Documentation for Cast Vote Records UML Model Version 18

25 April 2017

This document describes the NIST Interoperability public working group cast vote records (CVR) UML model version 17. This model supports 3 general use cases:

1. Interoperable exports of CVRs from devices such as scanners, DREs (direct record electronic), or other devices that create CVRs and perform contest rule post-processing.
    a. Resultant interoperable import into tabulators, such as election management systems (EMS), or import into auditing systems, or import into adjudication systems.
2. Interoperable exports of aggregated collections of CVRs from aggregating devices such as EMS.
3. Interoperable export of CVRs from adjudication devices.

A JSON schema can be generated from the UML model so that scanners and other devices can export and import CVRs in JSON and validate JSON usage against the schema. It should be straightforward to convert from the JSON format to other formats such as comma-separated value.

To support auditing, the model contains the following:

- Support for digitally-signed exports of CVR files.
- Support for ballot-level auditing, i.e., there is an identifier that can be linked to an ID (identifier) on the corresponding paper ballot provided the scanner creating the CVR can impress an identifier on the ballot as it is scanned.
- If the scanner does contest-rule post-processing of the ballot selections, indications of what actions the scanner took, e.g., "voter selection overridden by contest rules".
- Indications of marginal marks, mark quality/density (if scanner is capable).
- A CVR can include signed/hashed references to an associated image of the ballot or images of write-ins made by the voter on a paper ballot.
- Improved handling of multi-sheet ballots so that one can more accurately count the number of multi-sheet ballots scanned even if page 1 or any other page is missing.
- Capability to include batch IDs or other batch information.
- Support to include adjustments to vote selections made by adjudicators.

1. Examples of Structure

Several examples of usage structure may clarify the model and documentation. Briefly, the exports/imports are structured as optionally signed collections of CVRs that can be each linked to an election (e.g., a precinct with associated contests, candidates, ballot selections, parties, etc.), a generating device (e.g., a scanner or BMD (Ballot Marking Device) in use at the precinct), and the tabulating device (e.g., an EMS that aggregates and tabulates). The general file structure is as follows:

CastVoteRecordReport:

- Date/Time of report generation, Notes, Version, Digital signature
- Election associated with the CVRs:
  - Identification of the election, e.g., MD General Election Precinct A1
  - Collection of Candidate definitions, each with a unique identifier
  - Collection of Contest and BallotSelection definitions
- Device generating the export
- CastVoteRecord 1
- CastVoteRecord 2
- CastVoteRecord 3
- …

CVR general file structure

As with the NIST 1500-100 Election Results Reporting specification (https://www.nist.gov/itl/voting/nist-election-results-common-data-format-specification), definitions for the candidates and contests and ballot selections are defined once at the Election level, and the CVRs link to these definitions rather than repeating them so as to reduce file sizes.

An example of the structure for each CVR within the export/import file is as follows:

CastVoteRecord:

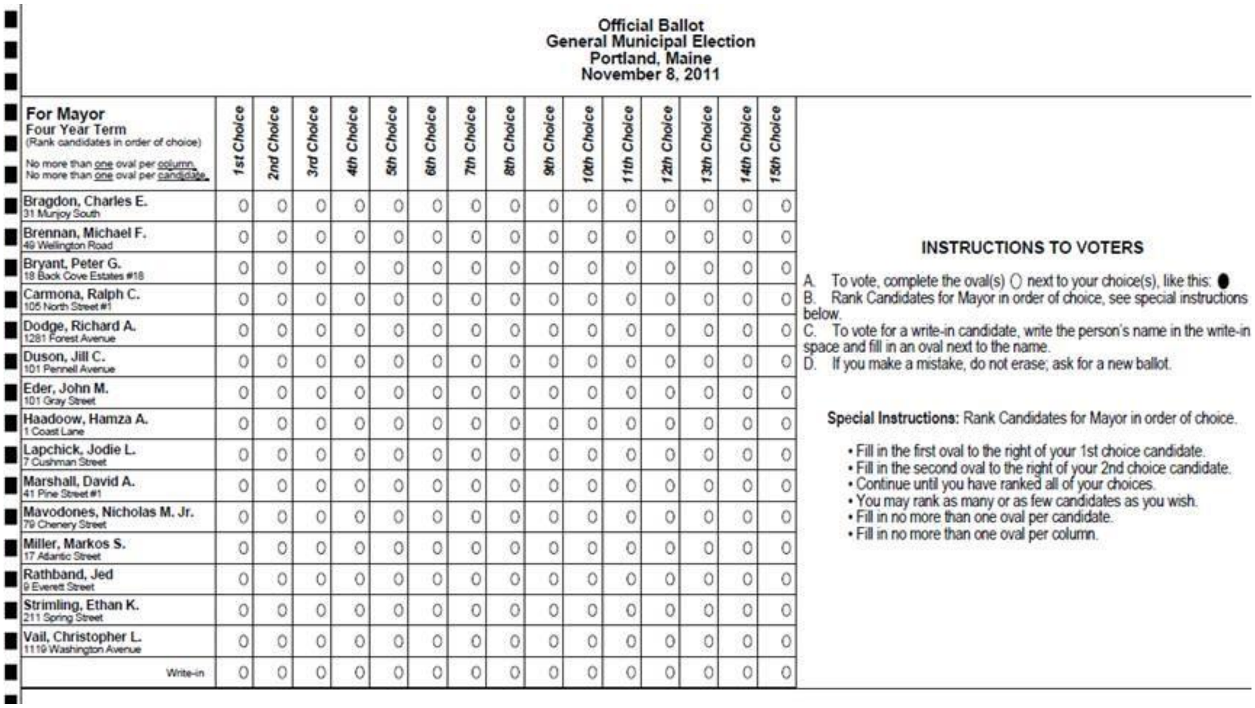
- ID and status information:
  - ID of corresponding ballot style
  - ID to link the CVR with corresponding paper ballot
  - ID of an associated batch and sequence number within the batch
  - ID of the party if a primary election
  - ID of the device that created the CVR and its voting location
  - Sheet number if a multi-page ballot
  - Status of the CVR as needed - counted? adjudicated? needs adjudication?
- ContestVote:
  - Link to a contest containing ballot selection(s) and/or undervoted
  - Status of the contest as needed - overvoted/undervoted? adjudicated? invalidated?
  - Vote variation, e.g., approval, n-of-m, RCV, range, etc.
  - BallotSelectionVote:
    - Link to a marked ballot selection
    - Status of the ballot selection as needed - marked by voter? by rules? by adjudication?
    - If write-in, link to the image of the write-in
    - Additional information as required depending on vote variation.
  - Additional BallotSelectionVotes as needed per vote variation.

CVR example structure

There may be additional ballot selections in the contest depending on the vote variation in effect, e.g., for RCV, multiple candidate ballot selections can be chosen, each one with an associated rank. Ballot selections are relatively simple in structure, but bear more explanation due to the requirement of handling multiple vote variations. Ballot selections are expected to be handled as follows:

- Contests that contain ballot selections and/or that are undervoted are included in the CVR.
  - If the contest is entirely undervoted, only the number of votes lost due to undervoting is included; no ballot selections are included.
- Ballot selections are included for the contest if:
  - The voter made a mark recognized by the scanner or DRE as a valid mark.
  - The voter made a mark recognized by the scanner as a marginal mark.
  - The scanner invalidated a voter mark or generated a new mark because of contest rules.
  - Adjudication invalidated a mark or generated a new mark.
- For each ballot selection, there are one or more associated vote marks and additional values depending on the vote variation and how the ballot is laid out, e.g.,
  - For Plurality voting, generally one mark.
  - For RCV or approval or cumulative or range, possibly multiple marks.
  - For RCV, a ranking may be included.
  - For cumulative or range, a total number of votes (or score) may be included.
  - Each marginal mark detected would be included as a vote mark, flagged as marginal.
  - Each mark must be flagged as to whether it is marginal, a valid voter mark, generated or invalidated due to post-processing or adjudication, etc.

A ballot selection is analogous to a line or row on a ballot, whereas vote marks are analogous to bubbles associated with the ballot selection, as shown in Figure 1, showing an example of a RCV contest.



Official Ballot
General Municipal Election
Portland, Maine
November 8, 2011

For Mayor Four Year Term (Rank candidates in order of choice) No more than one oval per column. No more than one oval per candidate.	1st Choice	2nd Choice	3rd Choice	4th Choice	5th Choice	6th Choice	7th Choice	8th Choice	9th Choice	10th Choice	11th Choice	12th Choice	13th Choice	14th Choice	15th Choice
Bragdon, Charles E. 31 Munjoy South	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Brennan, Michael F. 49 Wellington Road	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Bryant, Peter G. 18 Back Cove Estates #18	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Carmona, Ralph C. 105 North Street #1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Dodge, Richard A. 1281 Forest Avenue	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Duson, Jill C. 101 Pennell Avenue	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Eder, John M. 101 Gray Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Haadoow, Hamza A. 1 Coast Lane	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Lapchick, Jodie L. 7 Cushman Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Marshall, David A. 41 Pine Street #1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Mavodones, Nicholas M. Jr. 79 Chenery Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Miller, Markos S. 17 Atlantic Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Rathband, Jed 9 Everett Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Strimling, Ethan K. 211 Spring Street	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Vail, Christopher L. 1119 Washington Avenue	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Write-in	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

INSTRUCTIONS TO VOTERS

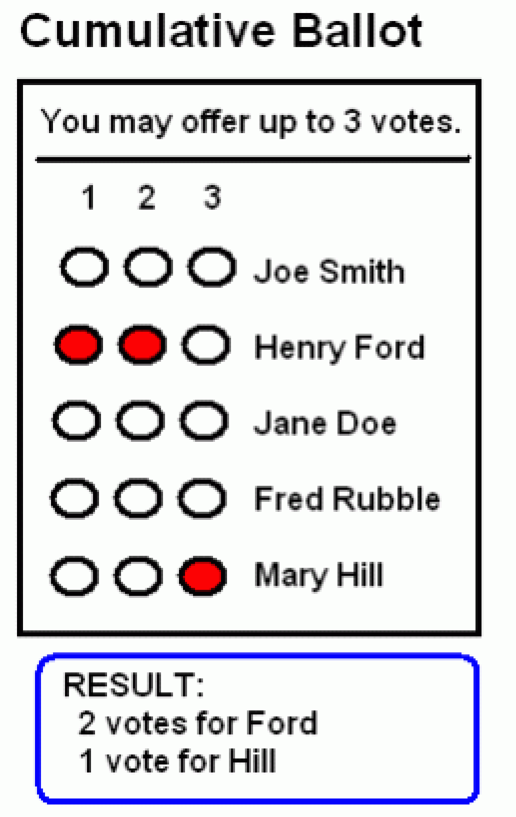
A. To vote, complete the oval(s) ○ next to your choice(s), like this: ●
B. Rank Candidates for Mayor in order of choice, see special instructions below.
C. To vote for a write-in candidate, write the person's name in the write-in space and fill in an oval next to the name.
D. If you make a mistake, do not erase; ask for a new ballot.

Special Instructions: Rank Candidates for Mayor in order of choice.

- Fill in the first oval to the right of your 1st choice candidate.
- Fill in the second oval to the right of your 2nd choice candidate.
- Continue until you have ranked all of your choices.
- You may rank as many or as few candidates as you wish.
- Fill in no more than one oval per candidate.
- Fill in no more than one oval per column.

Figure 1 - RCV ballot example

In Figure 1, depending on contest rules, more than one bubble filled in for a given candidate ballot selection may be allowable or may be an overvote. In Figure 2, showing an example for cumulative voting, multiple bubbles can be filled in vote each ballot selection, with a total number of votes (or score) included for each candidate.



Cumulative Ballot

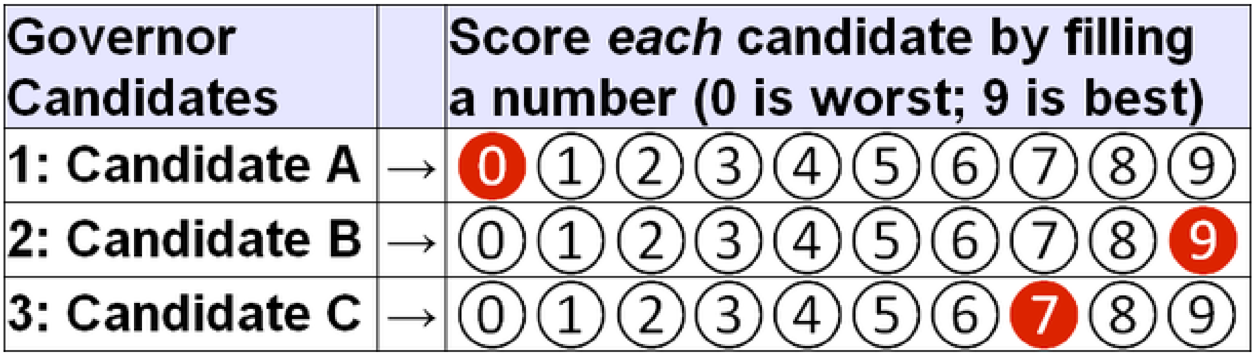
You may offer up to 3 votes.

1	2	3	
○	○	○	Joe Smith
●	●	○	Henry Ford
○	○	○	Jane Doe
○	○	○	Fred Rubble
○	○	●	Mary Hill

RESULT:
2 votes for Ford
1 vote for Hill

Figure 2 - Cumulative voting ballot example

In Figure 3, showing an example of range voting, one bubble can be filled in for each ballot selection, and each candidate will have an associated score or number of votes.



Governor Candidates		Score *each* candidate by filling a number (0 is worst; 9 is best)
1: Candidate A	→	[✓ 0] 1 2 3 4 5 6 7 8 9
2: Candidate B	→	0 1 2 3 4 5 6 7 8 [✓ 9]
3: Candidate C	→	0 1 2 3 4 5 6 [✓ 7] 8 9

Figure 3 - Range voting ballot example

The BallotSelectionVote class includes, for each ballot selection, ranks and scores for candidates as needed in the vote variation, as well as the position of each selection and each vote mark (bubble) associated with the ballot selection. An example of the structure is as follows:

BallotSelectionVote:

- Position on the ballot of the ballot selection.
- Status of the ballot selection, e.g., overvoted.
- Total number of votes, i.e., number of voter marks counted as votes.
- VoteMark – a mark made for the ballot selection:
    - Number of countable votes represented by the mark, e.g., 0 if an overvote or marginal, 1 if a valid voter mark and plurality voting, more for cumulative or range or other vote variations depending on how the ballot is laid out.
    - Position of the mark when multiple marks are possible.
    - Rank, for RCV
    - Status of the mark, e.g., adjudicated, valid voter mark, marginal, generated by scanner, etc.
    - Mark metric – vendor specific measurement of mark quality, density, and so forth
- Additional VoteMarks – if other marks are made for the ballot selection.

BallotSelectionVote example structure

The remainder of this document describes the classes and enumerations in the UML model.

2. Classes

Note: classes Candidate, Code, Contest, GpUnit, InternationalizedText, Party, and any associated subclasses, as well as enumerations IdentifierType and VoteVariation, are defined in the NIST 1500-100 Election Results Reporting specification.

CastVoteRecordReport

This is the root class; it contains information that applies to the entire report such as date/time generated and digital signature.

Attributes:

1. CastVoteRecord – zero or more – a collection of CVRs.
2. Election – zero or more – identifies an election for which CVRs are being reported.
3. GeneratedDate – one – date and time when the report was generated.
4. GpUnit – zero or more – the location, e.g., precinct, for which the report was generated.
5. Notes – zero or one – additional explanatory notes as applicable.
6. ReportGeneratingDevice – one or more – identifies the device generating the report.
7. ReportType – zero or more – identifies the type of report, from enumeration ReportType.
8. OtherReportType – zero or one – used to indicate the type of report when ReportType is *other*.
9. Signature – zero or one – for a digital signature applied to the report.
10. Version – one – the version of the CVR format, e.g., 1.0.

Election

Parent is CastVoteRecordReport. For a given election, defines the contests, ballot selections, candidates, and ballot measures referenced in the CVRs.

Attributes:

1. Candidate – zero or more – identifies the candidates referenced in the CVRs.
2. Code – zero or more – a code associated with the election, e.g., a precinct identifier if the election scope is a precinct.
3. Contest – zero or more – identifies the contests referenced in the CVRs.
4. ElectionScope – zero or one – identifies the election scope, i.e., the political geography corresponding to the election.
5. Name – zero or one – a text string identifying the election.

GpUnit, ElectionScope, SmallestVotingUnit

For GpUnit, parent is CastVoteRecordReport, for identifying the geographical location for the report generation device, which could be the location of a scanner or the location of an EMS that generates an aggregated collection of CVRs. For ElectionScope, parent is Election, for identifying the geographical scope of the election, e.g., a precinct, vote center, county, etc. For SmallestVotingUnit, parent is CastVoteRecord, for identifying the smallest voting unit associated with the CVR, such as a precinct or split. ElectionScope and SmallestVotingUnit are roles (other names) for GpUnit (described in NIST 1500-100).

ReportGeneratingDevice, OriginatingDevice

For ReportGeneratingDevice, parent is Election, for identifying the device that generated the report, e.g., an EMS. For OriginatingDevice, parent is CastVoteRecord, for identifying the device that created the CVR, e.g., a scanner. ReportGeneratingDevice and OriginatingDevice are roles (other names) for the ReportingDevice class, which is a subclass of GpUnit and has additional attributes (described in NIST 1500-100).

CastVoteRecord

Parent is CastVoteRecordReport. A CVR generated by a scanner or other vote capture device, with capability to contain all contests in which a ballot selection was made.

Attributes:

1. Annotation – zero or more – an annotation that applies to the entire CVR.
2. BallotIdentifier – zero or one – an ID for the ballot, will be the same for every sheet of a multi-sheet ballot.
3. BallotImage – zero or more – a pointer to an image file of the scanned paper ballot plus a hash or signature associated with the image file.
4. BallotStatus – zero or more – the status of the CVR, e.g., *adjudicated*, *counted*, etc.
5. OtherBallotStatus – zero or one – used to indicate the status when BallotStatus is *other*.
6. BallotStyleId – zero or one – an ID of the ballot style of the ballot used by the voter for this CVR.
7. BatchId – zero or one – an ID if the CVR is part of a batch.
8. BatchSequenceNumber – zero or one – a sequence number for the corresponding paper ballot.
9. ContestVote – zero or more – used to associate a contest in which a ballot selection was made.
10. Election – zero or more – used to associate the corresponding election.
11. Identifier – zero or one – used to link the CVR with the corresponding paper ballot, e.g., would contain the same value that the scanner impressed on the paper ballot as it was scanned.
12. NumWriteIns – zero or one – the number of write-ins in the CVR.
13. OriginatingDevice – zero or one – the device that created the CVR.
14. Party – zero or more – used to identify a party as needed, e.g., for partisan primaries.
15. SequenceNumber – zero or one – a number to identify the CVR within the export/import file.
16. SheetNumber – zero or one – number of the ballot sheet in the case of a multi-sheet ballot.
17. SmallestVotingUnit – zero or one – the precinct, split, or other location associated specifically with the CVR.

ContestVote

Parent is CastVoteRecord. Identifies a contest in which a ballot selection was made and associates the ballot selection(s). Overvotes plus Undervotes plus TotalVotes must equal the number of votes allowable in the contest, e.g., in a “chose 3 of 5” contest in which the voter chooses only 2, then Overvotes = 0, Undervotes = 1, and TotalVotes = 2, which adds up to the number of votes allowable = 3.

Attributes:

1. Annotation – zero or more – used for an annotation that applies to the contest.
2. BallotSelectionVote – zero or more – used when a ballot selection was made.

3. Contest – one – identifies the contest.
4. Overvotes – zero or one – the number of votes lost due to overvoting.
5. Status – zero or more – the status of the contest, e.g., *overvoted*, *undervoted*, etc., from enumeration ContestStatus.
6. OtherStatus – zero or one – used to indicate the status when Status is *other*.
7. TotalVotes – zero or one – the number of votes cast in the contest by the voter.
8. Undervotes – zero or one – the number of votes lost due to undervoting.
9. WriteInBallotSelectionVote – zero or more – used when the ballot selection is for a write-in.

BallotSelectionVote

Parent is ContestVote. Used for a ballot selection that has been:

a) marked by the voter, including a write-in, or
b) generated by the scanner per contest rules, or
c) marked through adjudication, or
d) determined by the scanner as being a marginal mark.

Attributes:

1. Annotation – zero or more – used for an annotation that applies to the ballot selection.
2. BallotSelection – zero or one – identifies the specific ballot selection in the contest.
3. Status – zero or more – the status of the ballot selection, e.g., *marked-voter*, from enumeration BallotSelectionStatus.
4. OtherStatus – zero or one – used to indicate the status of the ballot selection when Status is *other*.
5. Position – zero or one – position of the ballot selection as it appeared on the ballot.
6. TotalNumberVotes – zero or one – the total number of votes in the ballot selection. Usually 1 for plurality voting, but could be greater for RCV, approval, cumulative, and other vote variations.
7. VoteMark – zero or more – contains further information about the mark made (e.g., by the voter) for the ballot selection.

WriteInBallotSelectionVote

Parent is ContestVote, used when the ballot selection is for a write-in. WriteInBallotSelectionVote is a subclass of BallotSelectionVote and has additional attributes for the image and text of the write-in.

Attributes:

1. Image – zero or one – used for the image of the write-in from paper ballot in base64 binary encoding and/or a file identifier for the file containing the image of the write-in.
2. Text – zero or one – used for the text of the write-in. If the CVR is produced from a handwritten write-in on a paper ballot, the text would be added via adjudication.

VoteMark

Parent is BallotSelectionVote. Contains additional information about a (voter's) mark in a ballot selection. The number of potential VoteMarks that should be associated with a BallotSelection is (for paper ballots) the same as the number of ovals next to a particular choice. There will be usually 1 instance of VoteMark for plurality voting, but there could be multiple instances for RCV, approval, cumulative, or other vote variations in which a voter can select multiple candidates.

Attributes:

1. Annotation – zero or more – used for an annotation that applies to the ballot selection.
2. MarkMetric – zero or more – used if the scanner can associate a mark on a paper ballot with a density or quality or another metric.
3. NumberVotes – zero or one – the number of votes allocated by the voter for vote variations such as cumulative or range voting.
4. Position – zero or one – position of the ballot selection for ballot selections with multiple ovals where multiple selections can be made, e.g., picked from a horizontal or vertical row of candidates.
5. Rank – zero or one – the rank chosen by the voter for RCV.
6. Status – zero or more – further status of the marked ballot selection, e.g., *marked-voter*, from enumeration VoteMarkStatus.
7. OtherStatus – zero or one – used to indicate the status of the ballot selection when Status is *other*.

MarkMetric

Parent is VoteMark. Used for paper ballots containing voter marks when the scanner can assign some metric to the mark such as for mark density or quality.

Attributes:

1. Type – one – the type of metric, will be specific to a scanner manufacturer.
2. Value – one – the value of the metric.

Annotation

Parent is CastVoteRecord, ContestVote, BallotSelectionVote, or VoteMark. Used to record an annotation made by one or more individuals, e.g., adjudicating aspects of the ballot.

Attributes:

1. Message – zero or more – any message created by the annotator(s).
2. Name – zero or more – name of the annotator(s).
3. PreviousStatus – zero or one – if the annotation changed the status, this attribute can be used to preserve that status. The status value can be any value from the BallotSelectionStatus, BallotStatus, ContestStatus, or VoteMarkStatus enumerations.
4. TimeStamp – zero or one – the date and time of the annotation.

3. Enumerations

Enumeration BallotSelectionStatus

Used in ContestVote.Status to identify the status of a ballot selection.

Values:

1. adjudicated – the ballot selection was adjudicated.
2. generated-rules – the ballot selection was generated per contest rules.
3. invalidated-rules – the ballot selection was invalidated by the generating device because of contest rules.
4. needs-adjudication – the ballot selection was flagged by the generating device for adjudication.
5. overvoted – the ballot selection was overvoted.
6. undervoted – the ballot selection was undervoted.
7. other – used when no other value in this enumeration applies to the ballot selection.

Enumeration BallotStatus

Used in CastVoteRecord.Status to identify the status of the ballot.

Values:

1. adjudicated – the ballot has been adjudicated.
2. counted – the ballot has been counted.
3. invalid – the ballot is invalid (not a ballot).
4. needs-adjudicated – the ballot needs to be adjudicated.
5. unreadable – the ballot is not readable by the scanner.
6. other – used when no other value in this enumeration applies to the ballot.

Enumeration ContestStatus

Used in ContestVote.Status to identify the status of a contest in which ballot selection(s) were made.

Values:

1. adjudicated – the contest has been adjudicated.
2. invalidated-rules – the contest has been invalidated by the generating device because of contest rules.
3. overvoted – the contest was overvoted.
4. undervoted – the contest was undervoted.
5. other – used when no other value in this enumeration applies to the contest.

Enumeration HashType

Used in Hash.type to indicate a type of hash used for an image file.

Values:

1. md6

2. sha-512
3. other

Enumeration ReportingUnitType

Used in Election.ElectionScope to identify the geographical scope of the election.

Values:

1. combined-precinct – for a combined precinct.
2. polling-place – for a polling place.
3. precinct – for a precinct.
4. split-precinct – for a split precinct.
5. vote-center – for a vote center.
6. other – used when no other value in this enumeration applies to the geographical scope.

Enumeration ReportType

Used in CastVoteRecordReport.ReportType to indicate the type of report.

Values:

1. adjudicated – used when the report contains adjudications.
2. aggregated – used when the report is an aggregation of device reports.
3. originating-device-export – used when the report is an export from a device such as a scanner.
4. rcv-round – used when the report is the result of a ranked choice voting round.

Enumeration VoteMarkStatus

Used in VoteMark.Status to identify the status of a vote mark made for a ballot selection.

Values:

1. adjudicated-countable – the vote mark was adjudicated and is countable.
2. adjudicated-invalidated – the vote mark was invalidated because of adjudication and is not countable.
3. generated-countable – the vote mark was generated by the generating device per contest rules and is countable.
4. invalidated-marginal-mark – the vote mark contains a marginal mark by the voter and is not countable.
5. invalidated-rules – the vote mark was invalidated by the generating device because of contest rules and is not countable.
6. voter-countable – the vote mark made by the voter is recognized as valid and is countable.
7. other – used when no other value in this enumeration applies to the vote mark.

4. Definitions

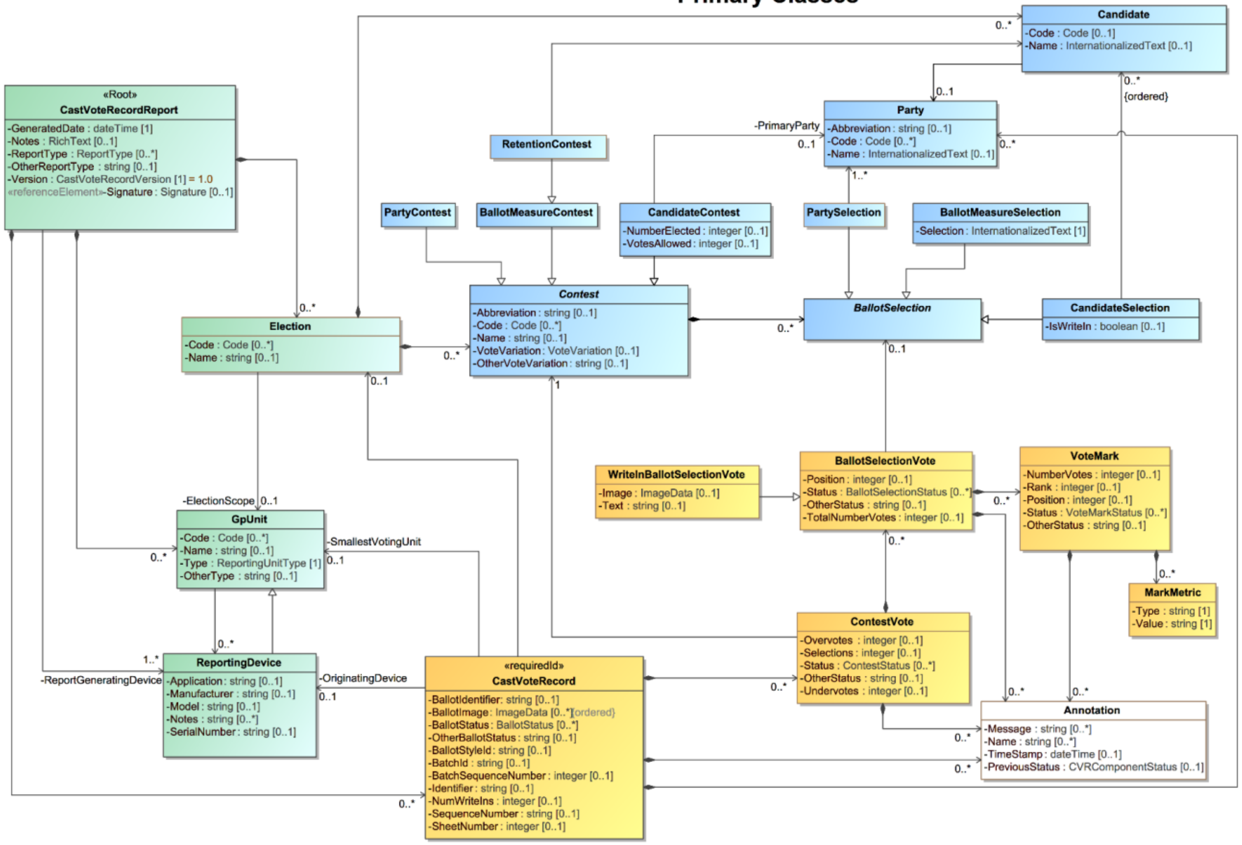
approval voting - a single-winner voting method used for elections. Each voter may "approve" of (i.e., select) any number of candidates. The winner is the most-approved candidate.

cumulative voting - a multiple-winner voting method intended to promote more proportional representation than winner-take-all elections. It permits voters in an election for more than one seat to put more than one vote on a preferred candidate.

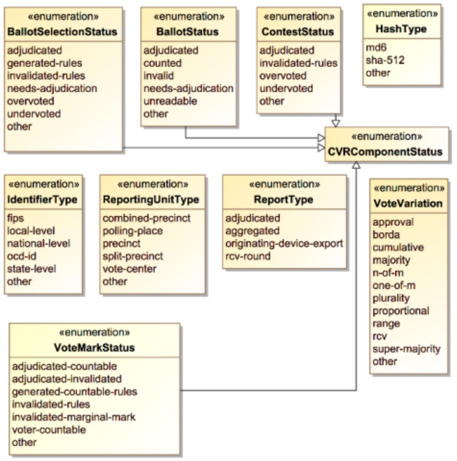
range voting - a voting method for single-seat elections, in which voters give each candidate a score, the scores are added (or averaged), and the candidate with the highest total is elected.

ranked choice voting (RCV) - a set of election 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 elects a winner with a majority of final round votes in a single-winner contest and provides proportional representation in multi-winner contests.

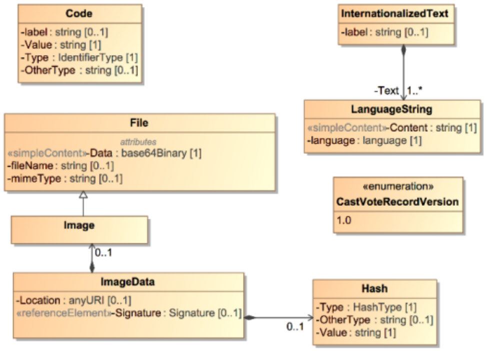
Primary Classes


«Root»
CastVoteRecordReport
-GeneratedDate : dateTime [1]
-Notes : RichText [0..1]
-ReportType : ReportType [0..*]
-OtherReportType : string [0..1]
-Version : CastVoteRecordVersion [1] = 1.0
«referenceElement»-Signature : Signature [0..1]
Election
-Code : Code [0..*]
-Name : string [0..1]
RetentionContest
PartyContest
BallotMeasureContest
CandidateContest
-NumberElected : integer [0..1]
-VotesAllowed : integer [0..1]
Contest
-Abbreviation : string [0..1]
-Code : Code [0..*]
-Name : string [0..1]
-VoteVariation : VoteVariation [0..1]
-OtherVoteVariation : string [0..1]
Candidate
-Code : Code [0..1]
-Name : InternationalizedText [0..1]
Party
-Abbreviation : string [0..1]
-Code : Code [0..*]
-Name : InternationalizedText [0..1]
-PrimaryParty
PartySelection
BallotMeasureSelection
-Selection : InternationalizedText [1]
BallotSelection
CandidateSelection
-IsWriteIn : boolean [0..1]
{ordered}
WriteInBallotSelectionVote
-Image : ImageData [0..1]
-Text : string [0..1]
BallotSelectionVote
-Position : integer [0..1]
-Status : BallotSelectionStatus [0..*]
-OtherStatus : string [0..1]
-TotalNumberVotes : integer [0..1]
VoteMark
-NumberVotes : integer [0..1]
-Rank : integer [0..1]
-Position : integer [0..1]
-Status : VoteMarkStatus [0..*]
-OtherStatus : string [0..1]
MarkMetric
-Type : string [1]
-Value : string [1]
ContestVote
-Overvotes : integer [0..1]
-Selections : integer [0..1]
-Status : ContestStatus [0..*]
-OtherStatus : string [0..1]
-Undervotes : integer [0..1]
Annotation
-Message : string [0..*]
-Name : string [0..*]
-TimeStamp : dateTime [0..1]
-PreviousStatus : CVRComponentStatus [0..1]
-ElectionScope
GpUnit
-Code : Code [0..*]
-Name : string [0..1]
-Type : ReportingUnitType [1]
-OtherType : string [0..1]
-SmallestVotingUnit
ReportingDevice
-Application : string [0..1]
-Manufacturer : string [0..1]
-Model : string [0..1]
-Notes : string [0..*]
-SerialNumber : string [0..1]
-ReportGeneratingDevice
-OriginatingDevice
«requiredId»
CastVoteRecord
-BallotIdentifier : string [0..1]
-BallotImage : ImageData [0..*]{ordered}
-BallotStatus : BallotStatus [0..*]
-OtherBallotStatus : string [0..1]
-BallotStyleId : string [0..1]
-BatchId : string [0..1]
-BatchSequenceNumber : integer [0..1]
-Identifier : string [0..1]
-NumWriteIns : integer [0..1]
-SequenceNumber : string [0..1]
-SheetNumber : integer [0..1]
0..*
0..1
1..*
1

Enumerations



Secondary Classes


Code
-label : string [0..1]
-Value : string [1]
-Type : IdentifierType [1]
-OtherType : string [0..1]
InternationalizedText
-label : string [0..1]
-Text
1..*
LanguageString
«simpleContent»-Content : string [1]
-language : language [1]
File
attributes
«simpleContent»-Data : base64Binary [1]
-fileName : string [0..1]
-mimeType : string [0..1]
«enumeration»
CastVoteRecordVersion
1.0
Image
0..1
ImageData
-Location : anyURI [0..1]
«referenceElement»-Signature : Signature [0..1]
0..1
Hash
-Type : HashType [1]
-OtherType : string [0..1]
-Value : string [1]