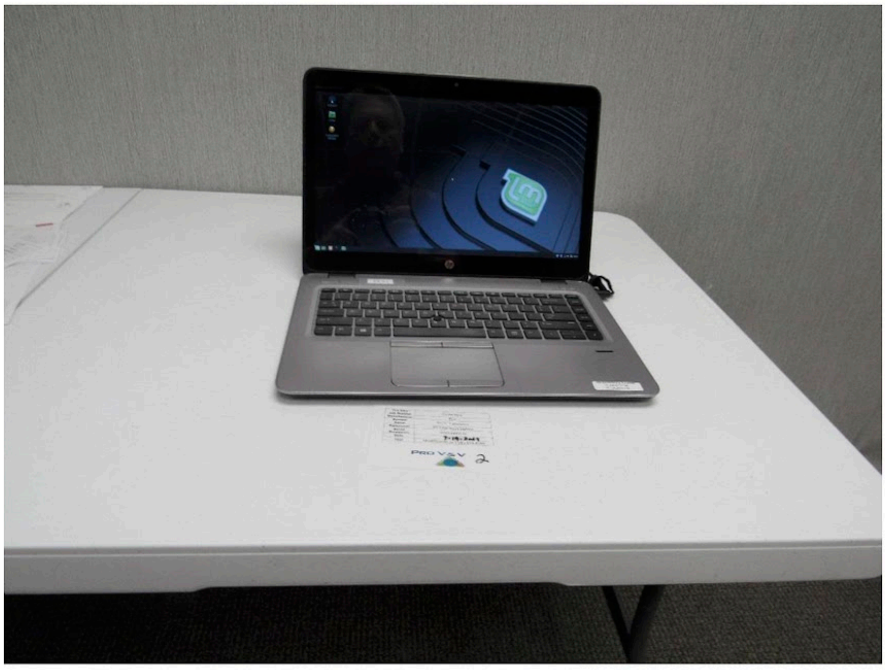
Material	Quantity	Product Number(s)	Description
8.5 x 11 Printer Paper	100 Sheets	HP 200350	Report Paper
Thumb Drive	4	N/A	Transport Media



Photograph 1: Universal RCV Tabulator Setup 1 (Windows)



Photograph 2: Universal RCV Tabulator Setup 2 (OSX)



Photograph 3: Universal RCV Tabulator Setup 3 (Linux)

Testing Summary

This testing campaign concentrated on the addition of a Universal RCV Tabulator to the ES&S EVS 6.0.4.0 EAC certified voting system and the associated characteristics that necessitate further review.

Technical Documentation Package (TDP) Review

The TDP review was done to verify that the submitted Universal RCV Tabulator documentation contained sufficient technical and administrative information to allow successful operation of the Universal RCV Tabulator with the baseline system. The TDP review also assessed the extent to which the modified TDP adhered to the VVSG 1.0. This evaluation was an iterative process with concerns communicated to the RCVRC as early in testing as possible for resolution. All changes made to the TDP during testing were subjected to re-examination.

RCV TDP

Document #	Document	Version #
000	Technical Data Package	0.3
100	Tabulation Options for RCV	0.1
105	Overview	0.1
110	Expected Outcome RCV Test Sets Single-Winner	0.1
115	Expected Outcome RCV Test Sets Multi-Winner	0.1
125	ES&S Ballot Layout Limits	0.2
130	RCV Tabulator User Guide Overview	0.3
150	Universal RCV Tabulator System Security Specifications	0.3
120	Installation Instructions for RCV Tabulator	0.7
121	Installation Instructions for Universal RCV Tabulator – MacOS	0.4
122	Installation Instructions for Universal RCV Tabulator – Linux	0.4
140	RCV Tabulator Operation Instructions	0.4
141	Universal RCV Tabulator Operation Instructions	0.4
142	Universal RCV Tabulator Operation Instructions	0.4
135	Configuration File Parameters	0.4
200	Universal RCV Tabulator Election Setup Checklist - MANUAL	0.3
201	Universal RCV Tabulator Election Setup Checklist - INTERACTIVE	0.3
205	Universal RCV Tabulator cvr Files	0.1
210	Universal RCV Tabulator Election Data Form	0.3
215	Universal RCV Tabulator cvr Files	0.1
220	Universal RCV Tabulator Election Rules	0.3
230	Universal RCV Tabulator Logic & Accuracy Test	0.1
250	Universal RCV Tabulator Election Processing	0.3
251	Universal RCV Tabulator Election Processing Linux OS	0.1
252	Universal RCV Tabulator Election Processing MacOS	0.1
225	Universal RCV Tabulator System Operator Log Errors	0.2

Summary Findings: *The addition of the Universal RCV Tabulator TDP did not affect the ability of the baseline TDP to satisfy all VVSG 1.0 requirements. Additionally, the final Universal RCV Tabulator documentation editions provided sufficient detail to adequately attend to the added functionality.*

Physical Configuration Audit (PCA)

The PCA was conducted to ensure that the configurations and setups of the Universal RCV Tabulator submitted for testing matched the vendor's technical documentation, and confirmed that the documentation was sufficient for the user to install, validate, operate, and maintain the Universal RCV Tabulator. All changes made to the Universal RCV Tabulator during testing were subjected to re-examination.

The PCA further established a configuration baseline of the software and hardware to be tested.

Summary Findings: *The addition of the three employed stand-alone and isolated Universal RCV Tabulator platforms matched the documentation and did not impact the ability of the baseline system as modified to satisfy the VVSG 1.0 requirements.*

Functional Configuration Audit (FCA)

The FCA testing utilized standardized test cases to verify that the Universal RCV Tabulator satisfied prescribed requirements and performed the functions claimed in the system documentation.

Summary Findings: *The Universal RCV Tabulator passed all of the applicable pre-vote, vote, and post-vote test case requirements. The Universal RCV Tabulator demonstrated the ability to function as required and intended, and does not impact the ability of the baseline system as modified to satisfy the VVSG 1.0 requirements.*

Source Code Review

The Universal RCV Tabulator source code review was to ascertain how completely the software conforms to the vendor's specifications. The source code inspection also assessed the extent to which the code adhered to the VVSG 1.1.

Summary Findings: *The Universal RCV Tabulator stands as a VVSG 1.1 compliant source code modification to a VVSG 1.0 system, and meets all of the VVSG 1.1 source code requirements.*

Security Testing

The security testing evaluated the effectiveness of the Universal RCV Tabulator in detecting, preventing, recording, reporting, and recovering from security threats. To evaluate the integrity of the Universal RCV Tabulator attempts were made to defeat the access controls and security measures documented in the system TDP.

Summary Findings: *A fundamental security review was performed on the Universal RCV Tabulator, which incorporates the security policies of the baseline system.*

The Universal RCV Tabulator was found to have an applied level of security that complies with the verified VVSG 1.0 security provisions. Additionally the presence of the Universal RCV Tabulator does not impact the ability of the baseline system as modified to satisfy the VVSG 1.0 security requirements.

Usability Testing

The usability testing encompassed a range of methods that examined how users in the target audience actually interact with the Universal RCV Tabulator.

Summary Findings: *The Universal RCV Tabulator has operations and controls that can easily be employed by users in the target audience making use of the system TDP.*

Accessibility Testing

The accessibility testing measured characteristics that indicated the degree to which the Universal RCV Tabulator is available to, and usable by, users in the target audience.

Summary Findings: *The Universal RCV Tabulator is accessible to users with a basic understanding of computers and election management.*

System Integration

The system level testing addressed the integrated operations of the Universal RCV Tabulator with the baseline system, and tested the Universal RCV Tabulator capabilities as part of the baseline system as a whole. This testing validated the seamless functional integration and data inputs/outputs of the baseline system with the Universal RCV Tabulator. For this testing the Universal RCV Tabulator were configured and operated as for normal operations.

Summary Findings: *Cast Vote Records (CVR) that were generated by the baseline system were imported into the Universal RCV Tabulator and processed through tabulation to report generation. No system issues were noted during testing.*

Accuracy Testing

Universal RCV Tabulator accuracy was based upon the totality of all testing.

Summary Findings: *During testing, actual results were validated against expected results. No accuracy issues were noted throughout the test campaign.*

Conclusion

Based on the testing performed and the results obtained, the Universal RCV Tabulator solution identified in this letter-report is believed to meet the requirements set forth by the VVSG 1.0.

Should you require additional information or would like to discuss this matter further, please contact me at 256-713-1111.

Sincerely,

Wendy Owens
VSTL Program Manager
wendy.owens@provandv.com

Certification of RCV Tabulator including compliance with NIST 1500-103 Cast Vote Record CDF, Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.1.0

TEST REPORT

FOR

EXAMINATION OF RANKED CHOICE VOTING TABULATOR v1.1.0
The Election Administration Resource Center dba Ranked Choice Voting Resource Center

TO

VOLUNTARY VOTING SYSTEM GUIDELINES (VVSG) 1.0

Prepared by: Wendy Owens
Wendy Owens, VSTL Program Manager

May 20, 2020

Pro V&V, Inc.
6705 Odyssey Drive, Suite C
Huntsville, AL 35806
256-713-1111

PRO V&V, Inc.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 References 1

1.2 Terms and Abbreviations 2

1.3 Background 2

1.4 Scope 2

2.0 SYSTEM IDENTIFICATION 3

2.1 Software 3

2.2 Hardware 4

2.3 Peripherals 4

2.4 Test Support Equipment/Materials 4

2.5 Technical Data Package 4

3.0 TEST PROCESS AND RESULTS 5

3.1 General Information 5

3.2 Test Approach 5

3.3 Test Process and Results 6

4.0 CONCLUSION 9

1.0 INTRODUCTION

Pro V&V performed an examination effort on the RCVRC Ranked Choice Voting Tabulator (RCV-Tabulator 1.1.0), which is a modification to RCV-Tabulator 1.0.1, which was in turn a modification to the ES&S EVS 6.0.4.0 voting system. Pro V&V performed a functional regression examination of the submitted RCV-Tabulator v1.1.0 configured as it will be for normal use.

The baseline RCV-Tabulator testing campaign was performed, with the support of ES&S, to verify that the submitted modification-addition meets the certification requirements found in the Voluntary Voting System Guidelines Version 1.0 (VVSG 1.0).

The modifications to the previously examined version that are listed in the following document were evaluated:

- Universal RCV Tabulator v1.1.0 Enhancements, v1.0, dated 4-17-2020

This Test Report documents the procedures followed, the results obtained, and the conclusions drawn from this examination.

1.1 References

The documents listed below were utilized in the development of the Test Report.

- Universal RCV Tabulator v1.1.0 Enhancements, v1.0, dated 4-17-2020
- Implementation Statement
- Election Assistance Commission 2005 Voluntary Voting System Guidelines (VVSG) Version 1.0, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission 2015 Voluntary Voting System Guidelines (VVSG) Version 1.1, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission Testing and Certification Program Manual, Version 2.0
- Election Assistance Commission Voting System Test Laboratory Program Manual, Version 2.0
- National Voluntary Laboratory Accreditation Program NIST Handbook 150, 2016 Edition, "NVLAP Procedures and General Requirements (NIST Handbook 150)", dated July 2016
- National Voluntary Laboratory Accreditation Program NIST Handbook 150-22, 2008 Edition, "Voting System Testing (NIST Handbook 150-22)", dated May 2008
- United States 107th Congress Help America Vote Act (HAVA) of 2002 (Public Law 107-252), dated October 2002
- Pro V&V, Inc. Quality Assurance Manual, Revision 7.0

- System Technical Data Package

1.2 Terms and Abbreviation

This subsection lists terms and abbreviations relevant to this examination:

"COTS" – Commercial Off-The-Shelf

"CVR" – Cast Vote Record

"EAC" – United States Election Assistance Commission

"EBM" -- Electronically-assisted Ballot Marker

"FCA" – Functional Configuration Audit

"HAVA" – Help America Vote Act

"ISO" – International Organization for Standardization

"PCA" – Physical Configuration Audit

"QA" – Quality Assurance

"TDP" – Technical Data Package

"VSTL" – Voting System Test Laboratory

"VVSG" – Voluntary Voting System Guidelines

1.3 Background

The ES&S EVS 6.0.4.0 voting system is a complete EAC certified voting system composed of EMS software applications, central count devices, and polling location devices.

The Universal RCV-Tabulator v1.1.0 processes digital cast vote records and election data from various voting machine vendors. Supported formats include ES&S (.xlsx), Dominion (by conversion to .csv) and NIST Common Data Format (.json). The Tabulator can ingest election data and tabulate single-winner and multi-winner ranked choice voting elections according to many rules used in current state, county, and city jurisdictions in the USA. The Tabulator generates various reports including a tabular results report in .csv format, a results summary .json file (for visualizing), a Common Data Format JSON CVR Record Report containing round-by-round "snapshots of each CVR for each round and a tabulation audit log.

1.4 Scope

The scope of testing focused on regression testing of the RCV-Tabulator v1.1.0. To this end, CVRs were imported and the results tabulated using the RCV-Tabulator. Additionally, ES&S CVRs were converted to Common Data Format (CDF) and CDFs files were tabulated.

2.0 SYSTEM IDENTIFICATION

This section contains detailed descriptions of the submitted RCV-Tabulator solution including: platforms for required technical configuration testing, items tested during the evaluation (including all software, hardware, and peripherals, both proprietary and COTS), and any test support equipment or materials necessary for test performance.

Application Summary

The RCV-Tabulator was configured and setup as it would be for normal operations as per the TDP. Both RCVRC and ES&S provided materials and phone support as needed to aid in the testing process.

For testing, the RCV-Tabulator resided on three independent test support platforms with the Cast Vote Records (CVR) originating from the ES&&S EVS 6.0.4.0 baseline system. The modified system supported voting variations are:

- Closed primaries
- Open primaries
- Partisan offices
- Non-partisan offices
- Write-in voting
- Ballot rotation
- Straight party voting
- Cross-party endorsement
- Split precincts
- Vote for N of M
- Ranked order voting (w/RCV-Tabulator)
- Provisional or challenged ballots

2.1 Software

The software listed in Table 2.1 was used during the examination of the RCV-Tabulator. Table 2-1 provides the component name, software/firmware version and a SHA-512 Hash, if calculated, during the examination.

Table 2-1 Software

Component Name	Version	Unique Identifier (SHA 512 Hash Value)
RCV Tabulator	Windows 1.1.0	baf69aa4dffbbdc360356d985e0a3a9c83b402fbeda4ffc7b0c1a0a18bbe261f268f059825aa28e2008263f534f2072fab48c2707b5268a842db3134e676e148
RCV Tabulator	OSX 1.1.0	8d8c1f2b6017f8e7cd43def6ed926fb2ca125cdb58110f0ad2d18af7129aa9148742f850aa9907edb7ac2ab5c1cc8e095fc52ea2ad828fdd160cb2b6b19c0c10

Table 2-1 Software *(continued)*

Component Name	Version	Unique Identifier (SHA 512 Hash Value)
RCV Tabulator	Linux 1.1.0	cdbbbae474983ab21845cdf996b85a632af4d5aa9a80c7ab433304 1f652c6c0c73202d442c49efc5baaca09fd9b6493678e336676b93d 639804e38b733cac3df

2.2 Hardware

The tables below provide hardware component name, description, model number, and the serial number for the units that were examined.

Table 2-2 Hardware

Component Name	Model/Version Number	Operating System	Serial Number(s)	Description
RCV-Tabulator	Dell Inspiron Desktop	Windows 10 Pro	1QLJDW2	RCV-Tabulator Platform
RCV-Tabulator	Apple Macbook	OS 10.13	CO2PP2BAGCN3	RCV-Tabulator Platform
RCV-Tabulator	HP Elite Book Laptop 840	Mint 19.1	SCG6284VLW	RCV-Tabulator Platform

2.3 Peripherals

Table 2-3 provides the information for the peripheral components used during the examination.

Table 2-3 Peripherals

Part Name	Model Number	Serial Number(s)	Description
Printer	Samsung Xpress M2020W Printer	CNB3M47DJ8	RCV Printer
Uninterruptable Power Supply	APC Backup UPS 600	4B191P18483	RCV External Power Supply

2.4 Test Support Equipment/Materials

The test materials listed in Table 2-4 were used during the examination.

Table 2-4 Test Support Materials

Component Name	Quantity	Description
8.5 x 11 Printer Paper	As needed	HP 200350
Thumb Drive	4	N/A

2.5 Technical Data Package

All relevant manufacturer-provided documentation provided during the examination is listed in Table 2-5.

Table 2-5 Documentation

Document Name	Document Number	Version
Technical Data Package	100	1.0
Tabulation Options for RCV	110	1.0
Process Ranked Choice Voting Contest	120	1.0
Expected Outcome RCV Test Sets Single-Winner	130	1.0
Expected Outcome RCV Test Sets Multi-Winner	140	1.0
ES&S Ballot Limitations & Maximum Testing Range	150	1.0
Universal RCV Tabulator User Guide Overview	160	1.0
Universal RCV Tabulator System Security Specifications	170	1.0
Installation Instructions for Universal RCV Tabulator - Windows OS	200	1.0
Installation Instructions for Universal RCV Tabulator - MacOS	210	1.0
Installation Instructions for Universal RCV Tabulator - Linux OS	220	1.0
HashCode Instructions - Windows OS	230	1.0
Hash Code Instructions - Linux OS	240	1.0
Hash Code Instruction – Linux OS	250	1.0
Configuration File Parameters	300	1.0
Universal RCV Tabulator Election Setup Checklist - MANUAL	310	1.0
Universal RCV Tabulator Election Setup Checklist -INTERACTIVE	320	1.0
Universal RCV Tabulator CVR Files	330	1.0
Universal RCV Tabulator System Election Data Form	340	1.0
Universal RCV Tabulator Config Files	350	1.0
Universal RCV Tabulator System Election Rules	360	1.0
Universal RCV Tabulator Logic & Accuracy Test	370	1.0
Universal RCV Tabulator Operator Election Processing Windows OS	400	1.0
Universal RCV Tabulator Operator Election Processing MacOS	410	1.0
Universal RCV Tabulator Operator Election Processing Linux	420	1.0
Universal RCV Tabulator System Operator Log Errors	430	1.0
Universal RCV Tabulator v1.1.0 Enhancements	900	1.0
NIST Cast Vote Records Common Data Format Specification	910	1.0

3.0 TEST PROCESS AND RESULTS

3.1 General Information

The examination of the RCV-Tabulator solution was performed by Pro V&V, Inc. If applicable, previous test results utilized as a basis for this test campaign are detailed below.

Previous Test Results

Document Name	Version	Issue Date
Letter Report, RCV-Tabulator v1.0.1	1	August 30, 2019

3.2 Test Approach

The RCV-Tabulators were configured as they would be for normal use during an election. RCVRC provided materials and support as needed to aid in the testing process.

3.3 Test Process and Results

Testing was accomplished by regression reviewing the modification to the system under test to the VVSG 1.0 requirements. Regression testing did not indicate that any further testing was warranted.

TDP Review

The objective of these reviews is to validate that the TDP is complete, versioned, and contains all needed documentation.

Summary Findings: *The addition of the RCV-Tabulator v1.1.0 TDP did not affect the ability of the baseline TDP to satisfy all VVSG 1.0 requirements. Additionally, the final RCV documentation editions provided sufficient detail to adequately attend to the added functionality.*

Physical Configuration Audit (PCA)

The PCA was conducted to ensure that the configurations and setups of the RCV-Tabulator submitted for testing matched the vendor's technical documentation, and confirmed that the documentation was sufficient for the user to install, validate, operate, and maintain the RCV. All changes made to the RCV-Tabulators during testing were subjected to re-examination. The PCA further established a configuration baseline of the software and hardware to be tested.

Summary Findings: *The addition of the three employed stand-alone and isolated RCV-Tabulator platforms matched the documentation and did not impact the ability of the modified RCV-Tabulator and baseline system as modified to satisfy the VVSG 1.0 requirements.*

Functional Configuration Audit (FCA)

The FCA Regression testing verified that the RCV-tabulator v1.1.0 continues to satisfy the prescribed requirements and performs the functions claimed in the system documentation.

Summary Findings: *The FCA Regression testing indicated that the v1.1.0 modifications did not impact the ability of the RCV-Tabulator to continue to satisfy all of the applicable pre-vote, vote, and post-vote test case requirements. The RCV-Tabulator demonstrated the ability to function as required and intended, and the v1.1.0 modifications did not impact the ability of the baseline system as modified to continue to satisfy the VVSG 1.0 requirements. FCA Regression testing did not indicate any further testing was warranted.*

Source Code Review

The RCV source code regression review was performed to verify that the software continues to conform to the vendor's specifications. The previous source code inspection also assessed the extent to which the code adhered to the VVSG 1.1.

Summary Findings: *The RCV-Tabulator stands as a VVSG 1.1 compliant source code modification to a VVSG 1.0 system, and continues to meet all of the VVSG 1.1 source code requirements.*

Security Testing

The security testing evaluated the continued effectiveness of the RCV-Tabulator in detecting, preventing, recording, reporting, and recovering from security threats. To evaluate the integrity of the RCV-Tabulator attempts were made to defeat the access controls and security measures documented in the system TDP.

Summary Findings: *A security regression review was performed on the RCV-Tabulator, which incorporates the security policies of the baseline system. The RCV-Tabulator was found to have an applied level of security that complies with the verified VVSG 1.0 security provisions. Additionally, the presence of the RCV-Tabulator v1.1.0 modifications do not impact the ability of the baseline system as modified to satisfy the VVSG 1.0 security requirements.*

Usability Testing

The usability testing encompassed a range of methods that examined how users in the target audience actually interact with the RCV-Tabulator.

Summary Findings: *The RCV-Tabulator v1.1.0 has operations and controls that can easily be employed by users in the target audience making use of the system TDP.*

Accessibility Testing

The accessibility testing measured characteristics that indicated the degree to which the RCV-Tabulator is available to, and usable by, users in the target audience.

Summary Findings: *The RCV-Tabulator is accessible to users with a basic understanding of computers and election management.*

System Integration

The system level testing addressed the integrated operations of the RCA-Tabulator v1.1.0 with the baseline system, and tested the RCV-Tabulator v1.1.0 capabilities as part of the baseline system as a whole. This testing validated the seamless functional integration and data inputs/outputs of the baseline system with the RCV-Tabulator. For this testing the RCV-Tabulators were configured and operated as for normal operations.

Summary Findings: *Cast Vote Records (CVR) that were generated by the baseline system were imported into the RCV-Tabulator and processed through tabulation to report generation. In addition, ES&S CVR's were successfully converted to CDF and results were accurately tabulated. No system issues were noted during testing*

Accuracy Testing

RCV-Tabulator accuracy was based upon the totality of all testing.

Summary Findings: *During testing, actual results were validated against expected results. Including, the conversion of ES&S CVR's into CDF with the results accurately tabulated. No accuracy issues were noted throughout the test campaign.*

Photograph 1: Dell Inspiron Desktop

Photograph 2: HP Elite Book Laptop

Photograph 3: Apple Macbook

4.0 CONCLUSION

Based on the testing performed and the results obtained, the modified RCV-Tabulator v1.1.0 solution identified in this test report is believed to meet the applicable requirements set forth by the VVSG 1.0.

Letter Report from Pro V&V after testing of Universal RCV Tabulator v. 1.2.0

TEST REPORT

FOR

EXAMINATION OF RANKED CHOICE VOTING TABULATOR v1.2.0
The Election Administration Resource Center dba Ranked Choice Voting Resource Center

TO

VOLUNTARY VOTING SYSTEM GUIDELINES (VVSG) 1.0

Prepared by: Wendy Owens
Wendy Owens, VSTL Program Manager

December 04, 2020

Pro V&V, Inc.
6705 Odyssey Drive, Suite C
Huntsville, AL 35806
256-713-1111

PRO V&V, Inc.

TABLE OF CONTENTS

1.0 INTRODUCTION 1
1.1 References 1
1.2 Terms and Abbreviations 2
1.3 Background 2
1.4 Scope 2
2.0 SYSTEM IDENTIFICATION 3
2.1 Software 3
2.2 Hardware 4
2.3 Peripherals 4
2.4 Test Support Equipment/Materials 4
2.5 Technical Data Package 5
3.0 TEST PROCESS AND RESULTS 6
3.1 General Information 6
3.2 Test Approach 6
3.3 Test Process and Results 6
4.0 CONCLUSION 9

1.0 INTRODUCTION

Pro V&V performed an examination effort on the RCVRC Ranked Choice Voting Tabulator (RCV-Tabulator 1.2.0), which is a modification to the RCV-Tabulator 1.1.0. Which is itself a modification to the baseline RCV-Tabulator 1.0.1. Pro V&V performed a functional regression examination of the submitted RCV-Tabulator v1.2.0 configured as it will be for normal use.

This and the baseline RCV-Tabulator testing campaigns were performed with the support of RCVRC, to verify that the submitted modification-additions meet the certification requirements found in the Voluntary Voting System Guidelines Version 1.0 (VVSG 1.0).

Included voting system additions are:

Vendor	System	Versions
ES&S*	EVS	5.0.0.0 to 6.0.4.0
Unisyn	OpenElect	2.1, 2.1.0.3
Clear Ballot	ClearVote	1.4, 1.5, 2.0
Hart IC	Verity	2.0, 2.2, 2.2.1, 2.2.2, 2.3, 2.3.3, 2.3.4, 2.4, 2.5
Dominion	Democracy Suite	5.2, 5.2.18.2*, 5.10.11.24*

**State only certified version*

The regression testing examined modifications are listed in the following document:

- Universal RCV Tabulator v1.2.0 Enhancements, v1, dated 10-13-2020

This Test Report documents the procedures followed, the results obtained, and the conclusions drawn from this examination.

1.1 References

The documents listed below were utilized in the development of the Test Report.

- Universal RCV Tabulator v1.2.0 Enhancements, v1, dated 10-13-2020
- Implementation Statement
- Election Assistance Commission 2005 Voluntary Voting System Guidelines (VVSG) Version 1.0, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission 2015 Voluntary Voting System Guidelines (VVSG) Version 1.1, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission Testing and Certification Program Manual, Version 2.0
- Election Assistance Commission Voting System Test Laboratory Program Manual, Version 2.0
- National Voluntary Laboratory Accreditation Program NIST Handbook 150, 2016 Edition, "NVLAP Procedures and General Requirements (NIST Handbook 150)", dated July 2016
- National Voluntary Laboratory Accreditation Program NIST Handbook 150-22, 2008 Edition, "Voting System Testing (NIST Handbook 150-22)", dated May 2008

- United States 107th Congress Help America Vote Act (HAVA) of 2002 (Public Law 107-252), dated October 2002
- Pro V&V, Inc. Quality Assurance Manual, Revision 7.0
- System Technical Data Package

1.2 Terms and Abbreviation

This subsection lists terms and abbreviations relevant to this examination:

"COTS" – Commercial Off-The-Shelf

"CVR" – Cast Vote Record

"EAC" – United States Election Assistance Commission

"EBM" -- Electronically-assisted Ballot Marker

"FCA" – Functional Configuration Audit

"HAVA" – Help America Vote Act

"ISO" – International Organization for Standardization

"PCA" – Physical Configuration Audit

"QA" – Quality Assurance

"TDP" – Technical Data Package

"VSTL" – Voting System Test Laboratory

"VVSG" – Voluntary Voting System Guidelines

1.3 Background

The RCV-Tabulator 1.2.0 is a modification to the RCV-Tabulator 1.1.0. The RCV-Tabulator 1.1.0 voting system is a complete voting system composed of EMS software applications, central count devices, and polling location devices, which is baselined from the RCV-Tabulator 1.0.1 utilizing the ES&S EVS 6.0.4.0 EAC certified voting system.

The Universal RCV-Tabulator v1.2.0 processes digital cast vote records and election data from various voting machine vendors. Supported formats include ES&S (.xlsx), Clear Ballot (xlsx), Hart InterCivic (.json), Dominion (.json), Unisyn (Common Data Format), and NIST Common Data Format (.json). The Tabulator can ingest election data and tabulate single-winner and multi-winner ranked choice voting elections according to many rules used in current state, county, and city jurisdictions in the USA. The Tabulator generates various reports including a tabular results report in .csv format, a results summary .json file (for visualizing), a Common Data Format JSON CVR Record Report containing round-by-round "snapshots" of each CVR for each round and a tabulation audit log.

1.4 Scope

The scope of testing focused on regression testing of the RCV-Tabulator v1.2.0. To this end, election data was imported and the results tabulated using the RCV-Tabulator. Additionally, CVRs were converted to Common Data Format (CDF) and CDFs files were tabulated. This followed as below.

Table 1-1 Tabulations

Vendor	Cast Vote Record	Common Data Format
ES&S	CVR (.xlsx)	CDF
Clear Ballot	CVR (.xlsx)	CDF
Unysin	N/A	CDF
Hart InterCivic	CVR (.json)	CDF
Dominion	CVR (.json)	CDF

2.0 SYSTEM IDENTIFICATION

This section contains detailed descriptions of the submitted RCV-Tabulator solution including: platforms for required technical configuration testing, items tested during the evaluation (including all software, hardware, and peripherals, both proprietary and COTS), and any test support equipment or materials necessary for test performance.

Application Summary

The RCV-Tabulator was configured and setup as it would be for normal operations as per the TDP. RCVRC provided materials and phone support as needed to aid in the testing process.

For testing, the RCV-Tabulator resided on three independent test support platforms with the Cast Vote Records (CVR) originating from the added voting systems. The modified system supported voting variations are:

Table 2-1 Voting Variations

Voting Variations	ES&S	Unisyn	Clear Ballot	Hart InterCivic	Dominion
Closed primaries	Yes	Yes	Yes	Yes	Yes
Open primaries	Yes	Yes	Yes	Yes	Yes
Partisan offices	Yes	Yes	Yes	Yes	Yes
Non-partisan offices	Yes	Yes	Yes	Yes	Yes
Write-in voting	Yes	Yes	Yes	Yes	Yes
Primary presidential delegation nominations	Yes	Yes	Yes	Yes	Yes
Ballot rotation	Yes	Yes	Yes	Yes	Yes
Straight party voting	Yes	Yes	Yes	Yes	Yes
Cross-party endorsement	Yes	No	Yes	Yes	Yes
Split precincts	Yes	Yes	Yes	Yes	Yes
Vote for N of M	Yes	Yes	Yes	Yes	Yes
Recall issues, with options	No	Yes	Yes	Yes	Yes
Cumulative voting	No	No	No	Yes	No
Ranked order voting (w/RCV-Tabulator)	Yes	Yes	Yes	Yes	Yes
Provisional or challenged ballots	Yes	Yes	Yes	Yes	Yes

2.1 Software

The software listed in Table 2-2 was used during the examination of the RCV-Tabulator. Table 2-1 provides the component name, software/firmware version and a SHA-512 Hash, if calculated, during the examination.

Table 2-2 Software

Component Name	Version	Unique Identifier (SHA 512 Hash Value)
RCV Tabulator	Windows 1.2.0	c7704870b5a442ec9fb7212427377dfb822a8a9ac0d4310b68edfd d9e88fbd55821f962072a293edeaf64529ef7288293dc9e854ed09d 31627004c304f00f689
RCV Tabulator	OSX 1.2.0	62e977774aac5149f537ab2d121e25899ff1f8f7a240c02d801198f 00284fc6b93e5676164467f31fb5da3bd4e009451d34c58a95d50b f5c032bb1a724fb136b
RCV Tabulator	Linux 1.2.0	fd9b728b685e72d9250c6b987b1d09900901942701d1dc46a3133 b6a9dbf75064927889a7dea36970902b28e1bc5d5de79e27d4e9a0 2693f850261995cafe00b

2.2 Hardware

The tables below provide hardware component name, description, model number, and the serial number for the units that were examined.

Table 2-3 Hardware

Component Name	Model/Version Number	Operating System	Serial Number(s)	Description
RCV-Tabulator	Dell Inspiron Desktop	Windows 10 Pro	1QLJDW2	RCV-Tabulator Platform
RCV-Tabulator	Apple MacBook	OS 10.13	CO2PP2BAGCN3	RCV-Tabulator Platform
RCV-Tabulator	HP Elite Book Laptop 840	Mint 19.1	SCG6284VLW	RCV-Tabulator Platform

2.3 Peripherals

Table 2-4 provides the information for the peripheral components used during the examination.

Table 2-4 Peripherals

Part Name	Model Number	Serial Number(s)	Description
Printer	Samsung Xpress M2020W Printer	CNB3M47DJ8	RCV Printer
Uninterruptable Power Supply	APC Backup UPS 600	4B191P18483	RCV External Power Supply

2.4 Test Support Equipment/Materials

The test materials listed in Table 2-5 were used during the examination.

Table 2-5 Test Support Materials

Component Name	Quantity	Description
8.5 x 11 Printer Paper	As needed	HP 200350
Thumb Drive	4	N/A

2.5 Technical Data Package

All relevant manufacturer-provided documentation that was provided during the examination is listed in the table below.

Table 2-6 Documentation

Document Name	Document Number	Version
Technical Data Package	100	1.0
Tabulation Options for RCV	110	1.0
Process Ranked Choice Voting Contest	120	1.0
Expected Outcome RCV Test Sets Single-Winner	130	1.0
Expected Outcome RCV Test Sets Multi-Winner	140	1.0
ES&S Ballot Limitations & Maximum Testing Range	150	1.0
Universal RCV Tabulator User Guide Overview	160	1.0
Universal RCV Tabulator System Security Specifications	170	1.0
Installation Instructions for Universal RCV Tabulator - Windows OS	200	1.0
Installation Instructions for Universal RCV Tabulator - MacOS	210	1.0
Installation Instructions for Universal RCV Tabulator - Linux OS	220	1.0
HashCode Instructions - Windows OS	230	1.0
Hash Code Instructions - Linux OS	240	1.0
Hash Code Instruction – Linux OS	250	1.0
Configuration File Parameters	300	1.0
Universal RCV Tabulator Election Setup Checklist - MANUAL	310	1.0
Universal RCV Tabulator Election Setup Checklist -INTERACTIVE	320	1.0
Universal RCV Tabulator CVR Files	330	1.0
Universal RCV Tabulator System Election Data Form	340	1.0
Universal RCV Tabulator Config Files	350	1.0
Universal RCV Tabulator System Election Rules	360	1.0
Universal RCV Tabulator Logic & Accuracy Test	370	1.0
Universal RCV Tabulator Operator Election Processing Windows OS	400	2.0
Universal RCV Tabulator Election Processing Windows OS	401	1.0
Universal RCV Tabulator Operator Election Processing MacOS	410	2.0
Universal RCV Tabulator Election Processing MacOS	411	1.0
Universal RCV Tabulator Operator Election Processing Linux	420	2.0
Universal RCV Tabulator Election Processing Linux	421	1.0
Universal RCV Tabulator System Operator Log Errors	430	1.0
Hints for Contest Info tab	440	1.0
Universal RCV Tabulator v1.1.0 Enhancements	900	1.0
NIST Cast Vote Records Common Data Format Specification	910	1.0

3.0 TEST PROCESS AND RESULTS

3.1 General Information

The examination of the RCV-Tabulator solution was performed by Pro V&V, Inc. If applicable, previous test results utilized as a basis for this test campaign are detailed below.

Table 3-1 Previous Test Results

Document Name	Version	Issue Date
Letter Report, RCV-Tabulator v1.0.1	1	August 30, 2019
Test Report for Examination of Ranked Choice Voting Tabulator v1.1.0	1	May 19, 2020

3.2 Test Approach

The RCV-Tabulators were configured as they would be for normal use during an election. RCVRC provided materials and support as needed to aid in the testing process.

3.3 Test Process and Results

Testing was accomplished by regression reviewing the modification to the system under test to the VVSG 1.0 requirements. Regression testing did not indicate that any further testing was warranted.

TDP Review
The objective of these reviews is to validate that the TDP is complete, versioned, and contains all needed documentation.

Summary Findings: *The addition of the RCV-Tabulator v1.2.0 TDP did not affect the ability of the baseline TDP to satisfy all VVSG 1.0 requirements. Additionally, the final RCV documentation editions provided sufficient detail to adequately attend to the added functionality.*

Physical Configuration Audit (PCA)
The PCA was conducted to ensure that the configurations and setups of the RCV-Tabulator submitted for testing matched the vendor's technical documentation, and confirmed that the documentation was sufficient for the user to install, validate, operate, and maintain the RCV-Tabulator. All changes made to the RCV-Tabulators during testing were subjected to re-examination. The PCA further established a configuration baseline of the software and hardware to be tested.

Summary Findings: *The final iteration of the three employed stand-alone and isolated RCV-Tabulator platforms matched the documentation and did not impact the ability of the modified RCV-Tabulator and baseline system as modified to satisfy the VVSG 1.0 requirements.*

Functional Configuration Audit (FCA)
The FCA regression testing verified that the RCV-tabulator v1.2.0 continues to satisfy the prescribed requirements and performs the functions claimed in the system documentation.

Summary Findings: *The FCA Regression testing indicated that the v1.2.0 modifications did not impact the ability of the RCV-Tabulator to continue to satisfy all of the applicable pre-vote, vote, and post-vote test case requirements. The RCV-Tabulator demonstrated the ability to function as required and intended, and the v1.2.0 modifications did not impact the ability of the baseline system as modified to continue to satisfy the VVSG 1.0 requirements. FCA Regression testing did not indicate any further testing was warranted.*

Source Code Review

The RCV source code regression review was performed to verify that the software continues to conform to the vendor's specifications. The previous source code inspection also assessed the extent to which the code adhered to the VVSG 1.1.

Summary Findings: *The RCV-Tabulator stands as a VVSG 1.1 compliant source code modification to a VVSG 1.0 system, and continues to meet all of the VVSG 1.1 source code requirements.*

Security Testing

The security testing evaluated the continued effectiveness of the RCV-Tabulator in detecting, preventing, recording, reporting, and recovering from security threats. To evaluate the integrity of the RCV-Tabulator attempts were made to defeat the access controls and security measures documented in the system TDP.

Summary Findings: *A security regression review was performed on the RCV-Tabulator, which incorporates the security policies of the baseline system. The RCV-Tabulator was found to have an applied level of security that complies with the verified VVSG 1.0 security provisions. Additionally, the presence of the RCV-Tabulator v1.2.0 modifications do not impact the ability of the baseline system as modified to satisfy the VVSG 1.0 security requirements.*

Usability Testing

The usability testing encompassed a range of methods that examined how users in the target audience actually interact with the RCV-Tabulator.

Summary Findings: *The RCV-Tabulator v1.2.0 has operations and controls that can easily be employed by users in the target audience making use of the system TDP.*

Accessibility Testing

The accessibility testing measured characteristics that indicated the degree to which the RCV-Tabulator is available to, and usable by, users in the target audience.

Summary Findings: *The RCV-Tabulator is accessible to users with a basic understanding of computers and election management.*

System Integration

The system level testing addressed the integrated operations of the RCA-Tabulator v1.2.0 with the baseline system, and tested the RCV-Tabulator v1.2.0 capabilities as part of the baseline system as a whole. This testing validated the seamless functional integration and data inputs/outputs of the baseline system with the RCV-Tabulator. For this testing the RCV-Tabulators were configured and operated as for normal operations.

Summary Findings: *Cast Vote Records (CVR) that were generated by the baseline system were imported into the RCV-Tabulator and processed through tabulation to report generation. In addition, as noted below CVR's were successfully converted to CDF's and results were accurately tabulated. No system issues were noted during testing*

Table 3-2 Tabulations

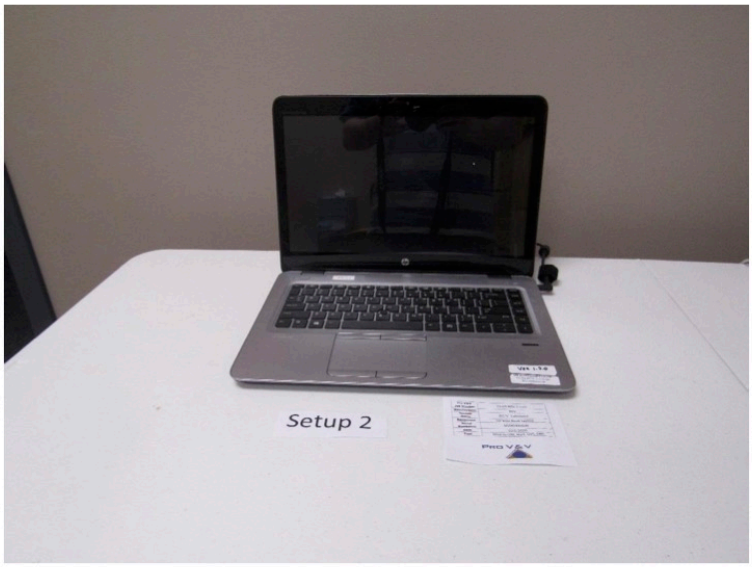
Vendor	Cast Vote Record	Common Data Format
ES&S	CVR (.xlsx)	CDF
Clear Ballot	CVR (.xlsx)	CDF
Unysin	N/A	CDF
Hart InterCivic	CVR (.json)	CDF
Dominion	CVR (.json)	CDF

Accuracy Testing

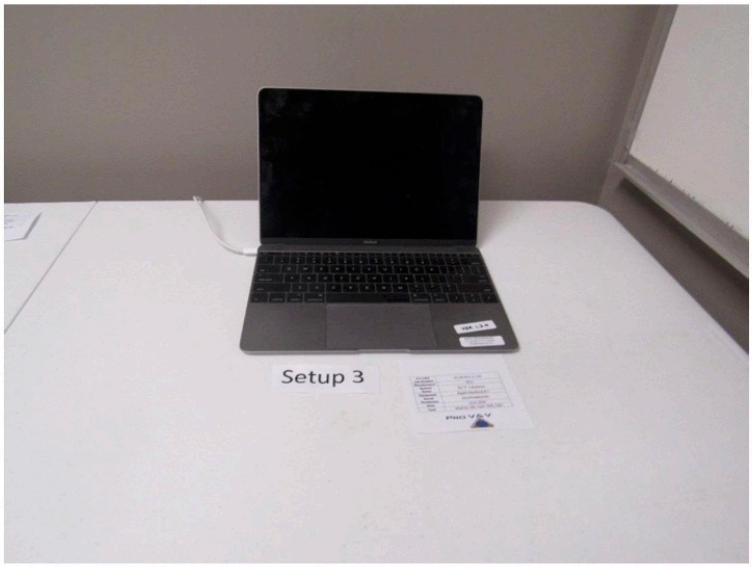
RCV-Tabulator accuracy was based upon the totality of all testing.

Summary Findings: *During testing, actual results were validated against expected results. Including, the conversion of CVR's into CDF's with the results accurately tabulated. No accuracy issues were noted throughout the test campaign.*

Photograph 1: Dell Inspiron Desktop



Photograph 2: HP Elite Book Laptop



Photograph 3: Apple MacBook

4.0 CONCLUSION

Based on the testing performed and the results obtained, the modified RCV-Tabulator v1.2.0 solution identified in this test report meets the requirements set forth by the VVSG 1.0.

Appendix C: Related Standards, Models and Information

VVSG certification

This Appendix analyzes voting variations tested for/included in current VVSG certification documents published by the EAC. We took screenshots from the EAC Certification Document for ES&S's EVS 6.0.4.0 to capture voting methods as described in certification documents.

Feature/Characteristic
Slate & Group Voting: one selection votes the slate.
Ballot Rotation:
Rotation of Names within an Office; define all supported rotation methods for location on the ballot and vote tabulation/reporting
Straight Party Voting:
Straight Party: A single selection for partisan races in a general election
Straight Party: Vote for each candidate individually
Straight Party: Modify straight party selections with crossover votes
Straight Party: A race without a candidate for one party
Straight Party: N of M race (where "N">1)
Straight Party: Excludes a partisan contest from the straight party selection
Cross-Party Endorsement:
Cross party endorsements, multiple parties endorse one candidate.
Split Precincts:
Split Precincts: Multiple ballot styles
Split Precincts: P & M system support splits with correct contests and ballot identification of each split
Split Precincts: DRE matches voter to all applicable races.
Split Precincts: Reporting of voter counts (# of voters) to the precinct split level; Reporting of vote totals is to the precinct level
Vote N of M:
Vote for N of M: Counts each selected candidate, if the maximum is not exceeded.
Vote for N of M: Invalidates all candidates in an overvote (paper)
Recall Issues, with options:
Recall Issues with Options: Simple Yes/No with separate race/election. (Vote Yes or No Question)
Recall Issues with Options: Retain is the first option, Replacement candidate for the second or more options (Vote 1 of M)
Recall Issues with Options: Two contests with access to a second contest conditional upon a specific vote in contest one. (Must vote Yes to vote in 2nd contest.)
Recall Issues with Options: Two contests with access to a second contest conditional upon any vote in contest one. (Must vote Yes to vote in 2nd contest.)

Closed Primary
Primary: Closed
Open Primary
Primary: Open Standard (provide definition of how supported)
Primary: Open Blanket (provide definition of how supported)
Partisan & Non-Partisan:
Partisan & Non-Partisan: Vote for 1 of N race
Partisan & Non-Partisan: Multi-member ("vote for N of M") board races
Partisan & Non-Partisan: "vote for 1" race with a single candidate and write-in voting
Partisan & Non-Partisan "vote for 1" race with no declared candidates and write-in voting
Write-In Voting:
Write-in Voting: System default is a voting position identified for write-ins.
Write-in Voting: Without selecting a write in position.
Write-in: With No Declared Candidates
Write-in: Identification of write-ins for resolution at central count
Primary Presidential Delegation Nominations & Slates:
Primary Presidential Delegation Nominations: Displayed delegate slates for each presidential party

Cumulative Voting
Cumulative Voting: Voters are permitted to cast, as many votes as there are seats to be filled for one or more candidates. Voters are not limited to giving only one vote to a candidate. Instead, they can put multiple votes on one or more candidate.
Ranked Order Voting
Ranked Order Voting: Voters can write in a ranked vote.
Ranked Order Voting: A ballot stops being counting when all ranked choices have been eliminated

Ranked Order Voting: A ballot with a skipped rank counts the vote for the next rank.
Ranked Order Voting: Voters rank candidates in a contest in order of choice. A candidate receiving a majority of the first choice votes wins. If no candidate receives a majority of first choice votes, the last place candidate is deleted, each ballot cast for the deleted candidate counts for the second choice candidate listed on the ballot. The process of eliminating the last place candidate and recounting the ballots continues until one candidate receives a majority of the vote
Ranked Order Voting: A ballot with two choices ranked the same, stops being counted at the point of two similarly ranked choices.
Ranked Order Voting: The total number of votes for two or more candidates with the least votes is less than the votes of the candidate with the next highest number of votes, the candidates with the least votes are eliminated simultaneously and their votes transferred to the next-ranked continuing candidate.

Overvotes (must support for specific type of voting system)
Overvotes: P & M: Overvote invalidates the vote. Define how overvotes are counted.
Overvotes: DRE: Prevented from or requires correction of overvoting.
Overvotes: If a system does not prevent overvotes, it must count them. Define how overvotes are counted.
Overvotes: DRE systems that provide a method to data enter absentee votes must account for overvotes.
Undervotes
Undervotes: System counts undervotes cast for accounting purposes
Blank Ballots
Totally Blank Ballots: Any blank ballot alert is tested.
Totally Blank Ballots: If blank ballots are not immediately processed, there must be a provision to recognize and accept them
Totally Blank Ballots: If operators can access a blank ballot, there must be a provision for resolution.

Election Business Process Model BPMN Diagrams, Use Cases of 1500-107 in Election Business Process Model BPMN Diagrams

Voting Methods and Tabulation Methods Use Cases at the Election Business Process Layer

These diagrams are the work product of the NIST-EAC Election Modeling Working Group. The full body of work may be found at:
https://pages.nist.gov/ElectionModeling/
The diagrams index is here:
https://pages.nist.gov/ElectionModeling/diagrams.html#Diagrams___18_5_1_11940316_1503510989936_603561_55115

The following diagrams are included here to provide the business process context in which voting methods or tabulation models have use cases, from the election business process model point of view.

Tally

The process named Tally Ballots encloses the creation of the CVR vote selection data set and tabulation of the CVRs.
Process 3.1.3 Tally, Tally the ballot selections in order to report election results

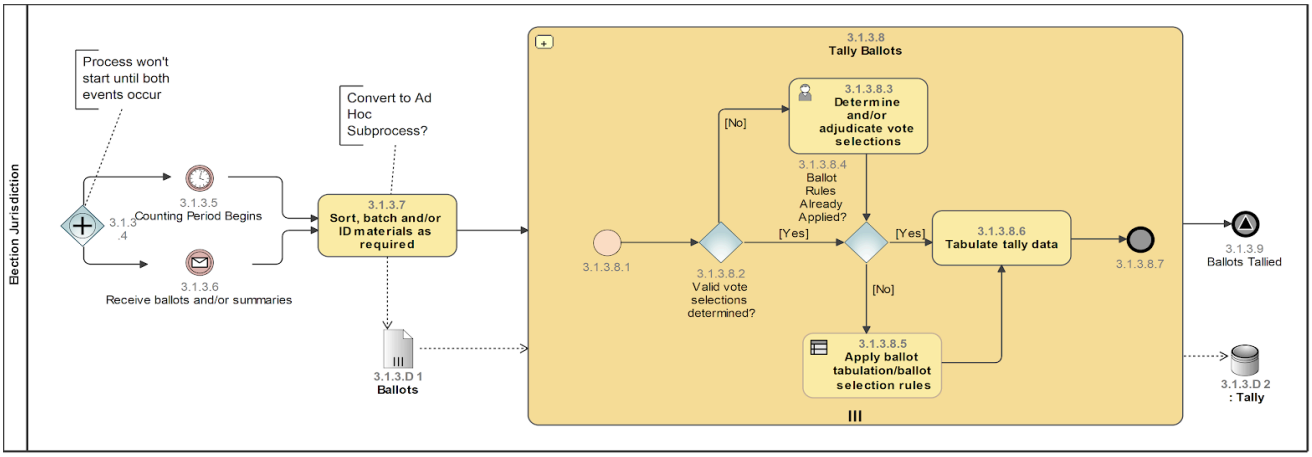


Figure 44. Tally Ballots process.

Logic And Accuracy Testing

1.5.7.7 Validate Reports Validate that the set of artifacts produced during the logic and accuracy testing matches the expected outcome

A sub-process of the Logic and Accuracy Testing model, “1.5.7.7 Validate Reports Validate that the set of artifacts produced during the logic and accuracy testing matches the expected outcome.”, requires validation

of configuration for a voting method and therefore uses tabulation to validate the results that are based on an input test ballot deck.

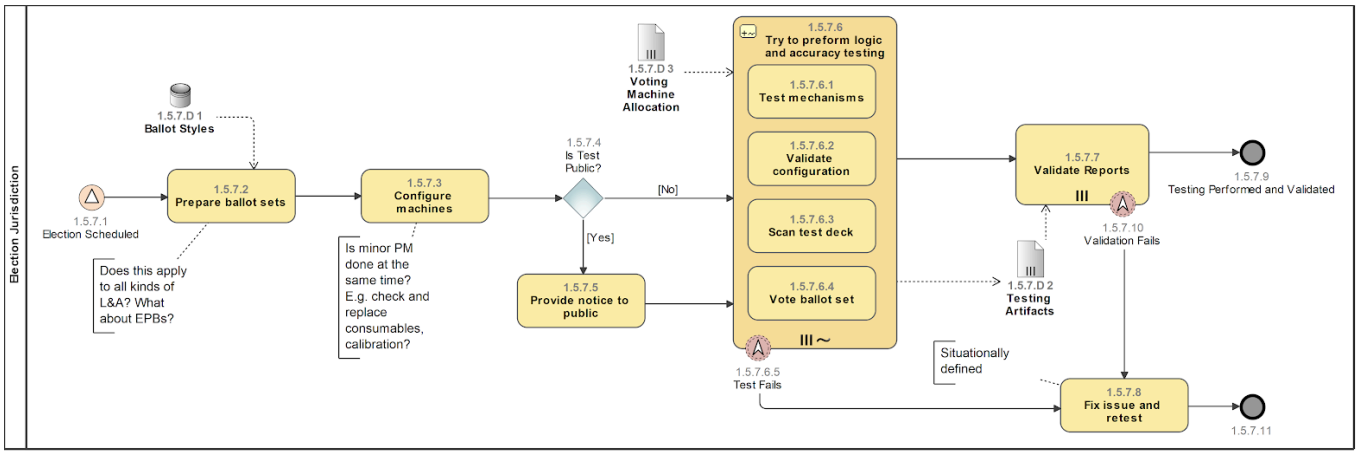


Figure 45. Logic and Accuracy Testing process.

Report Results

Process 3.1.4 Report Results

Maintaining an aggregate vote selection database as indicated in this process corresponds to the 1500-107 section 3.1.4 tabulation scenarios use cases where aggregation of results produced at distributed locations are collected into a central “Tally Database”, for example at a state or county level, or for batch tabulation use cases.

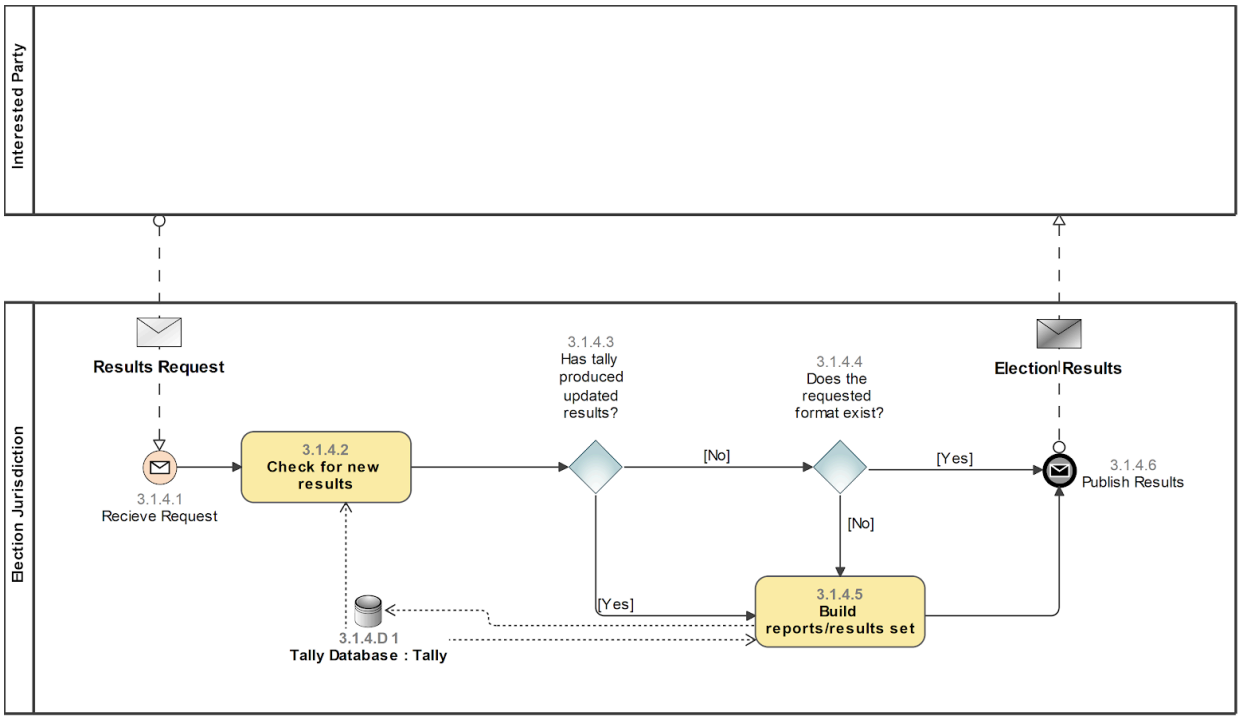


Figure 46. Report results process.

Verify Vote Count

Process 3.1.5 Verify Vote Count Ensure that the voters' ballots were counted as cast.

3.1.5.5 Perform manual or machine vote recount

This process includes tabulation. However, the tabulation may be performed either via hand-count or by a tabulator and is usually a recount or an audit where outcomes of original count to subsequent count or in general comparison of counts on the same input vote selection dataset are performed. Necessarily, the voting method and tabulation models are factors in producing these counts and determining correct outcome. This use case corresponds to this 1500-107 tabulation scenarios for recounts and audits.

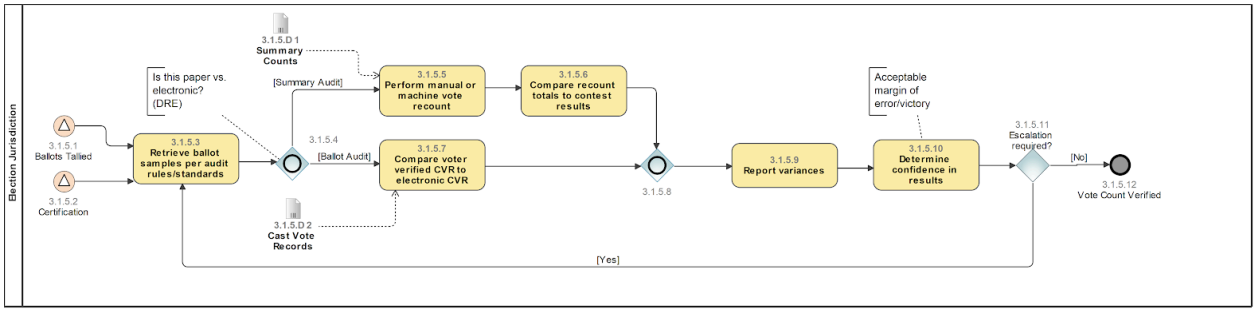
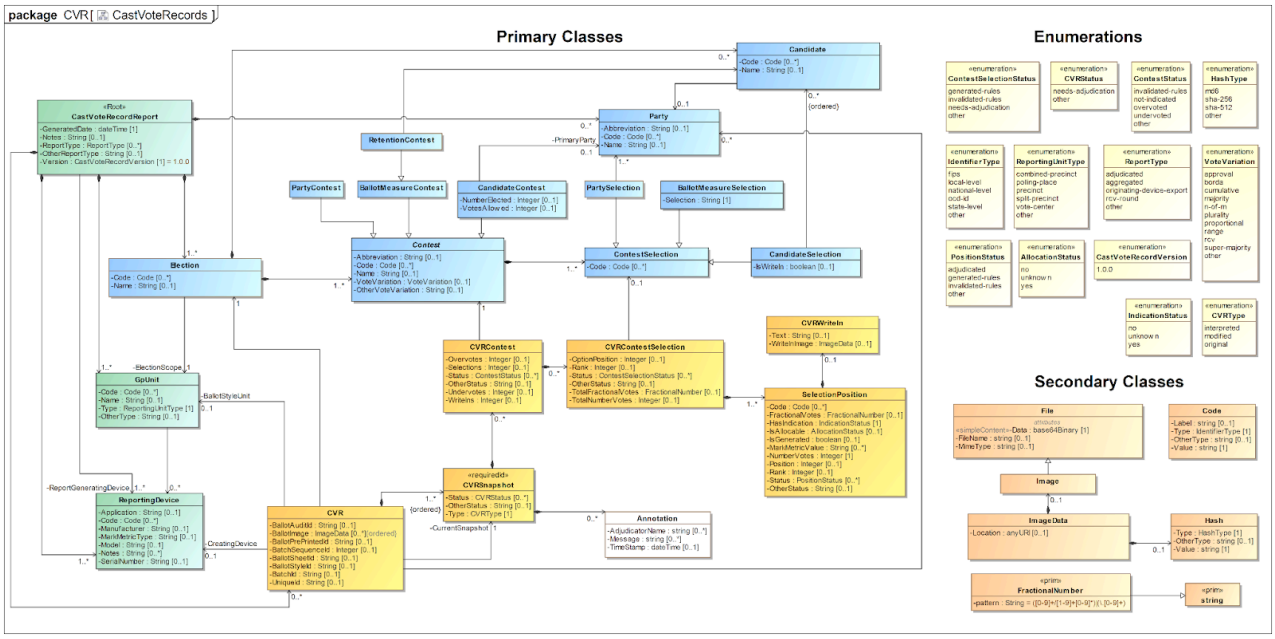


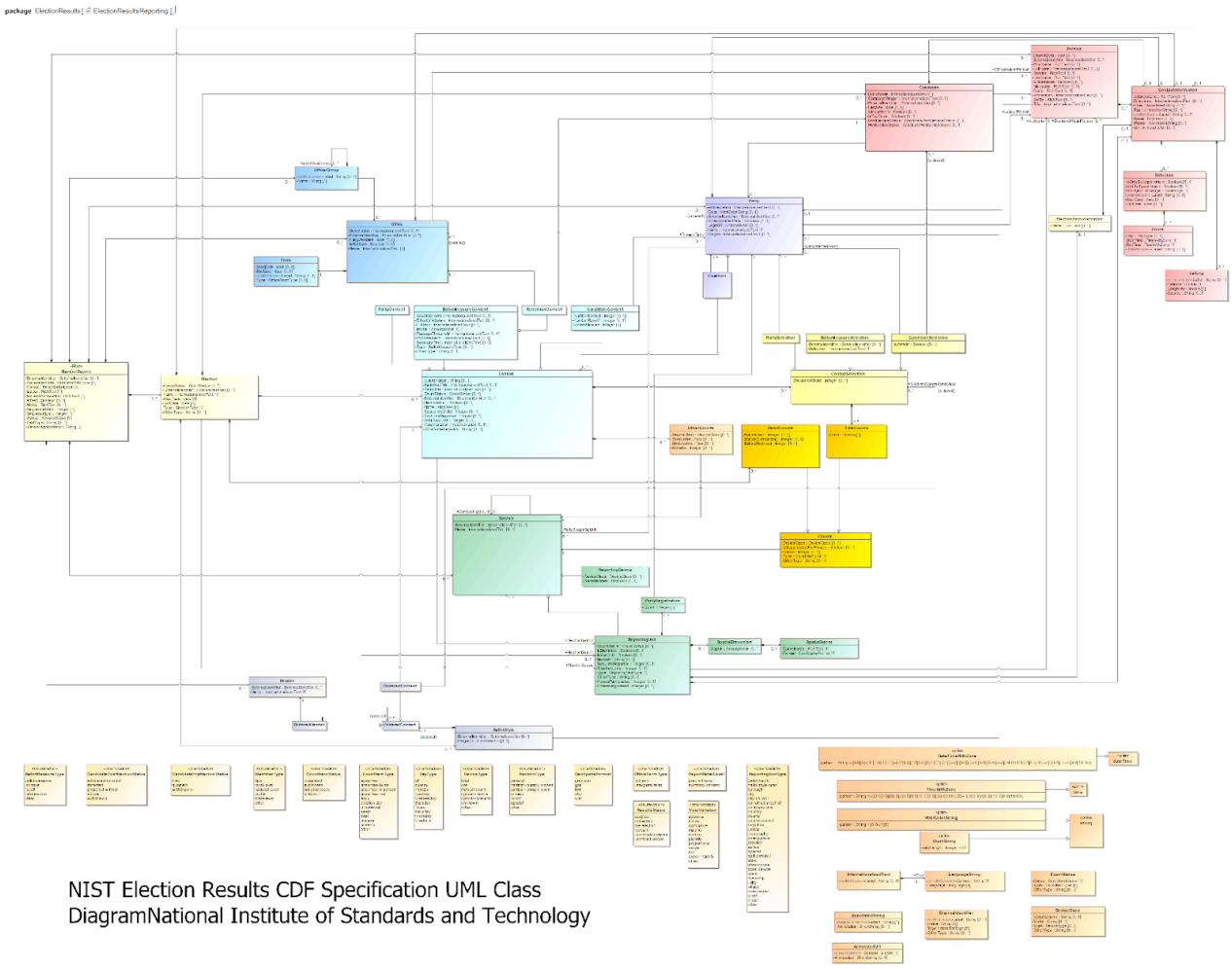
Figure 47. Verify vote count process.

NIST 1500-103 Cast Vote Record



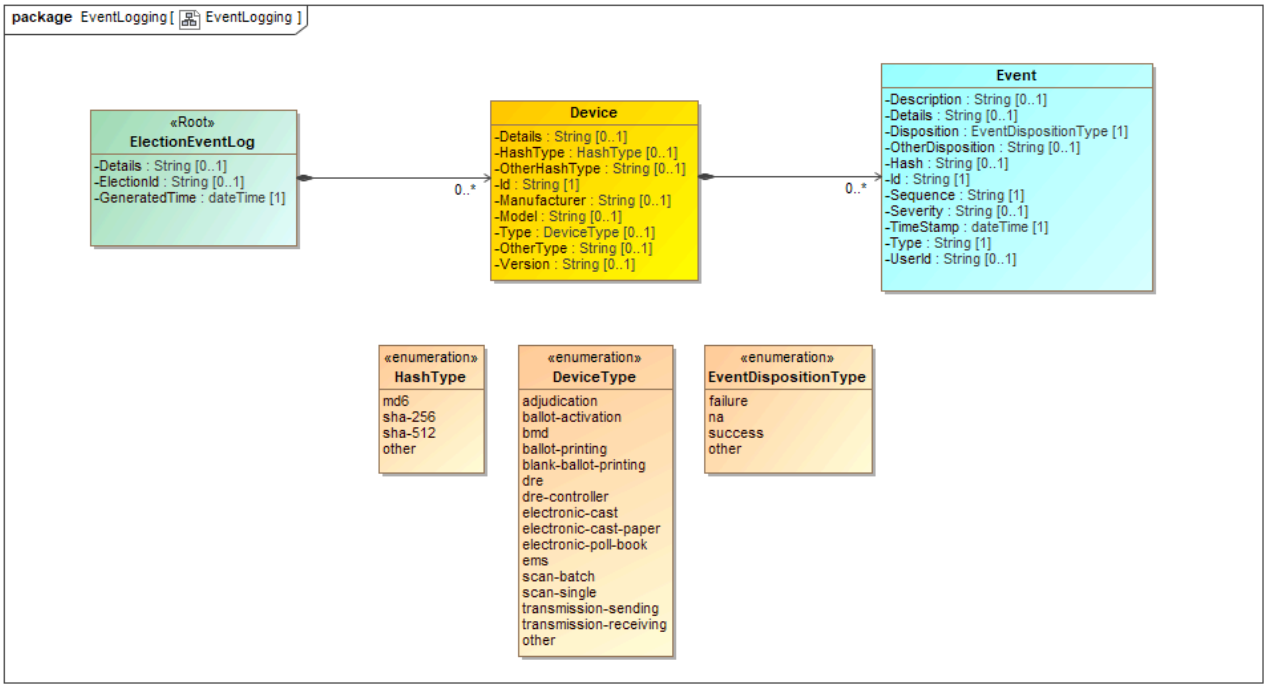
Full specification can be retrieved at https://doi.org/10.6028/NIST.SP.1500-103

NIST 1500-100 Election Results Reporting



Full specification can be retrieved at https://doi.org/10.6028/NIST.SP.1500-100r2

NIST 1500-101 Election Event Logging



Full specification can be retrieved at https://doi.org/10.6028/NIST.SP.1500-101