NIST Special Publication 1500
NIST SP 1500-20

# Ballot Definition Common Data Format Specification

*Version 1.0*

John Dziurłaj
Benjamin Long

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1500-20

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

## NIST Special Publication 1500
## NIST SP 1500-20

# Ballot Definition Common Data Format Specification

*Version 1.0*

John Dziurłaj
*The Turnout*

Benjamin Long
*Software and Systems Division*
*Information Technology Laboratory, NIST*

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1500-20

January 2023

U.S. Department of Commerce
*Gina M. Raimondo, Secretary*

National Institute of Standards and Technology
*Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology*

NIST SP 1500-20
January 2023

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 the National Institute of Standards and Technology, 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.

All NIST publications are available at http://www.nist.gov/publication-portal.cfm.

Publications in the SP1500 subseries are 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. The opinions, recommendations, findings, and conclusions in this publication do not necessarily reflect the views or policies of NIST or the United States Government.

## NIST Technical Series Policies

## Publication History

Approved by the NIST Editorial Review Board on 2023-01-24

## How to Cite this NIST Technical Series Publication

John Dziurłaj (2023) Ballot Definition Common Data Format Specification. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication 1500 (SP) NIST SP 1500-20. https://doi.org/10.6028/NIST.SP.1500-20

## NIST Author ORCID iDs

Benjamin Long: 0000-0002-4340-7674

## Contact Information

National Institute of Standards and Technology
Attn: Software and Systems Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8970) Gaithersburg, MD 20899-8930
Email: voting@nist.gov

## Abstract

This publication describes a ballot definition common data format for the interchange of logical and physical ballot style information. It contains a UML (Unified Modeling Language) model of the election data and a JSON (JavaScript Object Notation) and XML (eXtensible Markup Language) format derived from the UML model. It contains background information regarding how geopolitical geography is structured and used in the model and schemas. It also provides usage examples for anticipated use-cases. The format is comprehensive and at the same time very flexible, able to accommodate election scenarios used throughout the U.S. It is part of a series of common data format specifications for voting equipment.

## Keywords

Common data format (CDF); ballot definition (BD); ballot style; JavaScript Object Notation (JSON); unified markup language (UML); eXtensible Markup Language (XML)

## Table of Contents

## List of Tables

## List of Figures

## Acknowledgments

The authors wish to thank their colleagues of the National Institute of Standards and Technology Common Data Format (CDF) Ballot Styles Subgroup, who reviewed drafts of this document and contributed to its technical content.

The authors also gratefully acknowledge and appreciate the significant contributions from individuals and organizations in the public and private sectors, whose thoughtful and constructive comments improved the overall quality, thoroughness, and usefulness of this publication.

## Executive Summary

This publication presents a common data format (CDF) for interoperable ballot styles. Interoperable ballot styles can be defined once then shared between ballot-producing systems, scanners and other components. The format, known as the Ballot Definition Common Data Format Specification Version 1.0, is detailed in its coverage of ballot styles including providing structures for contests, candidates, parties and other structural details to how to locate contest option selections and other encodings on a ballot. This publication describes:

- a UML (Unified Modeling Language) model [1],
- derived XML (eXtensible Markup Language), JSON (JavaScript Object Notation) schemas, and microCDF profile [2][3][5]
- usage information and guidance, and
- background information.

The Election Results Reporting Common Data Format (NIST SP 1500-100r2) served as the basis of this specification and was extended to support interoperable ballot styles. Thus, this specification may also be used to convey pre-election or election definition information and may be preferred over NIST SP 1500-100 for systems that may wish to produce detailed ballot styles.

The format accommodates the following use-cases:

**Logical ballot definition** – for reporting of election data about offices, political parties, and the geopolitical geographies associated with the reporting jurisdiction, as well as any election-specific information such as the contests, candidates, or ballot measures, and their appearance on ballot styles; and

**Physical ballot definition** – a superset of logical ballot definition including details about ballot styles and how selections may be captured. Multiple ballot formats can be defined for the same ballot style, such that different ballot producing systems can coexist within the same ballot definition.

Physical ballot definition requires the use of the microCDF format. The microCDF enables interoperability between software independent artifacts such as paper ballots and vote records by defining a terse messaging format that can be encoded directly onto them. Ballot definition uses microCDF for ballot style identification and optionally, contest selection capture.

The specification is geared towards the following audiences:

- Election officials
- Voting equipment manufacturers
- Election-affiliated organizations and
- Election analysts and the general public

The XML and JSON schemas and microCDF profile associated with this specification are derived from the UML model, which defines the types, structure, and interrelationships of geopolitical geography across the United States. The model was designed to accommodate multiple types of contests and their many variations.

# 1. Introduction

The Ballot Definition (BD) is a UML-based (Unified Modeling Language)[1] Common Data Format (CDF) for exchanging ballot data between components of systems used for managing elections across states and territories of the United States. The format enables the interchange of election information between election management systems and serves as a means for combing election data from different election management systems (EMSs) or transferring data between components that require such information. Finally, the format supports defining details about ballot styles used in one or more elections so that ballots produced by one device may be read by another.

This specification includes the data model in UML that specifies and defines the data involved in pre-election setup including ballot styles. It also includes discussion and documentation for XML (eXtensible Markup Language)[2] and JSON (JavaScript Object Notation)[3] schemas that were derived from the UML model.

The primary features of this specification are:

- major classes of data and their attributes and associations are fully defined in a UML data model;
- the data model can be used to generate data formats (for example XML, JSON);
- ballot definition can be described in logical or physical terms based on use-case;
- election data can be reported at flexible levels from basic to very detailed;
- geopolitical units of geography can be specified in a flexible manner to mirror structures used across states, counties, and cities; and
- major elements such as contests, geopolitical units, and parties include the capability to support multiple types of identifiers and cross-references.

All terms used in this document are consistent with terms defined in VVSG 2.0 [10].

## 1.1. Why this specification is needed

The purpose of this specification is to provide a comprehensive, flexible, and interoperable ballot definition in XML and JSON format for manufacturers to integrate into their voting equipment and for election offices and other groups to use in their own software. Some advantages of using this specification are that:

- Ballot definitions can be shared and reused across components from various vendors, thus enabling interoperability between equipment such as ballot marking devices (BMDs) and scanners;
- the need for custom software and custom ballot definition formats is greatly reduced; and
- jurisdictions that use multiple versions of EMSs and tabulators can more easily transfer information between systems.

## 1.2. Intended Audience

The intended audience of this specification includes election officials, manufacturers and developers, and others in the election community, including the general public. While no specific background is required, this document uses terminology that may be most familiar to election officials and manufacturers.

## 1.3. Motivation and methodology

This specification was motivated by the desire to increase interoperability between components at different levels of government (e.g., EMSs at state and local level), and between components in voting systems (e.g., ballot producing systems and scanners).

In states where counties (or cities, towns, etc.) manage their respective ballot data, there exists a need for the county and the state election offices to exchange ballot data so that each have a complete set of ballot data containing all statewide and local-level contests and candidates. Currently, there exists no common format for exchange of this data, and in some cases the election offices must re-enter the data by hand.

Furthermore, the lack of a common data format between ballot producing systems and scanners has severely restricted how voting systems can be fielded. This is a finding from a NIST Grant/Contractor Report "Recommendations for voting system interoperability"[4] which described gaps in CDF support for describing how contest selections are captured.

To address the foregoing, the Ballot Definition CDF seeks to increase interoperability between EMSs, components that print ballots, components that mark ballots, and components that scan and interpret marks on ballots (e.g., ballot-aware scanners). Specifically, this specification seeks to enable ballots and other paper vote records to be read by scanners through a shared ballot definition.

The use of this specification, in conjunction with the microCDF (mCDF)[5] and mCDF Profile for Ballot Style Identification (described in Appendix A), should make it possible to produce interoperable optical mark recognition (OMR) style ballots. A separate mCDF Profile is available for vote record outputs from ballot marking devices, mCDF Profile for Contest Selection Capture[6], and seeks provide interoperability for such scenarios.

The following high-level use-cases were developed for ballot definition:

- **Logical ballot definition** – export from an EMS of the contests, candidates, ballot measures, ballot styles, information about offices, geopolitical geographies, and other data associated with the election jurisdiction.
- **Physical ballot definition** – export from an EMS of the physical structures required to be known by scanners in order to properly interpret a paper vote record, such as method of ballot style identification, location of contest option positions, and style designation.

Because the Election Results Reporting (ERR) CDF[7] contained many of the classes and relationships required to support logical ballot definition, the pre-election use-case subset of the ERR UML data model serves as the starting basis for the BD CDF.

The UML data model was subsequently expanded to define the data associated with ballot definition specific use cases and to show the relationship and organization of the data elements. Finally, XML and JSON schemas were generated from the UML data model. The schemas define the rules of the respective format.

The advantages of using a UML data model as an intermediate step to generating the XML and JSON schemas include that:

- the model is independent of the concrete format or other potential formats that could be derived, and relationships between data elements are easier to correctly define and visualize when they are independent of any specific data format.
- If changes are needed to the format, one can make changes to the UML model and generate a new version of the format.

Note that this specification addresses U.S. governmental elections and is not intended for use “as is” in other types of elections or in other countries. However, the specification was written with the intention that it be adaptable to other election environments.

This document represents novel work in interoperability in the voting space. It is anticipated that revisions to this specification will be made as vendors and others attempt to implement it.

### 1.4. Document Structure

**Section 2** starts with an overview of geopolitical geographies such as counties, districts, and precincts, describing how they are categorized, how they interrelate, and how election results are tied to them.

**Section 3** contains an overview of the two primary use cases for ballot definition and the UML data model that implements them.

**Section 4** contains overview documentation for the UML model and explains how it is mapped to XML and JSON.

**Section 5** contains documentation for the UML model.

**Section 6** provides examples on the usage of CDF.

**Section 7** gives suggestions for effective scanning and processes of interoperable ballot styles.

The appendices include profiles, definitions, acronyms, and instructions for downloading the files associated with this specification.

# 2. Background: Geopolitical Geography

This section provides an overview of the geopolitical geography in the United States as it relates to elections and election results reporting and provides background for how geopolitical geography is implemented in the UML model and JSON /XML schemas that are described in the remaining sections. Knowing what constitutes geopolitical geography and how it is interrelated and used in elections provides the underpinning for understanding the complexities of ballot style construction.

## 2.1. The Primary Types of Geopolitical Geography

The primary types of geopolitical geography include those that run elections such as states, counties, and cities, as well as the many types of election districts that are tied to contests, precincts, and various other geographical units associated with political boundaries. Generally, the media and election analysts wish to obtain voting results comprised of these units; thus, the process of running an election includes associating contests and vote counts with these units for reporting.

Ballot counts and vote counts for contests can be associated with different types of geopolitical geography, ranging from aggregated counts associated with a county or state down to more detailed counts associated with a precinct and breakdowns of a precinct. Precincts are generally the smallest unit of geopolitical geography. Precincts can be thought of as the building blocks that compose all other geopolitical geography, and are generally the entity that results are reported by.

Geopolitical geography can often be quite complex in that some are hierarchical, others overlap, and still others change their boundaries regularly, sometimes several times within a year. Changes to city and district boundaries affect precinct boundaries, splitting them into multiple parts (called split precincts), with each part requiring a distinct ballot style.

The following sections break down geopolitical geography into three primary types and show how the geographies interrelate. These three types are:

1. Governmental geography
2. Political geography
3. Administrative geography

### 2.1.1. Governmental geography

Governmental geography refers to entities that run elections and are well-established and do not change over time, with the exception of some cities. For many states, the governmental geography is hierarchical, as shown in Figure 1. This can be categorized as follows:

- States
- Counties
- Cities
- Towns and Townships
- Other Civil Divisions.

American Indian reservations may span states and are sovereign bodies for the purposes of their internal governmental elections. They are not represented hierarchically under the state, however voters residing in reservations vote in the same federal, state, and local elections as other voting residents of the state.

All states have counties, although some use different words to describe them, such as, parishes for Louisiana and boroughs for Alaska. Townships occur in 13 states and adhere to county boundaries. In the six New England states, townships run the election process, and there is no county government; thus, election results are reported directly to the state. Municipalities (cities, towns, or villages) in Michigan, Minnesota, and Wisconsin also run their elections, but report their information to the county, which then reports to the state. In New York City, each of the 5 boroughs run their own election but reports the results to the NYC Board of Election. Other civil divisions include boroughs as used in Connecticut, New Jersey, Pennsylvania, and other states.

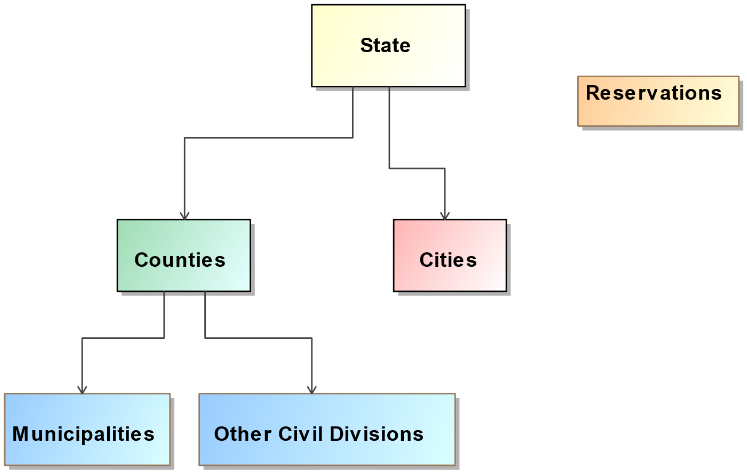


Figure 1 – Governmental geographies

Governmental geographies are associated with offices that are elected jurisdiction-wide (such as for Governor, County Clerk, Supervisor, Treasurer, Assessor, and Highway Commissioner) and thus, do not require different ballot style areas within the geography for those offices, that is, all voters in the jurisdiction vote for the office.

Governmental geographies do not cross the lines of the precincts that compose them. However, cities can change their boundaries through annexations, and, in some states, city boundaries can also cross county boundaries. Thus, changes to city boundaries may result in crossing the boundaries of one or more precincts, creating split precincts and requiring a distinct ballot style for each split precinct.

### 2.1.2. Political geography

Political geographies are those that tend to be population-based and therefore may change with each U.S. Census happening every 10 years in a process known as re-districting. Political geographies are also known as *election districts*, where people are elected to an office that has jurisdiction within a specific geography, for example, a U.S. Congressional district.

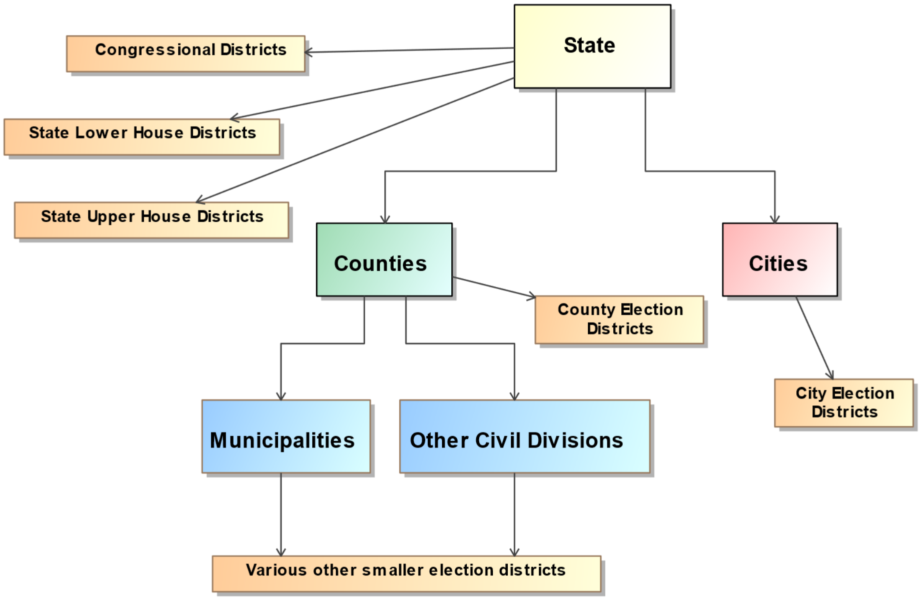

State
Congressional Districts
State Lower House Districts
State Upper House Districts
Counties
Cities
County Election Districts
City Election Districts
Municipalities
Other Civil Divisions
Various other smaller election districts

Figure 2 – Political geographies

Figure 2 shows the most common political geographies as they interrelate with the governmental geographies. Political geographies can be categorized as follows:

- U.S. Congressional districts
- State senate or upper-house districts
- State house or lower-house districts (in some states, several state house districts combine to form a state senate district)
- County election districts
- City election districts
- Numerous other forms of election districts

Because election districts can change as they are re-drawn, political geographies will often divide precincts, creating split precincts and requiring a distinct ballot style for each split precinct. Geographic Information Systems, or GIS, are often used to by election authorities to ease this task.

### 2.1.3. Administrative geography

Administrative geographies are identified this way because their boundaries are determined by election or civil administration. Administrative geographies include precincts and their various types such as wards, combined precincts, and split precincts. They can be very small, sometimes only applying to several streets or houses or even only a single house along a street. They can involve territory that is non-contiguous, for example, for some of the taxing and special districts. They can change many times throughout a given year, even daily in some cases.  Figure 3 shows the basic administrative geographies, which can be categorized as follows:

- Election administrative areas

    - Precincts, split precincts, combined precincts, wards
    - Polling places, vote centers
    - Various other ballot style areas
- Taxing districts, such as fire, water, sewer, transit, school, police, hospital, utilities
- Special districts, that is, unique areas brought together for a referendum.

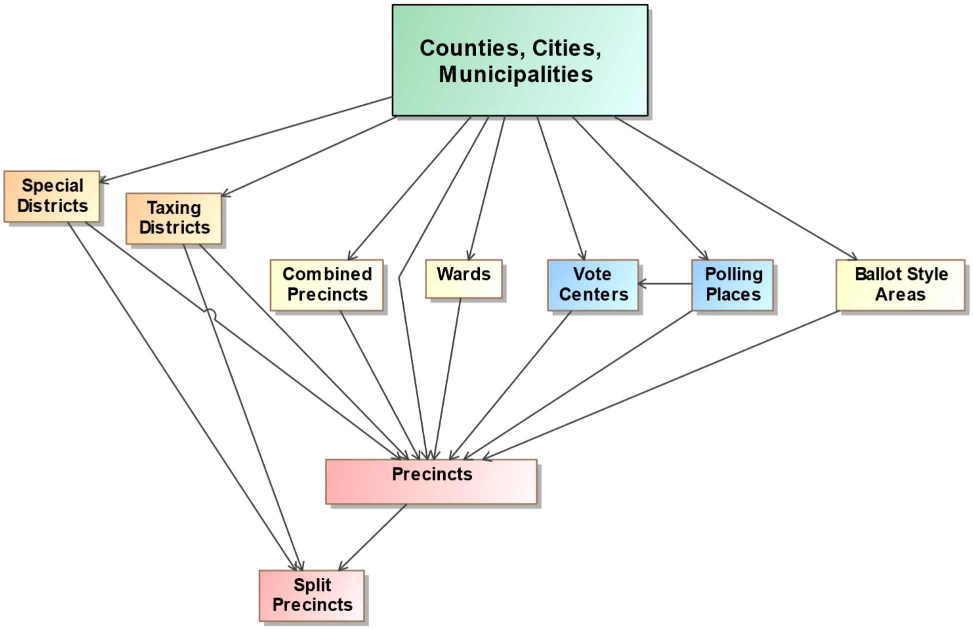


Figure 3 – Administrative geographies

## 2.2. Linking Geopolitical Geographies Together

As an example of administrative geographies and their relationship to political and governmental geographies, Figure 4 shows the wards and precincts that make up the city of Cambridge, MA, and Figure 5 shows how the wards and precincts in the city (where the precinct boundaries are delineated in red) compose the U.S. Congressional election districts [8]. The wards are implemented as collections of precincts and are shown in a distinct color in Figures 4 and 5.

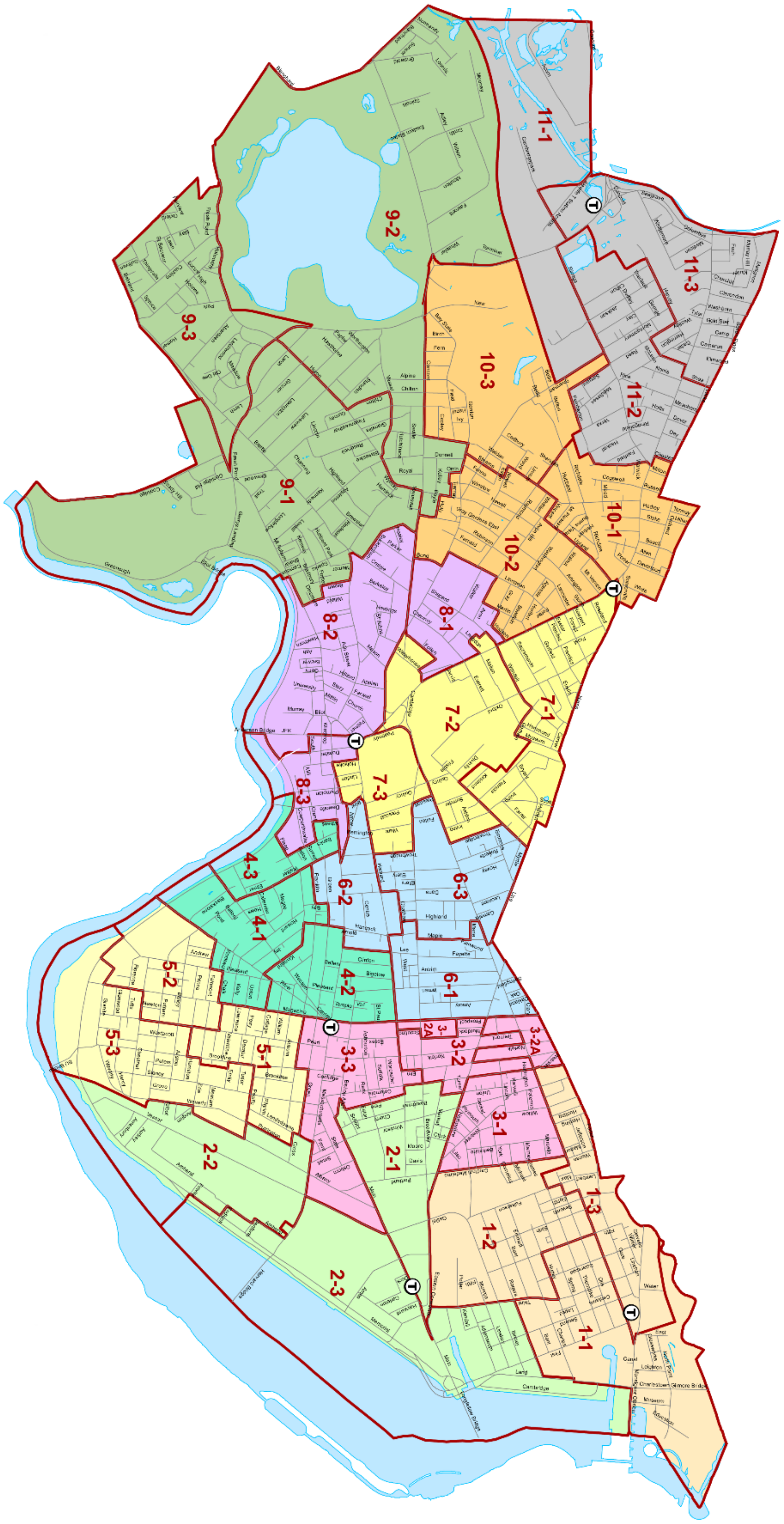


Figure 4 - Ward and Precincts in Cambridge, MA.

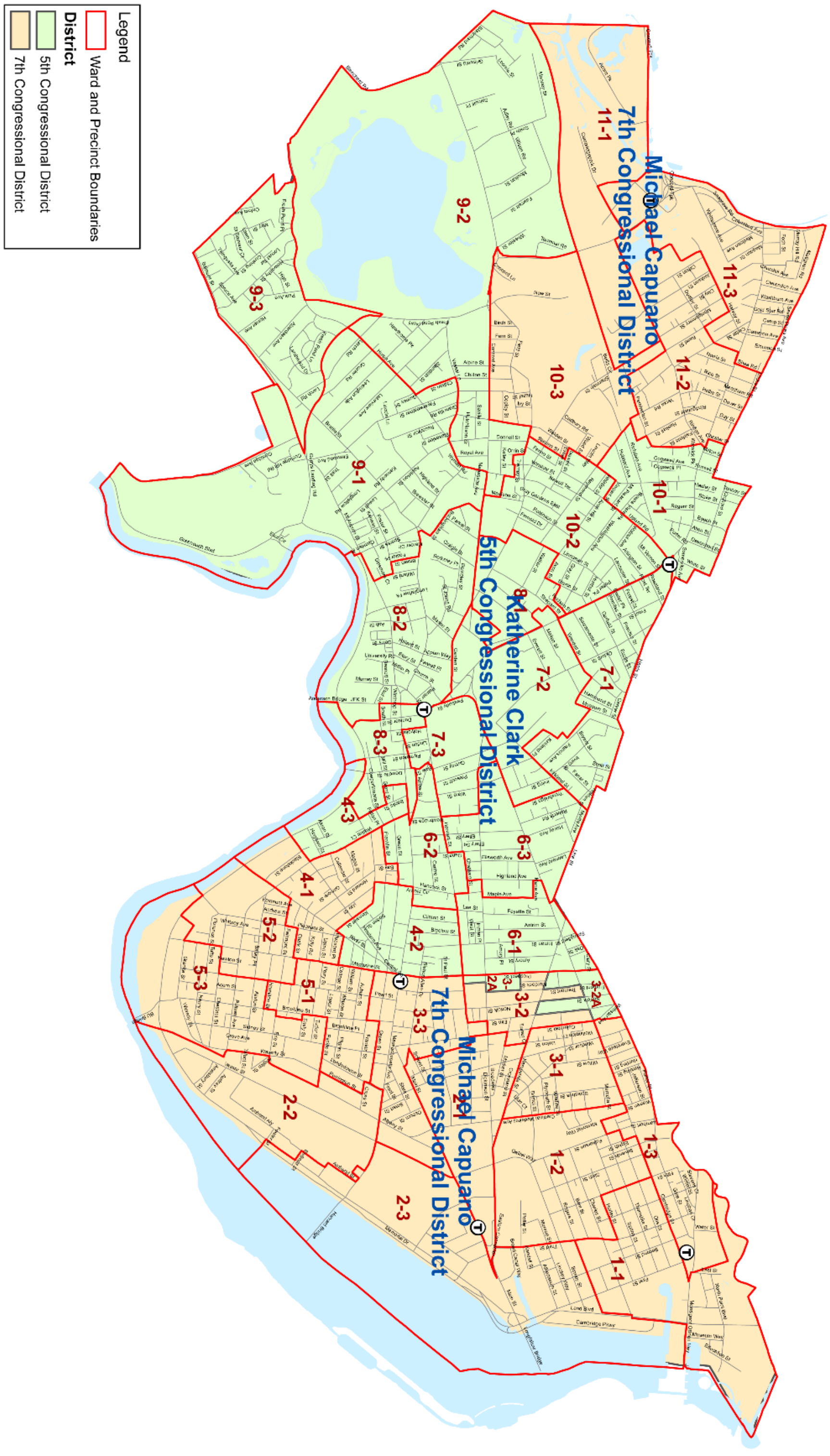


Figure 5 - Districts Overlaying Wards and Precincts in Cambridge, MA.

In many states, the boundaries of election districts may crisscross the precinct boundaries, creating one or more split precincts, with a distinct ballot style for each split precinct. Depending on the number of districts and how often they cross the precinct boundaries, the resulting number of ballot styles created could grow well beyond the number of whole precincts. It is possible that, despite best efforts, very low numbers of voters or even just one voter will require a distinct ballot style.

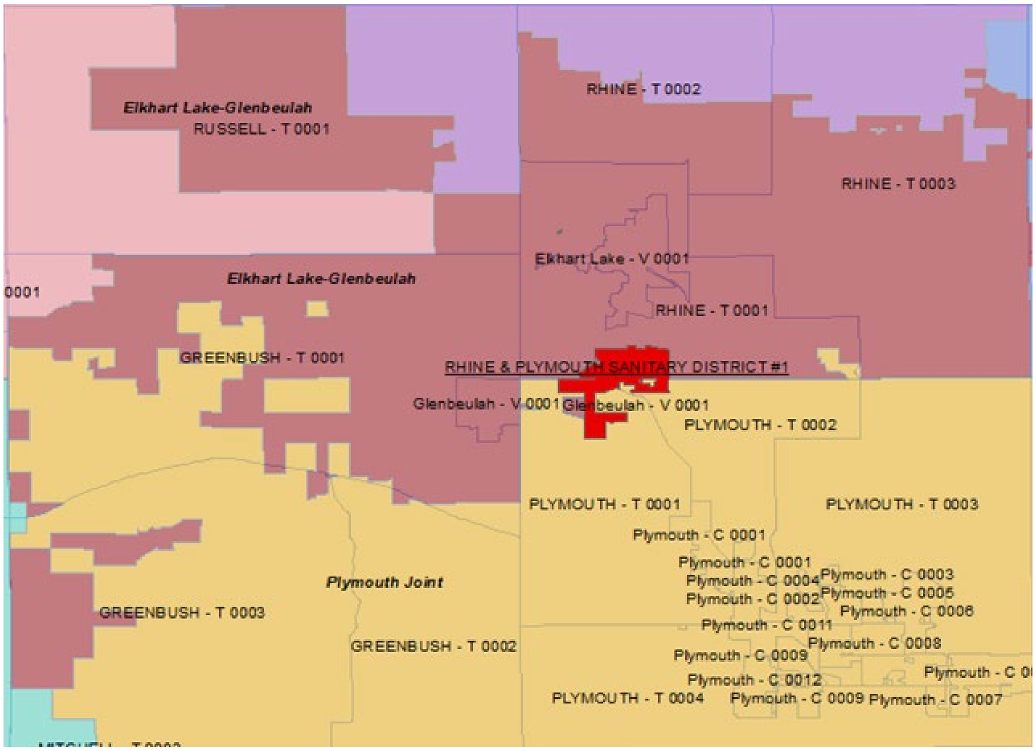


Figure 6 – Overlapping Non-hierarchical Election Districts

Figure 6 shows different overlaps between Precincts, School Districts, and a Sanitary District, all which create precinct splits. The Elkhart Lake-Glenbeulah school district in rust color, and the Plymouth Joint school district in yellow, do not follow the precinct lines or municipal boundaries (the gray lines). The Rhine and Plymouth Sanitary District #1 in red also does not follow the precinct lines or the school district lines. This creates multiple split precincts. For example, Town of Rhine ward 1 (RHINE – T 0001) has part of the Rhine & Plymouth Sanitary District #1 as well as small area of the Plymouth Joint school district. The rest of the precinct has no sanitary district and is in Elkhart Lake-Glenbeulah school district. This creates up to 3 different ballot styles in that one precinct, depending on the contests up for election.

Precincts can be split as well by changes to the other administrative geographies. Adding to the complexity, a number of states now use combined precincts and vote centers on election night, which associate multiple precincts with one polling place. This means that for a vote center handling multiple precincts that themselves may be split, there can be potentially many different ballot styles in use at the vote center, with each voting device needing to display or tabulate any one of the ballots. This adds further complexity and places additional demands on election jurisdictions and their ability to manage and report details of votes on election night and post-election [9].

To make this situation more manageable, some states and counties prefer over time to *heal* split precincts by combining them with other precincts or generally redrawing the precinct boundaries so that the number of ballot styles is reduced, and election management and reporting is less complicated.

## 2.3. Geopolitical Geography in the UML Model and Related Schemas

The previous discussion demonstrated that there are different types of geopolitical geography that overlap each other or behave hierarchically, resulting sometimes in very complex maps and many small geopolitical units that require distinct, different ballot styles. Election officials may spend considerable time managing this complexity.

Furthermore, each state and sometimes county or city will manage elections differently, using combinations of units such as combined precincts or wards, with specific rules about how the associated contests operate. When one combines the complexities of geopolitical geography with the different election rules employed in the U.S. states and territories, one sees that running an election can be an extremely complicated endeavor. Ballot definition mirrors this complexity.

Note that the different geographies form relationships much like a lattice, in which objects can be related in non-hierarchical ways. The UML model and JSON/XML schemas implement geopolitical geography in this way using an object that can be linked with other objects depending on the type of geopolitical geography. In the UML model, this object is referred to as the GpUnit (short for 'Geopolitical Unit") class, and in the XML schema it is called the *<GpUnit>* element. GpUnits can model a district, county, or precinct, and can be linked to each other to mirror the real-world geopolitical geography of the reporting jurisdiction.

GpUnits can be linked hierarchically when modeling jurisdictional geographies. To model a jurisdiction that runs or reports on elections, the lowest-level GpUnits, that is precincts, will be children of the election-running GpUnit, say a city, county, or state.

District GpUnits need to be linked to the precinct or split precinct GpUnits that compose them. The precincts and split precincts thus link the jurisdictional and district GpUnits together, as shown below in Figure 7 (and described in greater detail in section 5). The wards in Figure 7 are synonymous with precincts in Wisconsin and are the children of the combined precincts, and so forth on up to the state. The precincts and split precincts are also the children of the districts that they compose.

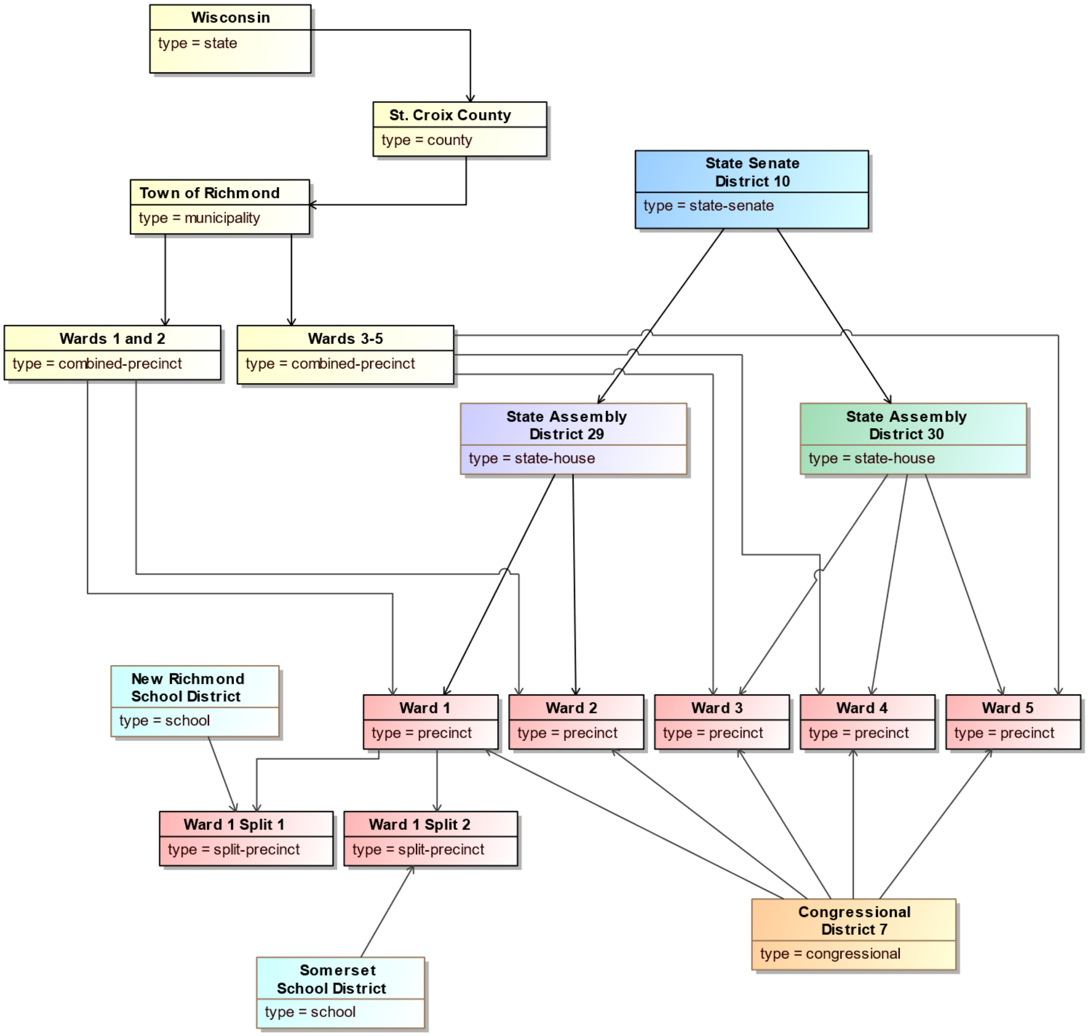


Figure 7 - GpUnit Structural Hierarchies

## 3. Use Cases

This section describes the two use cases for ballot definition:

1. **Logical ballot definition** – for reporting of election data about offices, political parties, and the geopolitical geographies associated with the reporting jurisdiction, as well as any election-specific information such as the contests, candidates, or ballot measures, and their appearance on ballot styles; and
2. **Physical ballot definition** – a superset of logical ballot definition including details about ballot styles and how selections may be captured.

The logical ballot definition use-case is substantially similar to the pre-election use-case of the Election Results Reporting (ERR) CDF; whether to use ballot definition (BD) or ERR to exchange such data comes down to if there is a desire to expand to other ballot definition or results reporting use-cases in the future.

### 3.1. Logical Ballot Definition Use Case

The logical ballot definition use case enables election officials to report on a variety of election-related data. The data could include information about a specific upcoming election, or it could provide more general information about the reporting jurisdiction so as to determine whether the election data is accurate and organized correctly or to convey information to the general public about contests and ballot information in an upcoming election. Logical ballot information may come from any of the databases or devices that an election jurisdiction uses to manage elections, including voter registration databases, ballot programming systems, candidate filing systems, EMSs, campaign finance systems, etc.

The data for the logical ballot definition use case includes the following:

- Pre-election reports in a variety of formats, including:
    - As one complete file or a sequence of files;
    - As additions to previous reports; and
    - As corrections to previous reports;
- Jurisdictions, districts, precincts and voting locations within the scope of the reporting jurisdiction and how they are organized;
- Political party information;
- Offices associated with contests and districts;
- Information about persons relevant to the jurisdiction such as authorities, office holders, etc.;
- Election type, date, place, jurisdiction, authority information, registered voters. The type of election could include:
    - Open and closed partisan primaries;
    - Runoffs;
    - Special elections; and
    - General elections;
- Ballot measures and contests, including controlling and retention contests;
- Candidates on the ballot and their associated party affiliations and contact information;

- Ballot styles containing contest and candidate information in the order as they appear on the ballot at specific precincts or split precincts.

## 3.2. Physical Ballot Definition Use Case

The BD CDF supports expressing the way contest selections are captured on a ballot. It supports the output of two ballot marking approaches:

1. Voters make indications for contest selections by marking a piece of paper. The expected location of a mark is designated as the space within a geometric shape such as a rectangle or ellipse (oval).
2. Voters make indications for contest selections by interacting with a ballot marking device that stores them onto a paper vote record. Contest selections are encoded in a human readable and machine-readable (microCDF) form.

### 3.2.1. Imaging Model

The BD CDF must have a method to represent and locate the content of a ballot. This representation is called an imaging model.

The BD imaging model is influenced by industry standard imaging models such as Adobe PostScript and PDF. The BD imaging model uses a two-dimensional coordinate plane whose origin is located at the top left-hand side of the page.

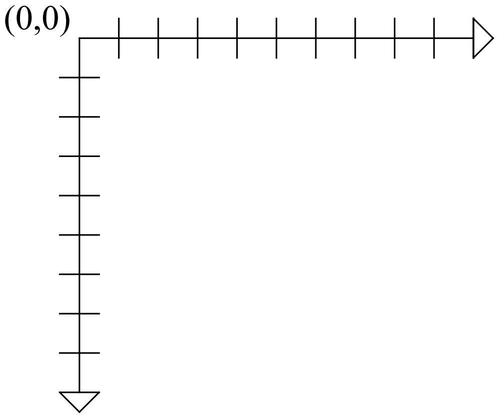


Figure 8 - Origin of coordinate plane at top-left of page

Parts of the ballot, such as the contest option and write-in positions are located using bounding boxes specified in absolute coordinate space. Bounding boxes can contain simple geometry, such as rectangles or ellipses. Note that the two-dimensional coordinate plane in these figures is merely shown for illustrative purposes to demonstrate how such a coordinate system may be used. These illustrations are not intended to imply any particular scale or unit of measure.

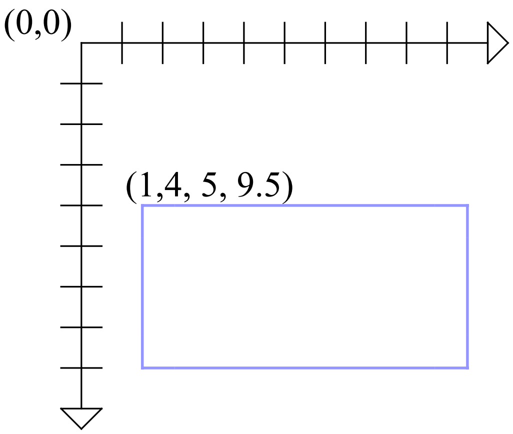


Figure 9 - Bounding box with coordinate system reference

### 3.2.2. Ballot Formats

A ballot format is a specific set of assumptions that a ballot-producing system (BPS) makes regarding how a ballot will be read. Such details do not vary from ballot to ballot, but do vary from vendor to vendor, or by the type of ballot (e.g., fill in the oval vs. scan a barcode, etc.). Each ballot format must have a unique identifier within the ballot definition in the case that multiple ballot-producing systems are in use at the same time. An interoperable ballot will have a corresponding identifier encoded on it for a scanner to be able to identify its ballot format.

Each ballot format must specify the unit of measurement (e.g., millimeter, point) it uses to describe parts of the ballot. All bounding box measurements associated with that ballot format will use that specified unit of measurement.

Finally, every ballot format may define where the encoded ballot identifiers are located on a ballot (microCDF Areas).

### 3.2.3. microCDF Areas

microCDF (mCDF)[5] areas are portions of the ballot that contain data in the mCDF serialization and encoded onto paper according to some symbology. The purpose of these areas is the same as code channels or barcodes appearing on ballots produced by contemporary systems, but in an interoperable form. The BD CDF supports the use of two messages:

- Ballot Style Identification (BSI) – for ballot formats that use optical mark recognition (described in Appendix A)

Contest Selection Capture (CSC) – for ballot formats that encode contest selections inside a mCDF instance (See reference[6]).

The symbology must be machine readable but may be human readable as well. mCDF areas specify the position and dimensions of the bounding box where the encoded mCDF data can be found. The BD CDF specification does not require the use of particular symbologies, but the ballot format should specify the symbology it uses.

### 3.2.4. Support for optical mark recognition ballots

*This section applies to ballots that contain contest option positions in predetermined areas.*

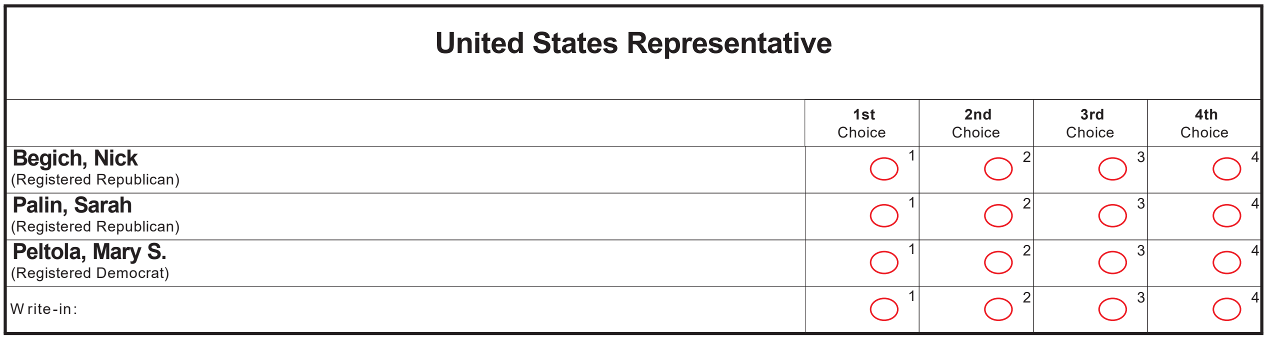
During scanning, mCDF area(s) are located and the BSI message is decoded. The message is used to locate the correct ballot style definition within the scanner’s storage that can be used to locate and describe the meaning of other marks on the ballot.

It can be placed on the front or back of the ballot, or both. There should only be one ballot style locator per side of the ballot sheet.

### 3.2.5. Physical Ballot Styles

Physical details specific to a particular ballot-producing system (BPS) are stored alongside the logical ballot style in “physical” class elements. Each logical ballot style can have zero or more physical contests. Each physical contest represents a particular rendering of the contest according to specific ballot style and ballot format of the BPS.

Each physical contest contains an ordered listing of physical contest options. Each physical contest option has one or more option positions. In N-of-M voting, there will only be a single option position per physical contest option, while other methods such as rank choice voting (RCV) may have many positions per physical contest option. Additionally, separate write-in response areas can be specified.



**United States Representative**

| | **1st** Choice | **2nd** Choice | **3rd** Choice | **4th** Choice |
|---|---|---|---|---|
| **Begich, Nick**<br>(Registered Republican) | ○ 1 | ○ 2 | ○ 3 | ○ 4 |
| **Palin, Sarah**<br>(Registered Republican) | ○ 1 | ○ 2 | ○ 3 | ○ 4 |
| **Peltola, Mary S.**<br>(Registered Democrat) | ○ 1 | ○ 2 | ○ 3 | ○ 4 |
| Write-in: | ○ 1 | ○ 2 | ○ 3 | ○ 4 |

Figure 10 - RCV contest where each contest option has four option positions

### 3.2.6. Support for ballot selection records

This section describes features required to support scanning of vote records that use ballot selection or “summary” cards. Such cards are outputs from a ballot marking device (including remote ballot marking devices) and encode their selections using various symbologies (e.g., QR Code, PDF417, etc.).

During scanning, mCDF area(s) are located and the CSC message is decoded. The message is used to locate the correct ballot style definition within the scanner’s storage. Once the proper ballot definition has been retrieved, contest option selections for a given contest are identified by the presence of zero or more CCS (CVR Contest Selection) segments. The information contained within the CSC combined with that of the ballot definition is enough to produce a cast vote record.

## 4. UML Model Overview

This section presents the UML model that was structured to implement the use cases. It includes a class diagram that shows a picture of the model and an overview of how to read the relationships between the classes so as to understand how the model and related schemas are structured.

The UML model represents a format-independent description of the data required by the two major use cases of the specification. Its primary benefit is that it unambiguously defines and describes the data elements and how they are related without requiring readers to know the technical details of any particular data format implementation, e.g., XML. By using a model-based approach, the resultant data format is more likely to be well-structured and more tolerant to modifications. The data format can be generated from the model using commercial tools, thus if changes need to be made to a format, the model can be changed, and the format can be re-generated.

Figure 11 shows a high-level view of the class diagram, minus its attributes and certain ancillary classes (the complete diagram is available for download, see Appendix D).

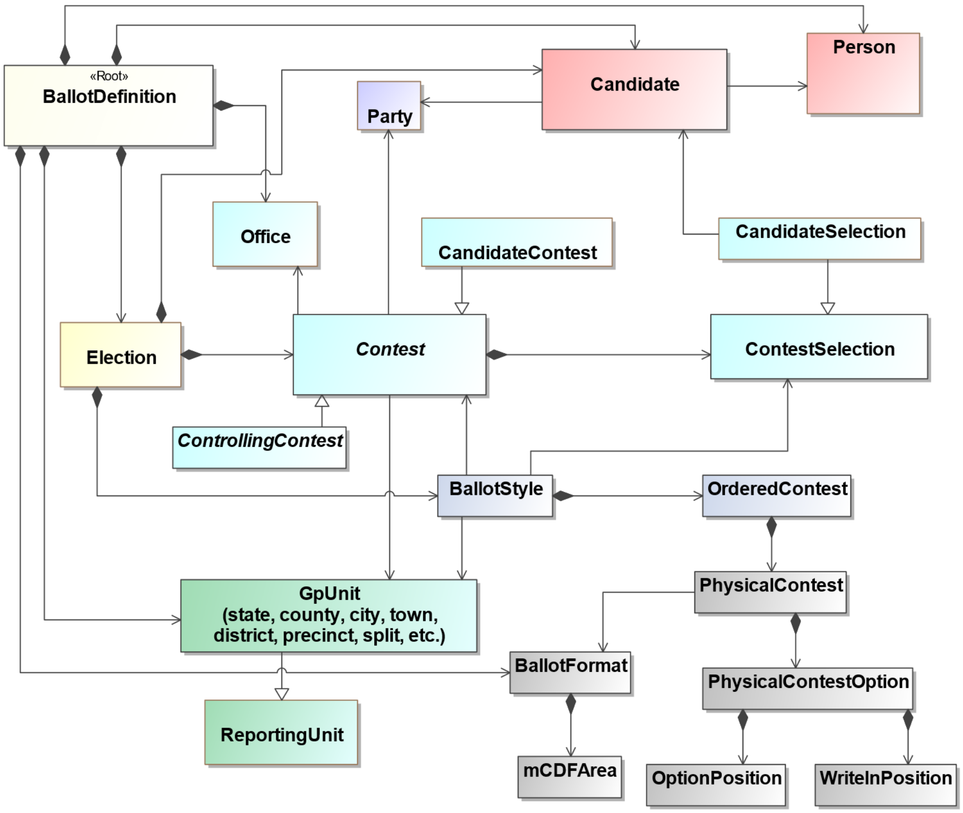


Figure 11 – Simplified UML Class Diagram

## 4.1. Major Classes

Each class represents a major data element, e.g., Contest, Candidate, or Party, and classes that are highly related to each other are shown in the same color. The major classes shown in Figure 11 are:

- **BallotDefinition** – the root class, includes attributes describing the issuer of the report, when generated, etc.;
- **Election** – for a specific election, includes attributes describing the type of election, date, etc.; it also links to ballot counts for the election;
- **GpUnit** – short for "Geopolitical Unit," for describing units of geopolitical geography so that they can be associated with contests, offices, and ballot styles. There is one subtype of GpUnit: ReportingUnit;
- **ReportingUnit** – for jurisdictions, districts, precincts, and other units for which election results can be associated; and
- **Office** – for political offices that are associated with contests and election districts and current office holders;
- **Party** – for political parties associated with the reporting jurisdiction and for associating the parties with candidates and contests;
- **Person** – for address and contact information for persons associated with the jurisdiction, including boards of authorities, candidates, current office holders, or other election-related officials;
- **Contest** – for contests and for linking together the major elements needed for contests such as candidates and contest options. There are five types of Contests:
    - *BallotMeasureContest* – for contests involving ballot measures (not shown in Figure 11);
    - *CandidateContest* – for contests involving candidates;
    - ControllingContest –—for a contest that controls other contests;
    - *PartyContest* – for party-list selections (not shown in Figure 11); and
    - *RetentionContest* – for judicial or other types of retention contests (not shown in Figure 11);
- **Candidate** – for describing candidates so that they can be associated with contests;
- **ContestSelection** – for describing the types of contest selections in a contest and associating them with election results:
    - *BallotMeasureSelection* – used if the contest selection is for a ballot measure (not shown in Figure 11);
    - *CandidateSelection* – used if the contest selection is for a candidate; and

  - *PartySelection* – used if the contest selection is for a party as in straight party or party-list selection (not shown in Figure 11);

- **BallotStyle** – for describing the contests and contest selections on a ballot and linking them to geopolitical units such as precincts;
- **PhysicalContest** – for physical details of a contest according to a particular ballot format;
- **PhysicalContestOption** – for physical details associated with a particular contest option;
- **OptionPosition** – for describing the physical position of an option on a ballot style;
- **WriteInPosition** – for describing a physical position of an area where a write-in can be made;
- **BallotFormat** – for describing details common across ballot styles produced by a particular manufacturer's application.

The attributes associated with the UML classes correspond closely to the XML and JSON schemas' attributes and sub-elements associated with the major elements, described in Section 5.

### 4.1.1. Understanding Relationships Between Classes

The major classes in the UML model result in major elements in the schemas, and the different types of relationships between the UML classes determine how the elements are structured (linked) in the schema. There are three types of relationships between the classes:

**Directed Composition**: see Figure 12, BallotDefinition and Election, read as, "A ballot definition is composed of elections." In the XML schema for example, the *<Election>* element will be generated as a sub-element of the *<BallotDefinition>* element. A directed composition relationship has a closed diamond at one end and an arrow pointing to the composing class:

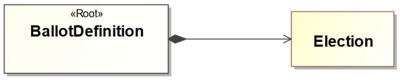


Figure 12 – Directed Composition Example

**Is a Type of (Generalization)**: see Figure 13, Contest and CandidateContest, read as, "A candidate contest is a type of contest." Contest is an abstract class; it is "implemented" by its concrete classes such as CandidateContest. In the XML schema, Contest will be generated as an abstract type and serve as an extension base to the *<CandidateContest>* element. A generalization relationship has an open triangle at one end, pointing from the subclass to the superclass:

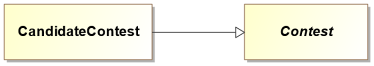


Figure 13 – "Is a Type of" Example

**Directed Association**: see Figure 14, Candidate and Party, read as, "A candidate is associated with or linked to a party." In the XML schema, the *<Candidate>* element will include a *<PartyId>* element, which will contain an identifier associated with a *<Party>* element. A directed association has an arrow at one end, goes in one direction, and serves to link the class to another associated class, e.g., the party linked to the candidate:

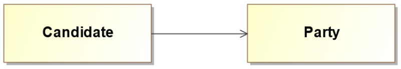


Figure 14 – Directed Association Example

For the directed composition associations in Figure 12, one can see that BallotDefinition is composed of BallotFormat, Election, GpUnit, Office, Party, and Person. Election is composed of BallotStyle, Candidate, and Contest. Contest is composed of ContestSelection.

For the generalization relationships in Figure 13, one can see that CandidateContest and ControllingContest is a type of Contest. ReportingUnit is a type of GpUnit. CandidateSelection is a type of ContestSelection.

Lastly, for the directed associations in Figure 14, one can see that many of the classes are associated with each other, as one would expect. For example, Party is associated with (or linked to) Candidate just as a political party would be associated with or linked to a candidate. Candidate is associated or linked to CandidateSelection, just as a selection on the ballot in a candidate contest would be for a specific candidate.

## 4.2. Mapping the Model to XML and JSON

This section contains an overview of how the UML model is mapped to the XML and JSON data formats. The schemas are available for download; see Appendix D for file locations.

### 4.2.1. Root Element

The UML class model is a graph data structure in the sense that there is no hierarchy implied by the model. However, the two supported implementation formats, XML and JSON, are hierarchical in structure (also called a tree). All hierarchies must start with a root, which is indicated in the UML model as a class with the «root» stereotype applied. In Figure 11, the BallotDefinition class will be generated in XML as the root element *<BallotDefinition>*.

### 4.2.2. Elements and Attributes

In the UML model, the classes become major elements in the XML schema. The attributes of a class become XML sub-elements of the major elements.

JSON (as with UML) only provides a single data structure for presenting information, objects, whereas XML provides two data structures for presenting data, elements and attributes. If a UML attribute has the «xmlAttribute» stereotype or the class contains an attribute with the «simpleContent» stereotype, the attribute becomes an XML attribute. The «simpleContent» stereotype indicates that the attribute is the target for the character data of the XML element. For example, the following UML represents a class named File with two attributes:

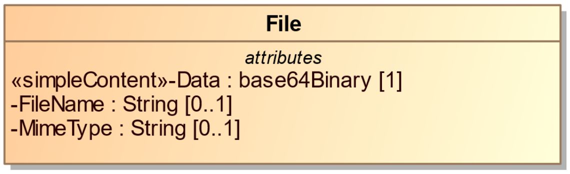

File
attributes
«simpleContent»-Data : base64Binary [1]
-FileName : String [0..1]
-MimeType : String [0..1]

Figure 15 – File

An example using XML is:

```
<File fileName="String"
mimeType="string">UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</File>
```

### 4.2.3. Data Type Mapping

The UML model uses a number of primitive and specialized types based on the XML schema, as shown in Table :

| UML | XML | JSON | JSON Format |
|---|---|---|---|
| String | xsd:string | string | |
| date | xsd:date | string | date |
| Boolean | xsd:boolean | boolean | boolean |
| base64Binary | xsd:base64Binary | string | byte |
| anyURI | xsd:anyURI | string | uri |
| float | xsd:float | number | |

Table 1 - UML, XML, JSON type equivalents

### 4.2.4. Representing Multiplicities

Some attributes in the UML model can repeat. Repetition is represented by multiplicities whose lower or upper cardinality is greater than one, e.g., 1..2, 2..4, 0..*, etc. An asterisk (*) for the upper cardinality represents an unbounded number of repetitions. A “1” for the lower cardinality indicates that the property is required.

UML attributes with multiplicities greater than one are represented in XML as repeating elements. For example:

```
<Uri Annotation="mobile">http://mobile.samplesite.com/</Uri>
<Uri Annotation="desktop">http://www.samplesite.com/</Uri>
```

In JSON, UML attributes with multiplicities greater than one are represented as an array of objects. For example:

```
"Uri":
    [
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "mobile",
            "Content": "http://mobile.samplesite.com/"
        },
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "desktop",
            "Content": "http://www.samplesite.com/"
        }
    ]
```

Even if an implementer wants to provide a single occurrence of an attribute, it must be wrapped in an array:

```
"Uri":
    [
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "mobile",
            "Content": "http://mobile.samplesite.com/"
        }
    ]
```

### 4.2.5. References

Some classes of data may be referenced repeatedly, for example political parties or geopolitical units. It would make sense to define single instances of these classes and then reference them whenever they are needed as opposed to creating new instances. The UML model represents these references as directed associations between classes.

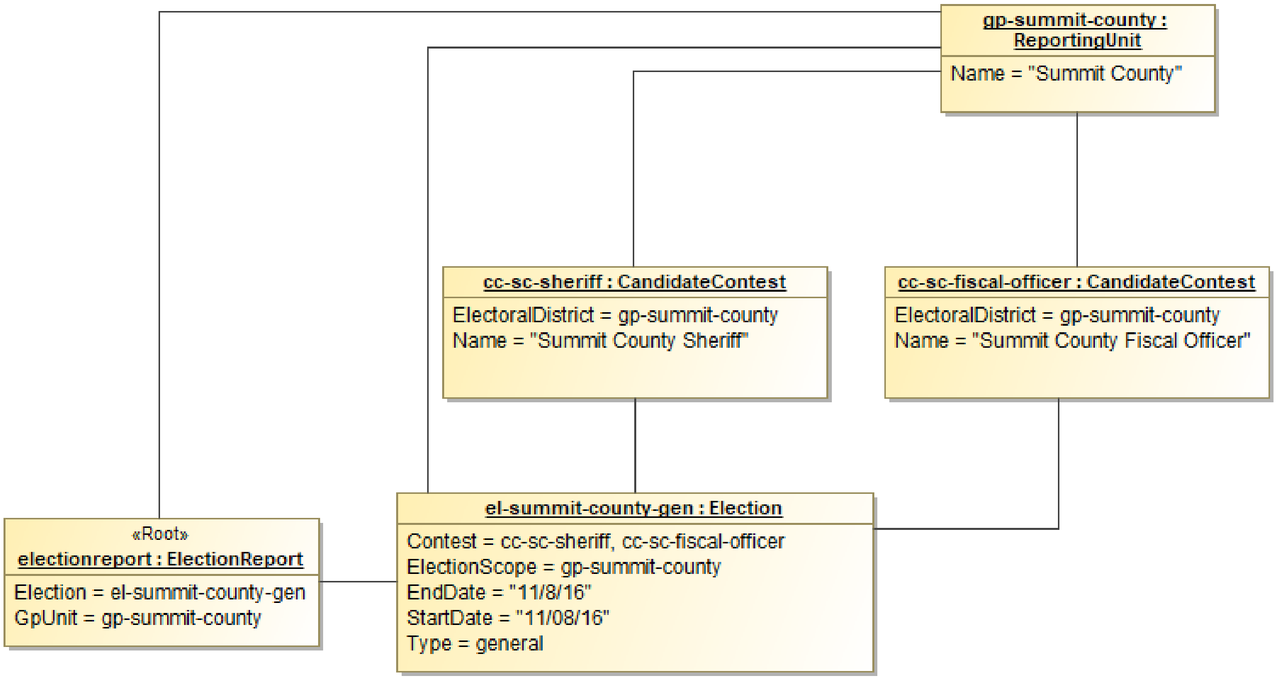

gp-summit-county : ReportingUnit
Name = "Summit County"
cc-sc-sheriff : CandidateContest
ElectoralDistrict = gp-summit-county
Name = "Summit County Sheriff"
cc-sc-fiscal-officer : CandidateContest
ElectoralDistrict = gp-summit-county
Name = "Summit County Fiscal Officer"
«Root»
electionreport : ElectionReport
Election = el-summit-county-gen
GpUnit = gp-summit-county
el-summit-county-gen : Election
Contest = cc-sc-sheriff, cc-sc-fiscal-officer
ElectionScope = gp-summit-county
EndDate = "11/8/16"
StartDate = "11/08/16"
Type = general

Figure 16 - UML Instance using references

In this figure, two contests are defined; they are of type CandidateContest and both have the same election district: Summit County. Only one reporting unit for Summit County needs to be defined, however, because the contests can reference the reporting unit as needed.

The XML representing the classes in Figure 16 is as follows:

```
<BallotDefinition          >
        <Elec  tion>
                <Contest ObjectId="cc        - sc - sherif" xsi:type="CandidateContest">
                        <ElectionDistrictId>gp           - summit - county</ElectionDistrictId>
                        <Name>Summit County Sheriff</Name>
                </Contest>
                <Contest ObjectId="cc        - sc - fiscal     - officer"
               xsi:type="Candida       teContest">
                        <ElectionDistrictId>gp           - summit - county</ElectionDistrictId>
                        <Name>Summit County Fiscal Officer</Name>
                </Contest>
                <ElectionScopeId>gp      - summit - county</ElectionScopeId>
                <Type>general</Type>
        </Election>
        <GpUnit ObjectId="gp       - summit - county">
                <Name>Summit County</Name>
        </GpUnit>
</BallotDefinition>
```

XML provides two built in types for handling references: one for establishing the reusable element (xsd:ID) and one for referencing it (xsd:IDREF). Identifiers are defined using the ObjectId attribute. The name of the identifier must be unique across the XML instance and conform to restrictions specified by the xsd:NCName datatype. *An xsd:NCName value must start with either a letter or underscore and may contain only letters, digits, underscores, hyphens, and periods.* JSON references are handled by the use of an @id property. An example follows:

```
{
        "@type": "BallotDefinition.BallotDefinition",
        "Election": [
                {
                        "@type": "BallotDefinition.Election",
                        "Contest": [
                                {
                                        "@id": "cc-sc-sheriff",
                                        "@type": "BallotDefinition.CandidateContest",
                                        "ElectionDistrict": "gp-summit-county",
                                        "Name": "Summit County Sheriff"
                                },
                                {
                                        "@id": "cc-sc-fiscal-officer",
                                        "@type": "BallotDefinition.CandidateContest",
                                        "ElectionDistrict": "gp-summit-county",
                                        "Name": "Summit County Fiscal Officer"
                                }
                        ],
                        "Type": "general"
                }
        ],
        "GpUnit": [
                {
                        "@id": "gp-summit-county",
                        "@type": "BallotDefinition.ReportingUnit",
                        "Name": "Summit County"
                }
        ]
}
```

# 5. Ballot Definition UML Model Documentation

This section contains documentation and discussion of the features included in the BD UML model. As noted previously, this model was used in deriving the XML and JSON schemas, and the schema usage closely follows that of the UML model.

The UML classes are described first, followed by the enumerations. Classes appear in alphabetical order except when a class is a subtype of another class. In that case, the subtype appears after its parent. Each description contains an image of the class (from the UML model) and a table containing details about each of the class's attributes. To denote that certain class properties derive from the class's associations with other classes, curly braces are used around those attribute names, e.g., if ClassA has an association with ClassB that is named "Automobile", then the table of attributes for ClassA would include "{Automobile}" as one of the properties.

Class properties and enumeration values are alphabetically ordered with some exceptions, e.g., "OtherStatus" comes after "Status" in some classes, and "other" is last in enumeration values.

## 5.1. Class AnnotatedString

Used as a type for character strings; it adds a 32-character annotation to a character string. AnnotatedString is a subtype of string.

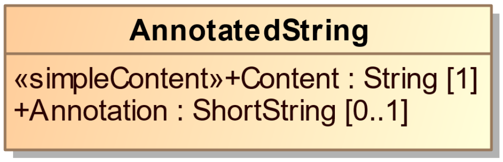

AnnotatedString
«simpleContent»+Content : String [1]
+Annotation : ShortString [0..1]

Figure 17 – AnnotatedString

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Annotation | 0..1 | ShortString | An annotation of up to 32 characters associated with a character string. |
| Content | 1 | String | The string to be annotated. |

Table 2 - Describes AnnotatedString properties

## 5.2. Class AnnotatedUri

Used as a type for character strings that represent Uniform Resource Identifiers (URI); it adds a 32-character annotation to a character string.

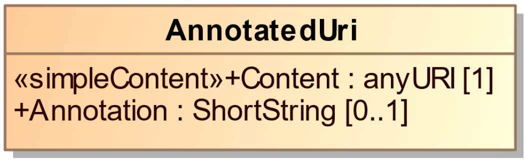

AnnotatedUri
«simpleContent»+Content : anyURI [1]
+Annotation : ShortString [0..1]

Figure 18 – AnnotatedUri

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Annotation | 0..1 | ShortString | An annotation of up to 32 characters associated with a character string. |
| Content | 1 | anyURI | The URI to be annotated. |

Table 3 - Describes AnnotatedUri properties

## 5.3. Class BallotDefinition

For defining items pertaining to the issuer and version of the definition and when it was generated.

BallotDefinition references the major elements that are not necessarily specific to an election and that therefore can exist in a logical ballot definition: GpUnit, Office and OfficeGroup, Party, Person, and Election.

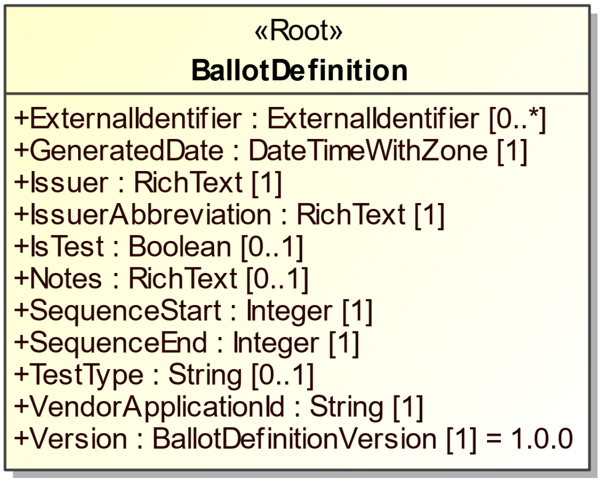

«Root»
BallotDefinition
+ExternalIdentifier : ExternalIdentifier [0..*]
+GeneratedDate : DateTimeWithZone [1]
+Issuer : RichText [1]
+IssuerAbbreviation : RichText [1]
+IsTest : Boolean [0..1]
+Notes : RichText [0..1]
+SequenceStart : Integer [1]
+SequenceEnd : Integer [1]
+TestType : String [0..1]
+VendorApplicationId : String [1]
+Version : BallotDefinitionVersion [1] = 1.0.0

Figure 19 – BallotDefinition

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| {BallotFormat} | 0..* | BallotFormat | For associating ballot formats with the definition. |
| {Election} | 0..* | Election | For associating elections with the definition. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating an ID with the definition. |
| GeneratedDate | 1 | DateTimeWithZone | Identifies the date and time that the ballot definition was generated. |
| {GpUnit} | 0..* | GpUnit | For associating geopolitical units with the definition. |
| {Header} | 0..* | Header | For associating headers with parts of a ballot style. |
| Issuer | 1 | RichText | Identification of the definition issuer. |
| IssuerAbbreviation | 1 | RichText | An abbreviation of the definition issuer such as the 2-character U.S. Census Bureau abbreviation of the state providing the election definition, e.g., AL, TX, MN, etc. |
| IsTest | 0..1 | Boolean | Used to indicate whether the definition is a test. Assumed to be “false” if not present. |
| Notes | 0..1 | RichText | For including an arbitrary message with the definition. |
| {Office} | 0..* | Office | For associating offices with the definition. |
| {OfficeGroup} | 0..* | OfficeGroup | For associating a name for a grouping of offices with the definition. |
| {Party} | 0..* | Party | For associating parties with the definition. |
| {Person} | 0..* | Person | For associating persons with the definition. |
| SequenceStart | 1 | Integer | The report’s number as part of a sequence of reports, used with SequenceEnd so as to be read as, e.g., 1 of 1, 1 of 2, 2 of 2, etc. Starts with “1”. |
| SequenceEnd | 1 | Integer | The upper bound of the sequence; e.g., “1” if there is only 1 report, “2” if there are two reports in the sequence, etc. |
| {Shape} | 0..* | Shape | For associating parts of the ballot with geometric shapes. |
| TestType | 0..1 | String | A description of the type of test, e.g., pre-election, logic and accuracy, etc. |

| | | | |
|---|---|---|---|
| VendorApplicationId | 1 | String | An identifier of the vendor application generating the ballot definition, e.g., X-EMS version 3.1.a. |
| Version | 1 | BallotDefinitionVersion | To identify the version of the BD specification being used, i.e., version 1.0.0. This will need to be updated for different versions of the specification. |

Table 4 - Describes BallotDefinition properties

## 5.4. Class BallotFormat

For providing details related to the concrete presentation of a ballot appropriate to the particular voting technology being used. All physical details of a ballot style are associated with a particular BallotFormat.

BallotFormat sets the MeasurementUnit appropriate for the marking surface used.

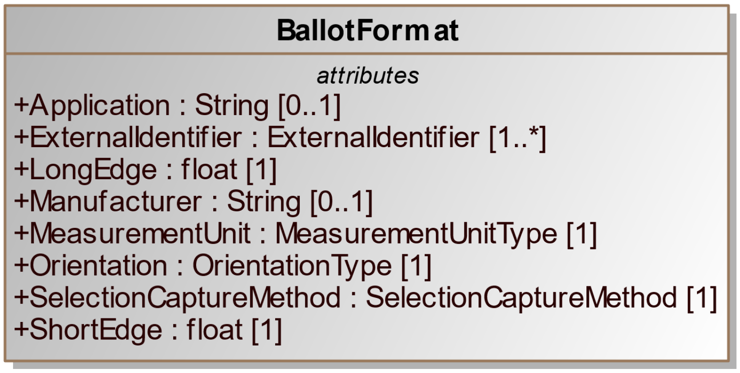

BallotFormat
attributes
+Application : String [0..1]
+ExternalIdentifier : ExternalIdentifier [1..*]
+LongEdge : float [1]
+Manufacturer : String [0..1]
+MeasurementUnit : MeasurementUnitType [1]
+Orientation : OrientationType [1]
+SelectionCaptureMethod : SelectionCaptureMethod [1]
+ShortEdge : float [1]

Figure 20 – BallotFormat

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Application | 0..1 | String | The application that generates ballots in the ballot format. |
| ExternalIdentifier | 1..* | ExternalIdentifier | For associating IDs with the ballot format. |
| {FiducialMark} | 0..* | FiducialMark | For associating a ballot format with one or more global fiducial marks appearing on a ballot. |
| LongEdge | 1 | float | Measurement of the long edge of the ballot sheet. |
| Manufacturer | 0..1 | String | Manufacturer of the ballot format. |
| {mCDFArea} | 0..* | mCDFArea | For locating the data encoded using a mCDF Profile (e.g., mCDF Profile for Contest Selection Capture). |
| MeasurementUnit | 1 | MeasurementUnitType | The measurement units used to express locations of ballot content, such as contest option positions. |
| Orientation | 1 | OrientationType | Orientation of the page relative to the coordinate plane. |
| SelectionCaptureMethod | 1 | SelectionCaptureMethod | The method used to read contest selections or indications on the ballot. |
| ShortEdge | 1 | float | Measurement of the short edge of the ballot sheet. |

Table 5 - Describes BallotFormat properties

## 5.5. Class BallotStyle

For defining a ballot style composed of ordered content (i.e., Headers or Contests) and their contest selections, and associating the ballot style with a political party, a reference to an image of the ballot, and a reference to a precinct or other geopolitical unit that the ballot is unique to. Election includes BallotStyle.

BallotStyle references OrderedContent to include content that appears on that ballot style. To preserve any rotation associated with the ballot, it is expected that the generating application will list the occurrences of OrderedContest in the order as on the ballot for the associated geopolitical unit.

BallotStyle references one or more GpUnit instances defined for the associated precincts or split precincts. If the ballot style is associated with multiple precincts (or other geographies), multiple references to the precinct GpUnit instances can be included.

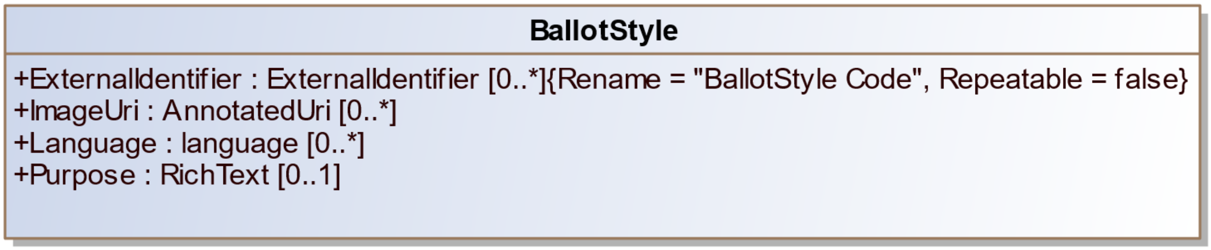

BallotStyle
+ExternalIdentifier : ExternalIdentifier [0..*]{Rename = "BallotStyle Code", Repeatable = false}
+ImageUri : AnnotatedUri [0..*]
+Language : language [0..*]
+Purpose : RichText [0..1]

Figure 21 – BallotStyle

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating IDs with the ballot style. |
| { GpUnit } | 1..* | GpUnit | Unique identifier for one or more GpUnit instances. For associating specific election administrative areas (e.g., precincts or splits) with the ballot style. |
| ImageUri | 0..* | AnnotatedUri | URI for a sample ballot image. |
| Language | 0..* | language | For the written languages appearing on the ballot style. |
| { OrderedContent } | 0..* | OrderedContent | For associating a ballot style with ballot content, such as contests or headers. |
| { Party } | 0..* | Party | For associating one or more parties with the ballot style. |
| Purpose | 0..1 | RichText | Freeform text to further describe the purpose (e.g., federal-only ballot) of the ballot style. |

Table 6 - Describes BallotStyle properties

## 5.6. Class BoundedObject

A class representing a bounding box. The box represents in absolute coordinate space its location and size. Its location is given by x,y coordinates, size by h,w values, the side of the sheet it appears on and the sheet number of the ballot style.

PhysicalContest includes BoundedObject.

BoundedObject has three subclasses, used contextually:

- FiducialMark, included by BallotFormat
- mCDFArea, included by BallotFormat
- OptionPosition, included by PhysicalContestOption
- WriteInPosition, included by PhysicalContestOption

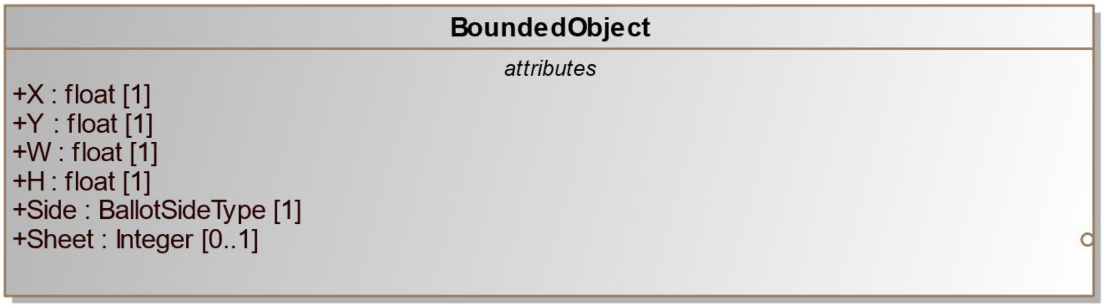


Figure 22 – BoundedObject

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| H | 1 | float | Height in ballot format measurement units. |
| Sheet | 0..1 | Integer | The ballot sheet the bounded object appears on. Sheets start at 1 and increase monotonically. Does not apply to mCDFArea, or global fiducials. |
| Side | 1 | BallotSideType | The side of the sheet of paper the bounded object appears. |
| W | 1 | float | Width in ballot format measurement units. |
| X | 1 | float | X-coordinate for locating a bounded object (top left anchor). |
| Y | 1 | float | Y-coordinate for locating a bounded object (top left anchor). |

Table 7 - Describes BoundedObject properties

## 5.7. Class FiducialMark

FiducialMark is a subclass of BoundedObject for geometry found at fixed positions of the page. BallotFormat references FiducialMark for geometry found at fixed locations on the ballot. PhysicalContest references FiducialMark for geometry proximate to the contest.

Figure 23 – FiducialMark

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Shape} | 1 | Shape | For associating a FiducialMark with the geometric shape that represents it. |

Table 8 - Describes FiducialMark properties

## 5.8. Class mCDFArea

A part of the ballot containing an instantiation of the microCDF. It inherits the properties of BoundedObject.

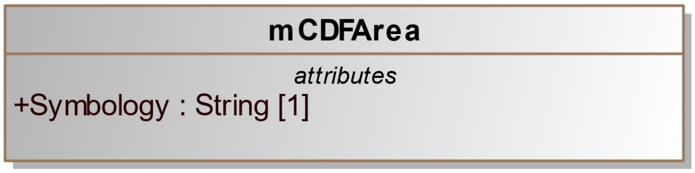


Figure 24 – mCDFArea

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Symbology | 1 | String | The name of the symbology used. |

Table 9 - Describes mCDFArea properties

## 5.9. Class OptionPosition

For associating a physical contest option to the location(s) where a selection can be indicated. It inherits the properties of BoundedObject.

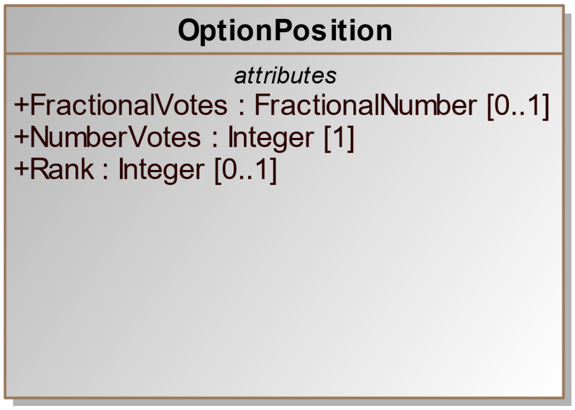

OptionPosition
attributes
+FractionalVotes : FractionalNumber [0..1]
+NumberVotes : Integer [1]
+Rank : Integer [0..1]

Figure 25 – OptionPosition

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| FractionalVotes | 0..1 | FractionalNumber | The proper fractional number of votes represented by the option position. |
| { Indicator } | 0..1 | Shape | For associating an OptionPosition with the shape that conveys the expected location of a mark. |
| NumberVotes | 1 | Integer | The number of votes represented by the option position. |
| Rank | 0..1 | Integer | The rank represented by the option position, if a RCV contest. |

Table 10 - Describes OptionPosition properties

## 5.10. Class WriteInPosition

Response area for representing a particular candidate, e.g., John Smith hand printed. It inherits the properties of BoundedObject.

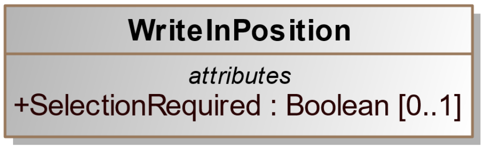


Figure 26 – WriteInPosition

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| SelectionRequired | 0..1 | Boolean | Whether a selection in an OptionPosition is required in order to count the write-in. Assumed to be false if not present. |

Table 11 - Describes WriteInPosition properties

## 5.11. Class Candidate

For defining information about a candidate in a contest. CandidateOption references Candidate instances to associate one or more candidates with a contest selection. Election includes Candidate.

Candidate uses the Party association to reference the candidate's political party. If the candidate is endorsed by other parties for a particular contest, the endorsing parties are referenced using the CandidateOption attribute.

ExternalIdentifier can be used to associate IDs with the candidate. If the type is not listed in enumeration IdentifierType, use other and include the type (that is not listed in the enumeration) in OtherType.

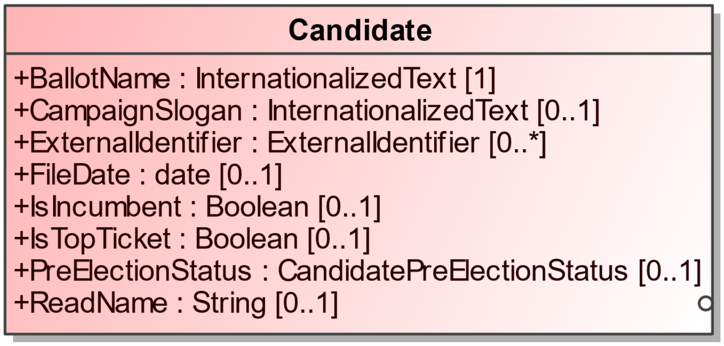

Candidate
+BallotName : InternationalizedText [1]
+CampaignSlogan : InternationalizedText [0..1]
+ExternalIdentifier : ExternalIdentifier [0..*]
+FileDate : date [0..1]
+IsIncumbent : Boolean [0..1]
+IsTopTicket : Boolean [0..1]
+PreElectionStatus : CandidatePreElectionStatus [0..1]
+ReadName : String [0..1]

Figure 27 – Candidate

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| BallotName | 1 | InternationalizedText | For the candidate's name as listed on the ballot. |
| CampaignSlogan | 0..1 | InternationalizedText | The slogan or motto used by the candidate in their campaign. |
| {ContactInformation} | 0..1 | ContactInformation | For associating contact information for the candidate. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating IDs with the candidate. |
| FileDate | 0..1 | date | Date when the candidate filed for the contest. |
| IsIncumbent | 0..1 | Boolean | Boolean to indicate whether the candidate is the incumbent for the office associated with the contest. Assumed to be "false" if not present. |
| IsTopTicket | 0..1 | Boolean | Boolean to indicate whether the candidate is the top of a ticket that includes multiple candidates. Assumed to be "false" if not present. |
| {Party} | 0..1 | Party | For associating a party with the candidate. |
| {Person} | 0..1 | Person | For associating more detailed information about the candidate. |
| PreElectionStatus | 0..1 | CandidatePreElectionStatus | Registration status of the candidate, e.g., filed, qualified, etc. |
| ReadName | 0..1 | String | The name spelled phonetically for reading by a screen reader. |

Table 12 - Describes Candidate properties

## 5.12. Class ContactInformation

For defining contact information about objects such as persons, boards of authorities, organizations, etc. Election, ElectionAdministration, Person, GpUnit, and Office include ContactInformation.

To include an address for the contact, use multiple occurrences of AddressLine. It is expected that the generating application will list the name of the person/organization in the first occurrence of AddressLine, with subsequent ordered occurrences for street address, city, state, zip code, etc. Directions can be used to supply any additional address-related information that may appear in multiple languages.

ContactInformation includes LatLng so as to associate latitude/longitude with the contact address.

Email, Fax, and Phone are of type AnnotatedString, which permits up to a 32-character annotation to be associated with the data.

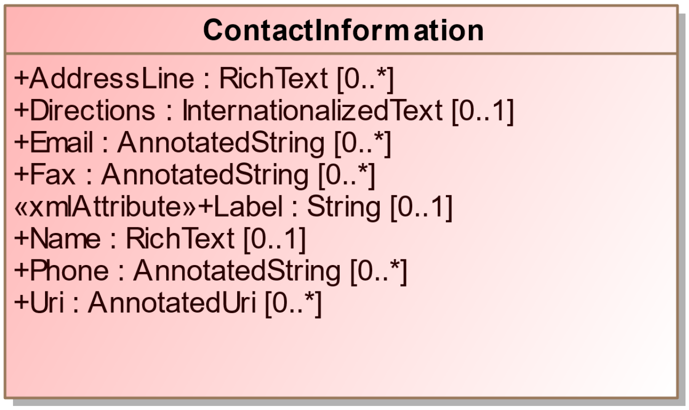

ContactInformation
+AddressLine : RichText [0..*]
+Directions : InternationalizedText [0..1]
+Email : AnnotatedString [0..*]
+Fax : AnnotatedString [0..*]
«xmlAttribute»+Label : String [0..1]
+Name : RichText [0..1]
+Phone : AnnotatedString [0..*]
+Uri : AnnotatedUri [0..*]

Figure 28 – ContactInformation

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| AddressLine | 0..* | RichText | For associating an address with the contact. |
| Directions | 0..1 | InternationalizedText | Directional information in addition to address information. |
| Email | 0..* | AnnotatedString | Email address associated with the contact. |
| Fax | 0..* | AnnotatedString | Fax number associated with the contact. |
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| {LatLng} | 0..1 | LatLng | For latitude and longitude information associated with the contact. |
| Name | 0..1 | RichText | Name associated with the contact. |
| Phone | 0..* | AnnotatedString | Phone number associated with the contact. |
| {Schedule} | 0..* | Schedule | For associating a schedule with the contact. |
| Uri | 0..* | AnnotatedUri | URI associated with the contact. |

Table 13 - Describes ContactInformation properties

## 5.13. Class Contest

For defining a contest and linking the contest to the associated candidates, ballot measures, parties, or controlled contests. Election includes Contest.

Contest is an abstract class with four subclasses that get used according to the type of contest:

- BallotMeasureContest, used for a contest involving a ballot measure
- CandidateContest, used for a contest involving one or more candidates for an office
- ControllingContest, an abstract class used for a contest that affects contests under its control.
- RetentionContest, used for a judicial or other type of retention contest

Contest includes ContestOption to link the selections on the ballot to the contest, e.g., to link one or more candidates to a candidate contest. Like Contest, ContestOption is also an abstract class and has subclasses that correspond to those of Contest, as follows:

- BallotMeasureContest includes BallotMeasureOption
- CandidateContest includes CandidateOption
- The subtypes of ControllingContest include either ActivationOption or PartyOption
- PartyContest includes PartyOption
- RetentionContest includes BallotMeasureOption

Contest includes a required ElectionDistrict reference to a GpUnit defined for the geographical scope of the contest. For example, in a state senate contest, ElectionDistrict would reference a GpUnit of type ReportingUnit element defined for the district associated with the contest. Office also includes an optional reference that serves the same purpose. Note that for contests that are state-wide or county-wide and so forth, the same GpUnit defined for the state or county, etc., can be re-used.

SequenceOrder is used for results display ordering, i.e., to display contests according to a particular ordering. For example, “100” may indicate a U.S. Senatorial contest, “200” may indicate a U.S. Congressional contest, etc. SequenceOrder is not appropriate to use as the contest order on the ballot; contest order on each ballot can be preserved, however, using the BallotStyle element, which associates ballot styles with their corresponding precincts or other geopolitical units.

When including ExternalIdentifier elements, if the type is not listed in enumeration IdentifierType, use other and include the type (that is not listed in the enumeration) in OtherType.

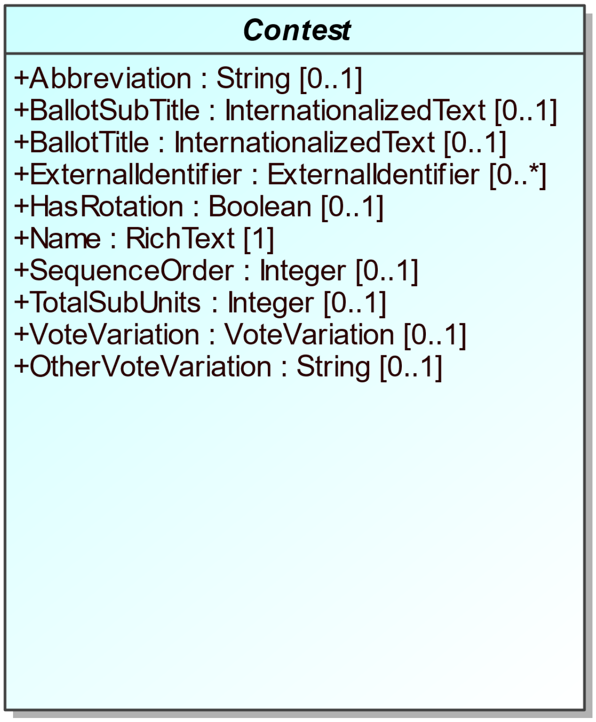

Contest
+Abbreviation : String [0..1]
+BallotSubTitle : InternationalizedText [0..1]
+BallotTitle : InternationalizedText [0..1]
+ExternalIdentifier : ExternalIdentifier [0..*]
+HasRotation : Boolean [0..1]
+Name : RichText [1]
+SequenceOrder : Integer [0..1]
+TotalSubUnits : Integer [0..1]
+VoteVariation : VoteVariation [0..1]
+OtherVoteVariation : String [0..1]

Figure 29 – Contest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Abbreviation | 0..1 | String | Abbreviation for the contest. |
| BallotSubTitle | 0..1 | InternationalizedText | Subtitle of the contest as it appears on the ballot. |
| BallotTitle | 0..1 | InternationalizedText | Title of the contest as it appears on the ballot. |
| {ContestOption} | 0..* | ContestOption | For associating contest options for the contest, e.g., candidates, ballot measure options. |
| {ElectionDistrict} | 1 | ReportingUnit | Link to a GpUnit instance. For associating the contest with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating an ID with the contest. |
| HasRotation | 0..1 | Boolean | Boolean to indicate whether the selections in the contest are rotated. Assumed to be “no” if not present. |
| Name | 1 | RichText | Name of the contest, not necessarily as it appears on the ballot. |
| SequenceOrder | 0..1 | Integer | Ordering for listing the contest for purposes of display. If not present, no order is assumed. |
| TotalSubUnits | 0..1 | Integer | Total number of subunits, e.g., precincts that have this contest on the ballot. |
| VoteVariation | 0..1 | VoteVariation | Vote variation associated with the contest, e.g., n-of-m. |
| OtherVoteVariation | 0..1 | String | For use when VoteVariation is other. |

Table 14 - Describes Contest properties

## 5.14. Class BallotMeasureContest

For ballot measure (i.e., referenda or a tax measure) and judicial retention contests. It inherits the attributes of Contest.

If the type of ballot measure is not listed in enumeration BallotMeasureType, use other and include the type (that is not listed in the enumeration) in OtherType.

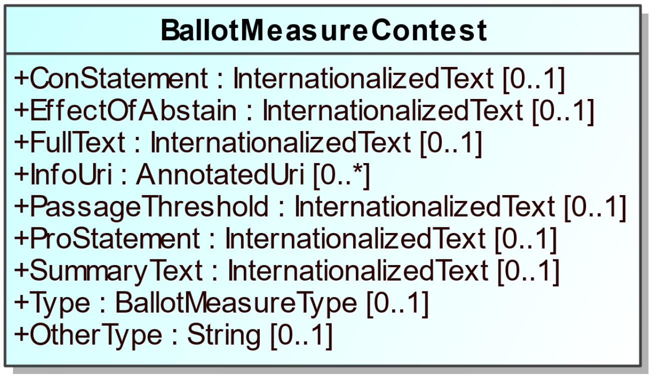

BallotMeasureContest
+ConStatement : InternationalizedText [0..1]
+EffectOfAbstain : InternationalizedText [0..1]
+FullText : InternationalizedText [0..1]
+InfoUri : AnnotatedUri [0..*]
+PassageThreshold : InternationalizedText [0..1]
+ProStatement : InternationalizedText [0..1]
+SummaryText : InternationalizedText [0..1]
+Type : BallotMeasureType [0..1]
+OtherType : String [0..1]

Figure 30 – BallotMeasureContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| ConStatement | 0..1 | InternationalizedText | For a statement on the ballot associated with a “no” vote. |
| EffectOfAbstain | 0..1 | InternationalizedText | For a statement on the ballot detailing the effect of abstaining from voting on the ballot measure. |
| FullText | 0..1 | InternationalizedText | For full text on the ballot of the ballot measure. |
| InfoUri | 0..* | AnnotatedUri | For associating a URI with the ballot measure contest. |
| PassageThreshold | 0..1 | InternationalizedText | For a statement on the ballot of the number or percentage of votes needed to approve or pass the ballot measure. |
| ProStatement | 0..1 | InternationalizedText | For a statement on the ballot associated with a “yes” vote. |
| SummaryText | 0..1 | InternationalizedText | For a summary on the ballot of the ballot measure. |
| Type | 0..1 | BallotMeasureType | For indicating the type of ballot measure. |
| OtherType | 0..1 | String | Used when BallotMeasureType is other. |

Table 15 - Describes BallotMeasureContest properties

## 5.15. Class RetentionContest

For judicial retention or other types of retention contests. Retention contests can be treated essentially as ballot measure contests, however this element differs from BallotMeasureContest in that it can include a reference to a candidate or the associated office.

This element uses BallotMeasureContest as a superclass. Therefore, it inherits the attributes of Contest as well as BallotMeasureContest.

Figure 31 – RetentionContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Candidate } | 1 | Candidate | Link to a Candidate instance. For associating a candidate with the retention contest. |
| { Office } | 0..1 | Office | Link to an Office instance. For associating an office description with the retention contest. |

Table 16 - Describes RetentionContest properties

## 5.16. Class CandidateContest

For a contest that involves selecting one or more candidates. It inherits the attributes of Contest.

This class optionally references Office and Party. If the candidate contest is associated with a ticket (of candidates) and each candidate in the ticket is associated with a separate office, the association to Office can reference each of the separate offices. For example, if the contest is for the state governor ticket but Governor and Lieutenant (Lt.) Governor are both separate offices, the association references first to the Office instance defined for the Governor's office and then to the Lt. Governor's office. In this case, it is expected that the generating application will list the multiple references according to a jurisdiction-defined ordering scheme, e.g., Governor first and Lt. Governor second.

Note that when using the CandidateOption class to associate the candidates with a contest selection for the contest, the order of the candidates should match the order of offices. Again, using the example of the state governor ticket, if the offices are listed with Governor first and Lt. Governor second, then the order of the candidates in the ContestOption instance should be identical, with the Governor candidate first and the Lt. Governor candidate second.

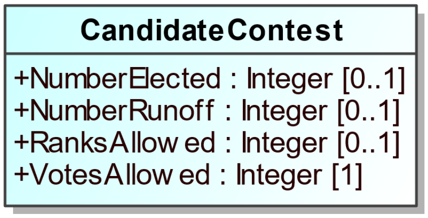

CandidateContest
+NumberElected : Integer [0..1]
+NumberRunoff : Integer [0..1]
+RanksAllowed : Integer [0..1]
+VotesAllowed : Integer [1]

Figure 32 – CandidateContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| NumberElected | 0..1 | Integer | Number of candidates that are elected in the contest ("n" of n-of-m). |
| NumberRunoff | 0..1 | Integer | The number of candidates in a runoff contest. |
| { Office } | 0..* | Office | For associating office descriptions. |
| { PrimaryParty } | 0..* | Party | For associating parties with the contest. |
| RanksAllowed | 0..1 | Integer | Maximum number of ranks per voter in this contest. |
| VotesAllowed | 1 | Integer | Maximum number of votes per voter in this contest. |

Table 17 - Describes CandidateContest properties

## 5.17. Class ControllingContest

For a contest that affects other contests under its control.

ControllingContest is an abstract subclass of Contest with three subclasses that get used according to the type of contest:

- ActivationContest, used for a contest that conditionally activates other contests
- PartyPreferenceContest, for a contest that activates partisan sections of the ballot
- StraightPartyContest, used for a contest for a straight party selection on the ballot

ControllingContest contains ControlledContest to link to the contests under its control.

Figure 33 – ControllingContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { ControlledContest } | 1..* | Contest | A contest on a ballot whose state is dependent on the selections made in a separate controlling contest. |

Table 18 - Describes ControllingContest properties

## 5.18. Class ActivationContest

For a contest containing an option that conditionally activates other contests on the ballot.

This element uses ControllingContest as a superclass. Therefore, it inherits the attributes of Contest as well as ControllingContest.

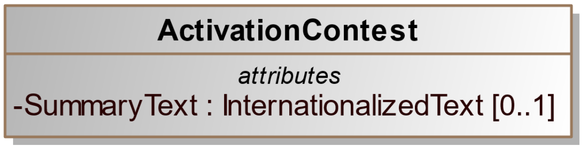


Figure 34 – ActivationContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Candidate } | 0..1 | Candidate | Link to a Candidate instance. For associating a candidate with the activation contest (e.g., a candidate subject to recall). |
| SummaryText | 0..1 | InternationalizedText | For a summary on the ballot of the activation contest, e.g., "Should John Smith be recalled?". |

Table 19 - Describes ActivationContest properties

## 5.19. Class PartyPreferenceContest

For a controlling contest that may appear on an open primary ballot and allows the voter to select which political party's primary they wish to vote. A valid selection in this contest protects the voter from voiding the partisan selection of the ballot by making selections in more than one party's contest.

This element uses ControllingContest as a superclass. Therefore, it inherits the attributes of Contest as well as ControllingContest.

Figure 35 – PartyPreferenceContest

## 5.20. Class StraightPartyContest

For a contest that involves choosing a party, typically for a straight party selection on the ballot.

This element uses ControllingContest as a superclass. Therefore, it inherits the attributes of Contest as well as ControllingContest.

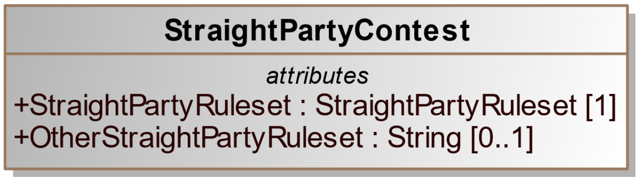

StraightPartyContest
attributes
+StraightPartyRuleset : StraightPartyRuleset [1]
+OtherStraightPartyRuleset : String [0..1]

Figure 36 – StraightPartyContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| StraightPartyRuleset | 1 | StraightPartyRuleset | The ruleset specifying how to consider indications made in the controlled contests. |
| OtherStraightPartyRuleset | 0..1 | String | For use when StraightPartyRuleset value is other. |

Table 20 - Describes StraightPartyContest properties

## 5.21. Class PartyContest

Use PartyContest for a contest that where voters choose a party on the ballot. When a selection in this contest causes other indirect selections, StraightPartyContest should be used instead.

It inherits the attributes of Contest.

Figure 37 – PartyContest

## 5.22. Class ContestOption

Used for the contest selections in a contest (e.g., for candidates, for ballot measures). Contest includes ContestOption.

ContestOption is an abstract class with four subclasses that get used according to the type of contest:

- ActivationOption, used if the contest type is for an activation contest
- BallotMeasureOption, used if the contest type is for a ballot measure, including for retentions
- CandidateOption, used if the contest type is for one or more candidates, to link the contest selection to the candidate instances and endorsement parties; and
- PartyOption, used if the contest type is for a party, e.g., for a party list contest.

SequenceOrder is included to specify an ordering for the contest selections for purposes of display only. The original ballot ordering can be preserved, however, by using the BallotStyle class.

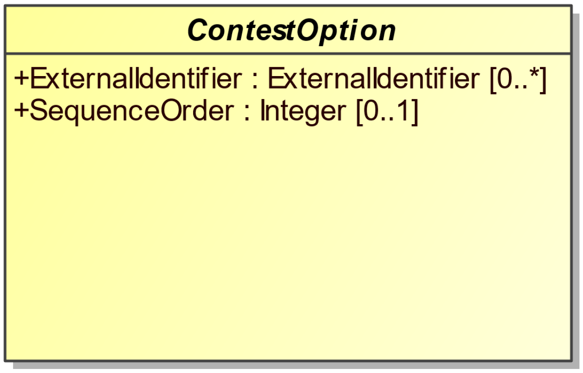

ContestOption
+ExternalIdentifier : ExternalIdentifier [0..*]
+SequenceOrder : Integer [0..1]

Figure 38 – ContestOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating an ID with the contest option. |
| SequenceOrder | 0..1 | Integer | Order in which the candidate is listed on the ballot for purposes of display. If not present, no order is assumed. |

Table 21 - Describes ContestOption properties

## 5.23. Class ActivationOption

For the contest selections in an ActivationContest. It inherits the attributes of ContestOption.

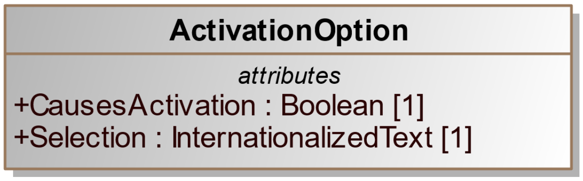

ActivationOption
attributes
+CausesActivation : Boolean [1]
+Selection : InternationalizedText [1]

Figure 39 – ActivationOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| CausesActivation | 1 | Boolean | Boolean to indicate whether this selection, when chosen by the voter will cause controlled contests to activate. |
| Selection | 1 | InternationalizedText | The text on the ballot associated with the selection. |

Table 22 - Describes ActivationOption properties

## 5.24. Class BallotMeasureOption

For a contest selection in a ballot measure contest. Because judicial or other retention contests are often treated like ballot measure contests, this element can be used also for retention contests. It inherits the attributes of ContestOption.

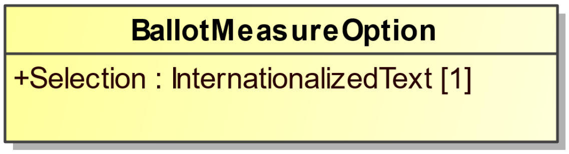


Figure 40 – BallotMeasureOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Selection | 1 | InternationalizedText | Contains the text used to indicate a vote for or against the ballot measure, e.g., “yes”, “no”. |

Table 23 - Describes BallotMeasureOption properties

## 5.25. Class CandidateOption

For the contest selections in a candidate contest, including for write-ins. It inherits the attributes of ContestOption. References to multiple Candidate instances can be included if necessary, e.g., when the contest selection would be for a ticket of candidates (unless the ticket itself is defined as a candidate).

EndorsementParty is used to reference any associated endorsement parties other than the specific party of the candidate (Candidate references Party for that purpose). For example, if a candidate of one party is also endorsed by a second party, use EndorsementParty to reference the second party. A second example would be for ballot fusion as used in some states, where the same candidate is listed multiple times in the same contest, but with different endorsement parties.

When multiple candidates are referenced for a ticket and the ordering of the candidates is important to preserve, it is expected that the generating application will list the references to Candidate instances according to the ordering scheme in place. For example, if the contest is for a ticket in which each candidate is associated with a different office, then the order of the candidates should match the same ordering of the <Office> element references within <OfficeIds> in the <Contest xsi:type="CandidateContest" ... /> element.

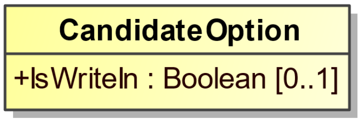


Figure 41 – CandidateOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Candidate } | 0..* | Candidate | For associating a candidate with the candidate selection on the ballot and for cases where the contest selection is for multiple candidates, e.g., a ticket. |
| IsWriteIn | 0..1 | Boolean | Indicates whether the candidate is a write-in, e.g., true or false. Assumed to be false if not present. |
| { EndorsementParty } | 0..* | Party | For associating one or more endorsing parties with the candidate selection. |

Table 24 - Describes CandidateOption properties

## 5.26. Class PartyOption

For a contest selection involving a party such as for a party list selection or straight party selection on the ballot. It inherits the attributes of ContestOption.

Figure 42 – PartyOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Party } | 1..* | Party | Link to one or more Party instances. For associating one or more parties with the party selection. |

Table 25 - Describes PartyOption properties

## 5.27. Class Election

For defining details of the election and associated information such as candidates and contests.

Election includes links to the major instances that are specific to an election: BallotStyle, Candidate, and Contest.

Election includes a required reference to ElectionScope, which links to a GpUnit instance for the purpose of identifying the geographical scope of the election. For example, for an election within a county, ElectionScope would reference a GpUnit defined for the county. If it is desired to include election authority information, the GpUnit can include ElectionAdministration.

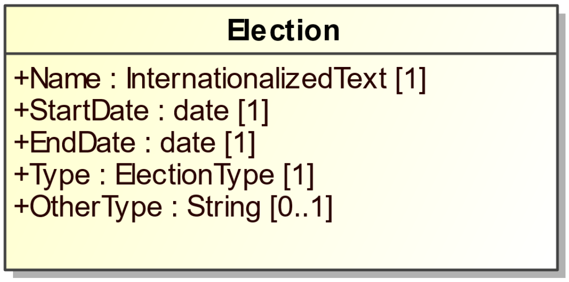


Figure 43 – Election

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| {BallotStyle} | 0..* | BallotStyle | For defining ballot styles associated with the election. |
| {Candidate} | 0..* | Candidate | For defining candidates associated with the election. |
| {ContactInformation} | 0..1 | ContactInformation | For associating various contact information with the election. |
| {Contest} | 0..* | Contest | For defining contests associated with the election. |
| {ElectionScope} | 1 | ReportingUnit | Unique identifier for a GpUnit element. For associating the election with a reporting unit that represents the geographical scope of the election, e.g., a state, a county, etc. |
| {ExternalIdentifier} | 0..* | ExternalIdentifier | For associating IDs with the election. |
| Name | 1 | InternationalizedText | For including a name for the election; the name could be the same name as appears on the ballot. |
| StartDate | 1 | date | Calendar start date of the election, e.g., “2018-11-04”. |
| EndDate | 1 | date | Calendar end date of the election; for a typical one-day election, the end date is the same as the start date. |
| Type | 1 | ElectionType | Enumerated type of election, e.g., partisan-primary, open-primary, etc. |
| OtherType | 0..1 | String | Used when Type is other. |

Table 26 - Describes Election properties

## 5.28. Class ElectionAdministration

Used to provide various information about an election authority. ReportingUnit includes ElectionAdministration.

ElectionAdministration includes ContactInformation for the election authority and, using ElectionOfficialPerson references one or more Person instances defined for individuals/organizations associated with the election authority.

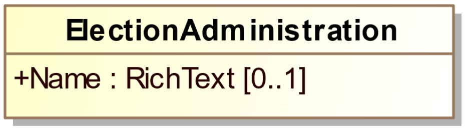


Figure 44 – ElectionAdministration

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { ContactInformation } | 0..1 | ContactInformation | For including various contact information. |
| { ElectionOfficialPerson } | 0..* | Person | Unique identifier for one or more Person instances defined for the election authority. |
| Name | 0..1 | RichText | Name of the election authority. |

Table 27 - Describes ElectionAdministration properties

## 5.29. Class ExternalIdentifier

For associating a jurisdiction's codes, i.e., identifiers, with objects such as candidates, offices, or geopolitical units such as counties, towns, precincts, etc. Multiple occurrences of ExternalIdentifier can be used to associate multiple codes, e.g., if there is a desire to associate multiple codes with a particular object such as FIPS (Federal Information Processing Standard) codes as well as OCD-IDs (Open Civic Data Identifiers).

For elements that link to ExternalIdentifier instances, if the type is not listed in enumeration IdentifierType, use other and include the type (that is not listed in the enumeration) in OtherType.

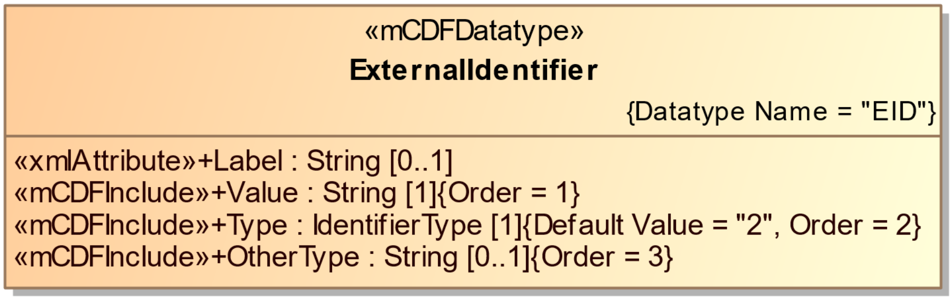

«mCDFDatatype»
ExternalIdentifier
{Datatype Name = "EID"}
«xmlAttribute»+Label : String [0..1]
«mCDFInclude»+Value : String [1]{Order = 1}
«mCDFInclude»+Type : IdentifierType [1]{Default Value = "2", Order = 2}
«mCDFInclude»+OtherType : String [0..1]{Order = 3}

Figure 45 – ExternalIdentifier

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| Type | 1 | IdentifierType | An identifier type, e.g., FIPS. |
| OtherType | 0..1 | String | Used when IdentifierType value is other. |
| Value | 1 | String | The identifier used by the jurisdiction. |

Table 28 - Describes ExternalIdentifier properties

## 5.30. Class GpUnit

Class for describing a geo-politically bounded area of geography such as a city, district, or jurisdiction, or a precinct or split-precinct.

Reporting units can link to each other to form a hierarchically-oriented model of a state's (or a county's, etc.) jurisdictions, districts, and precincts.

GpUnit is an abstract class with one subclass:

- ReportingUnit, for associating vote counts with geopolitical units such as cities, districts, counties, precincts, etc.

Election and Contest contain a required reference to GpUnit representing the jurisdiction of the election or contest respectively; Office contains a similar reference that is optional. BallotStyle references GpUnit to link a ballot style to its corresponding geopolitical unit.

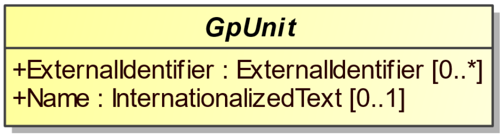

GpUnit
+ExternalIdentifier : ExternalIdentifier [0..*]
+Name : InternationalizedText [0..1]

Figure 46 – GpUnit

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { ComposingGpUnit } | 0..* | GpUnit | Unique identifier for one or more GpUnit instances. For creating a reference to another GpUnit that is contained with the parent GpUnit. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating an ID with the GpUnit, e.g., a district's or county's code. |
| Name | 0..1 | InternationalizedText | Name of the geopolitical unit. |

Table 29 - Describes GpUnit properties

## 5.31. Class ReportingUnit

For defining a geopolitical unit such as state, county, township, precinct, etc., using the ReportingUnitType enumeration. It inherits the attributes of GpUnit.

This class optionally references Person to associate one or more individuals, e.g., authorities, for the reporting unit. It also includes ContactInformation to provide contact addresses for the reporting unit, such as an address of a vote center.

Election references this class so as to identify the geographical scope of the election. In this case, the GpUnit element defined for the scope of the election may include ElectionAdministration to include election authority-related information.

The Type attribute uses the ReportingUnitType enumeration to specify the type of geopolitical geography being defined. ReportingUnitType contains the most common types of geographies, e.g., state, county, precinct, and so forth. If the reporting unit type is not listed in enumeration ReportingUnitType, use other and include the reporting unit type (that is not listed in the enumeration) in OtherType.

The IsDistricted boolean can be used in a number of ways. It is not strictly necessary, as it is possible to identify districts by their Type attribute or by examining the Contest instance's ElectionDistrict reference, which links to the election district associated with the contest. However, if a district is defined but is not linked from a contest, or if the type of district is not listed in the ReportingUnitType enumeration and therefore OtherType is used, then IsDistricted is necessary to identify the GpUnit as a district. The IsDistricted boolean can also be used to signify that a GpUnit defined as a jurisdiction, e.g., a county, is also used as a district for, e.g., county-wide contests.

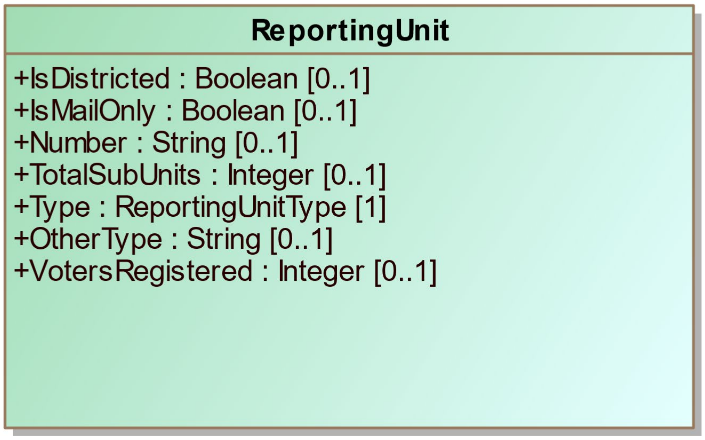

ReportingUnit
+IsDistricted : Boolean [0..1]
+IsMailOnly : Boolean [0..1]
+Number : String [0..1]
+TotalSubUnits : Integer [0..1]
+Type : ReportingUnitType [1]
+OtherType : String [0..1]
+VotersRegistered : Integer [0..1]

Figure 47 – ReportingUnit

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| {Authority} | 0..* | Person | A link to one or more Person instances describing an authority responsible for the reporting unit. |
| {ContactInformation} | 0..1 | ContactInformation | For associating contact information with the reporting unit. |
| {ElectionAdministration} | 0..1 | ElectionAdministration | For use when the reporting unit serves as the authority in the election. |
| IsDistricted | 0..1 | Boolean | Boolean to indicate whether the reporting unit is a district; assumed to be "false" if not present. |
| IsMailOnly | 0..1 | Boolean | Boolean to indicate whether the reporting unit handles only mail-in or absentee ballot elections, assumed to be "false" if not present. |
| Number | 0..1 | String | A number associated with the reporting unit; for compatibility with VIP. |
| {PartyRegistration} | 0..* | PartyRegistration | For associating a count of registered voters per party with the geopolitical unit. |
| {SpatialDimension} | 0..1 | SpatialDimension | For describing the reporting unit's spatial extent (a polygon that shows the related area). |
| TotalSubUnits | 0..1 | Integer | Total number of associated subunits such as precincts. |
| Type | 1 | ReportingUnitType | Enumerated type of reporting unit, e.g., state, county, district, precinct, etc. |
| OtherType | 0..1 | String | For use when ReportingUnitType value is other. |
| VotersRegistered | 0..1 | Integer | Number of registered voters residing within the boundaries of the geopolitical unit. |

Table 30 - Describes ReportingUnit properties

## 5.32. Class Header

For defining a reusable set of headers.

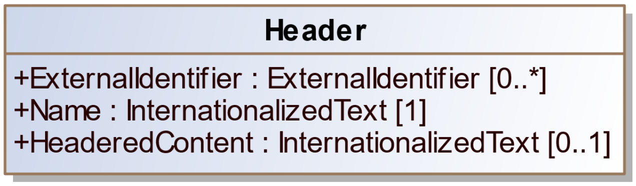

Header
+ExternalIdentifier : ExternalIdentifier [0..*]
+Name : InternationalizedText [1]
+HeaderedContent : InternationalizedText [0..1]

Figure 48 – Header

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating an ID with the header. |
| HeaderedContent | 0..1 | InternationalizedText | For textual content appearing under a header (e.g., ballot instructions). |
| Name | 1 | InternationalizedText | Name of the header, as it is to appear on a ballot style. |

Table 31 - Describes Header properties

## 5.33. Class Hours

Hours is used to specify a specific day and hours on that day, including the time zone. Multiple occurrences of Hours can be used if the schedule includes a range of days and hours.

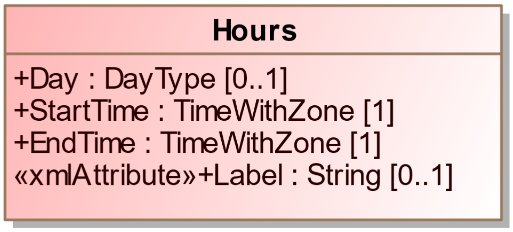

Hours
+Day : DayType [0..1]
+StartTime : TimeWithZone [1]
+EndTime : TimeWithZone [1]
«xmlAttribute»+Label : String [0..1]

Figure 49 – Hours

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Day | 0..1 | DayType | Day of the week. |
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| StartTime | 1 | TimeWithZone | Start time of the schedule. |
| EndTime | 1 | TimeWithZone | End time of the schedule. |

Table 32 - Describes Hours properties

## 5.34. Class InternationalizedText

For strings that can contain multi-national text, for use with text as shown on a ballot containing multi-national text. The label attribute can be used to assign an identifier to the text as desired.

Text uses the xsd:language type such that its language attribute must be set to a value that identifies the language.

Values for language are from ISO 639 [12] and include:

- en – English
- en-US – U.S. English
- en-GB – U.K. English
- fr – French
- es – Spanish
- zh – Chinese
- ja – Japanese
- ko – Korean

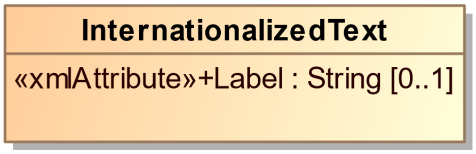


Figure 50 – InternationalizedText

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| { Text } | 1..* | LanguageString | Used to hold a string of text with an associated table indicating the language used. |

Table 33 - Describes InternationalizedText properties

## 5.35. Class LanguageString

Used to hold a string of text with an associated table indicating the language used.

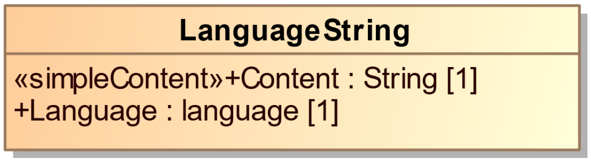


Figure 51 – LanguageString

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Content | 1 | String | The string in the specified language. |
| Language | 1 | language | Identification of the language, such as 'es'. |

Table 34 - Describes LanguageString properties

## 5.36. Class LatLng

Associates latitude/longitude with a contact address.

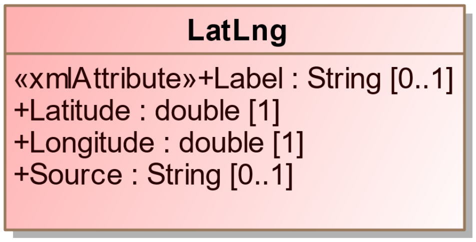


Figure 52 – LatLng

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| Latitude | 1 | double | Latitude of the contact location. |
| Longitude | 1 | double | Longitude of the contact location. |
| Source | 0..1 | String | System used to perform the lookup from location name to lat/lng, e.g., the name of a geocoding service. |

Table 35 - Describes LatLng properties

## 5.37. Class Office

For defining an office and information associated with a contest and/or a district. BallotDefinition includes Office. CandidateContest and RetentionContest reference Office.

Office includes Term for defining details about the term of an office such as start/end dates and the type of term. OfficeGroup is included from BallotDefinition to assign a name to a grouping of office definitions.

Office includes an optional ElectionDistrict reference to a GpUnit for the purpose of identifying the geographical scope of the office. For example, for an office for a state senate seat, ElectionDistrict would include a reference to the GpUnit defined for the district associated with that office.

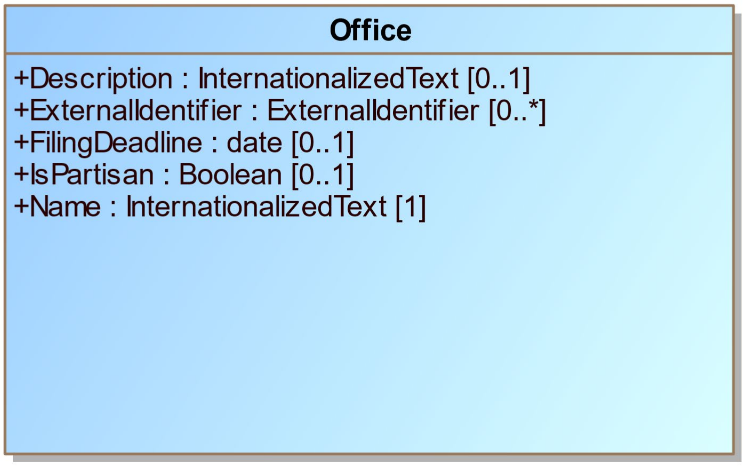

Office
+Description : InternationalizedText [0..1]
+ExternalIdentifier : ExternalIdentifier [0..*]
+FilingDeadline : date [0..1]
+IsPartisan : Boolean [0..1]
+Name : InternationalizedText [1]

Figure 53 – Office

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| {ContactInformation} | 0..1 | ContactInformation | For associating various contact information with the office. |
| Description | 0..1 | InternationalizedText | A description of the office, possibly as shown on the ballot to the voter. |
| {ElectionDistrict} | 0..1 | ReportingUnit | Link to a GpUnit instance. For associating the office with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating IDs with the office. |
| FilingDeadline | 0..1 | date | Date and time when a candidate must have filed for the contest for the office. |
| IsPartisan | 0..1 | Boolean | Boolean to indicate whether the office is partisan, e.g., true or false. If not present, assumption is true. |
| Name | 1 | InternationalizedText | Name of the office; can appear on the ballot. |
| {OfficeHolderPerson} | 0..* | Person | Links to one or more Person instances defined for the office holder. |
| {Term} | 0..1 | Term | For including office term-related information. |

Table 36 - Describes Office properties

## 5.38. Class OfficeGroup

Used to assign a name to a grouping of office definitions. It includes references to Office instances and a name to identify the grouping of references, e.g., “Judicial” or “Statewide”, etc. SubOfficeGroup can be used to create a nested hierarchy of groupings. BallotDefinition includes OfficeGroup.

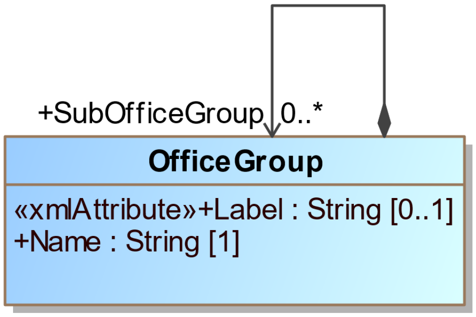


Figure 54 – OfficeGroup

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| Name | 1 | String | Name of the office grouping. |
| { Office } | 0..* | Office | Link to one or more Office instances. |
| { SubOfficeGroup } | 0..* | OfficeGroup | For defining a nested hierarchy of Office instance groupings. |

Table 37 - Describes OfficeGroup properties

## 5.39. Class OrderedContent

An abstract base class for content that can appear under a particular ballot style.

OrderedContent is an abstract class with two subclasses that get used according to the type of content:

- OrderedContest, used for the appearance of a contest.
- OrderedHeader, used for the appearance of a header, optionally with the inclusion of contests.

Figure 55 – OrderedContent

## 5.40. Class OrderedContest

For the appearance of a contest on a particular ballot style. This property uses OrderedContent as a superclass.

Figure 56 – OrderedContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Contest } | 1 | Contest | The contest associated represented by OrderedContest. |
| { OrderedContestOption } | 0..* | ContestOption | The contest selections for the ballot. |
| { Physical } | 0..* | PhysicalContest | For describing the physical aspects of the contest. |

Table 38 - Describes OrderedContest properties

## 5.41. Class OrderedHeader

For the appearance of a header on a particular ballot style. This property uses OrderedContent as a superclass.

Figure 57 – OrderedHeader

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Header } | 1 | Header | Association to the header to be used. |
| { OrderedContent } | 0..* | OrderedContent | For associating a header with ballot content, such as contests or nested headers. |

Table 39 - Describes OrderedHeader properties

## 5.42. Class Party

Used to describe a political party that can then be referenced in other elements. BallotDefinition includes Party. Candidate, PartyRegistration, and Person reference Party.

Party is an abstract type with one subtype Coalition, used to define coalitions.

The Color attribute specifies a 6-digit RGB code displayable using HTML.

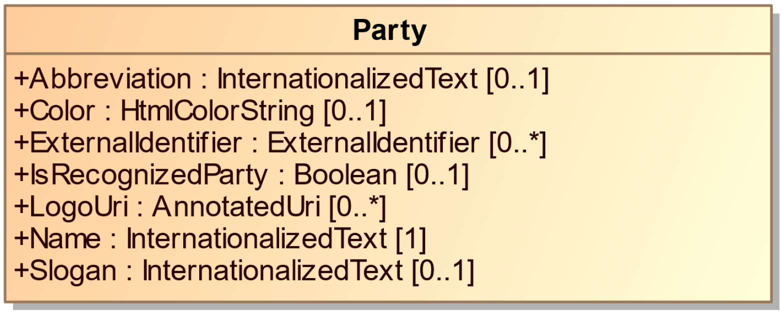

Party
+Abbreviation : InternationalizedText [0..1]
+Color : HtmlColorString [0..1]
+ExternalIdentifier : ExternalIdentifier [0..*]
+IsRecognizedParty : Boolean [0..1]
+LogoUri : AnnotatedUri [0..*]
+Name : InternationalizedText [1]
+Slogan : InternationalizedText [0..1]

Figure 58 – Party

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Abbreviation | 0..1 | InternationalizedText | Short name for the party, e.g., “DEM”. |
| Color | 0..1 | HtmlColorString | For associating an HTML RGB color coding with the party. |
| { ContactInformation } | 0..1 | ContactInformation | For associating contact information regarding the party, e.g., party offices. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating IDs with the party. |
| IsRe cognizedParty | 0..1 | Boolean | For indicating whether the party is recognized by the election authority; “false” is assumed if not present. |
| { LeaderPerson } | 0..* | Person | Identification of a Party's leader. |
| LogoUri | 0..* | AnnotatedUri | A URI to the party’s graphical logo. |
| Name | 1 | InternationalizedText | Official full name of the party, e.g., “Republican”; can appear on the ballot. |
| { PartyScopeGpUnit } | 0..* | GpUnit | The GpUnit(s) the party operates in or the top-most GpUnit. |
| Slogan | 0..1 | InternationalizedText | The slogan or motto used by a political party. |

Table 40 - Describes Party properties

## 5.43. Class Coalition

For defining a coalition, i.e., a collection of parties organized for the purpose of endorsing a candidate in a contest. It inherits the properties of Party.

Coalition instances themselves are composed of multiple Party references along with a reference to an associated Class Contest.

If there are no associated Contests, a general default is that the coalition endorses the associated parties.

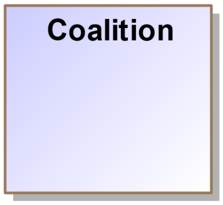


Figure 59 – Coalition

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Contest } | 0..* | Contest | For associating contests with the coalition. |
| { Party } | 0..* | Party | For associating parties with the coalition. |

Table 41 - Describes Coalition properties

## 5.44. Class PartyRegistration

For tracking the number of registered voters per party per geopolitical unit, i.e., for reporting on the number of registered voters of a particular party in a district or other type of reporting unit. Referenced by GpUnit.

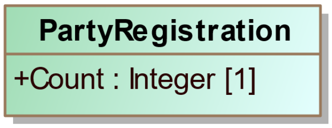


Figure 60 – PartyRegistration

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Count | 1 | Integer | A count for tracking the number of registered voters. |
| { Party } | 1 | Party | Link to a Party instance. For associating a political party. |

Table 42 - Describes PartyRegistration properties

## 5.45. Class Person

For defining information about a person; the person may be a candidate, election official, authority for a reporting unit, etc. BallotDefinition includes Person. Candidate, ElectionAdministration and GpUnit references Person. Person optionally references ContactInformation for associating contact information.

Multiple occurrences of the MiddleName attribute can be used as needed, e.g., for names such as “John Andrew Winston Smith”.

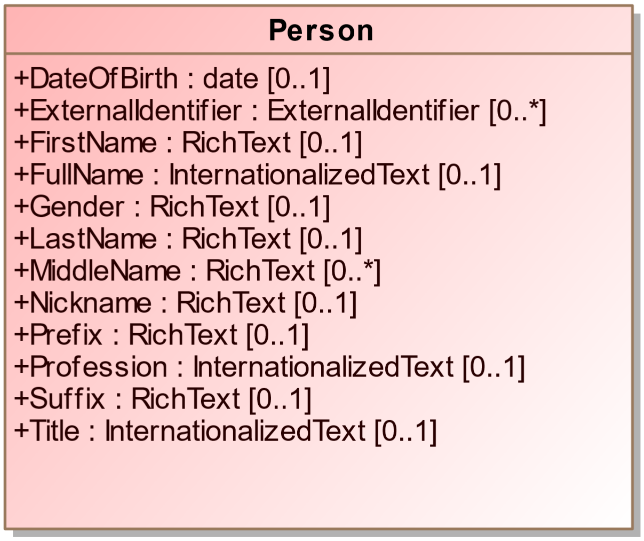

Person
+DateOfBirth : date [0..1]
+ExternalIdentifier : ExternalIdentifier [0..*]
+FirstName : RichText [0..1]
+FullName : InternationalizedText [0..1]
+Gender : RichText [0..1]
+LastName : RichText [0..1]
+MiddleName : RichText [0..*]
+Nickname : RichText [0..1]
+Prefix : RichText [0..1]
+Profession : InternationalizedText [0..1]
+Suffix : RichText [0..1]
+Title : InternationalizedText [0..1]

Figure 61 – Person

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| {ContactInformation} | 0..* | ContactInformation | For associating contact information with the person. |
| DateOfBirth | 0..1 | date | Person's date of birth. |
| ExternalIdentifier | 0..* | ExternalIdentifier | For associating IDs with the person. |
| FirstName | 0..1 | RichText | Person's first (given) name. |
| FullName | 0..1 | InternationalizedText | Person's full name. |
| Gender | 0..1 | RichText | Person's gender. |
| LastName | 0..1 | RichText | Person's last (family) name. |
| MiddleName | 0..* | RichText | Person's middle name. |
| Nickname | 0..1 | RichText | Nickname associated with the person. |
| {Party} | 0..1 | Party | Links to a Party instance. For associating a political party with the person. |
| Prefix | 0..1 | RichText | A prefix associated with the person, e.g., Mr. |
| Profession | 0..1 | InternationalizedText | Person's profession. |
| Suffix | 0..1 | RichText | A suffix associated with the person, e.g., Jr. |
| Title | 0..1 | InternationalizedText | A title associated with the person. |

Table 43 - Describes Person properties

## 5.46. Class PhysicalContest

For the appearance of a contest on a particular ballot style with physical details such as the locations and shapes of contest option positions.

Figure 62 – PhysicalContest

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { BallotFormat } | 1 | BallotFormat | For associating a ballot format with a physical contest. |
| { Extent } | 0..* | BoundedObject | The extent of the contest. If the contest spans multiple sections of the ballot (e.g., columns or pages), then multiple extents may be provided. |
| { FiducialMark } | 0..* | FiducialMark | For associating a ordered physical contest with one or more local fiducial marks appearing near the contest. |
| { PhysicalContestOption } | 0..* | PhysicalContestOption | The contest options associated with the contest, including physical details. |

Table 44 - Describes PhysicalContest properties

## 5.47. Class PhysicalContestOption

For associating a ContestOption to its physical manifestation. Each PhysicalContestOption is specified in terms of a particular BallotFormat. A PhysicalContestOption may have multiple OptionPosition instances, for voting methods that require it (e.g., rank choice voting). WriteInPosition specifies the response area where name(s) may written-in.

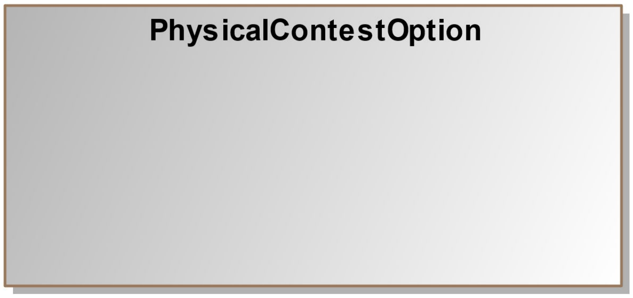


Figure 63 – PhysicalContestOption

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { ContestOption } | 0..1 | ContestOption | For associating a PhysicalContestOption with a ContestOption. This should always be provided unless the contest option is for a write-in. |
| { OptionPosition } | 1..* | OptionPosition | For defining locations where a selection can be indicated. |
| { WriteInPosition } | 0..* | WriteInPosit ion | For defining locations where a name can be hand-written. |

Table 45 - Describes PhysicalContestOption properties

## 5.48. Class Schedule

For defining a schedule associated with a particular election office or location. ContactInformation includes Schedule.

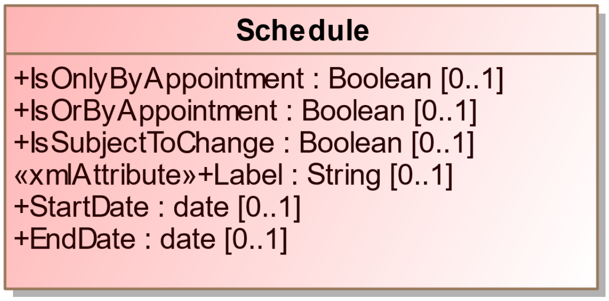

Schedule
+IsOnlyByAppointment : Boolean [0..1]
+IsOrByAppointment : Boolean [0..1]
+IsSubjectToChange : Boolean [0..1]
«xmlAttribute»+Label : String [0..1]
+StartDate : date [0..1]
+EndDate : date [0..1]

Figure 64 – Schedule

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| { Hours } | 0..* | Hours | For specifying a range of hours for a schedule. |
| IsOnlyByAppointment | 0..1 | Boolean | If an appointment is only by appointment; assumed to be “no” if not present. |
| IsOrByAppointment | 0..1 | Boolean | If an appointment can by appointment presumably as desired; assumed to be “no” if not present. |
| IsSubjectToChange | 0..1 | Boolean | If an appointment may be subject to change; assumed to be “no” if not present. |
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| StartDate | 0..1 | date | For the starting date of the schedule. |
| EndDate | 0..1 | date | For the ending date of the schedule. |

Table 46 - Describes Schedule properties

## 5.49. Class Shape

For defining a geometric shape. Geometry can be used to represent content on a physical ballot style, such as for associating an OptionPosition to its shape. It is also used by FiducialMark to define the shape it takes.

Strokes when applied, are applied evenhanded.

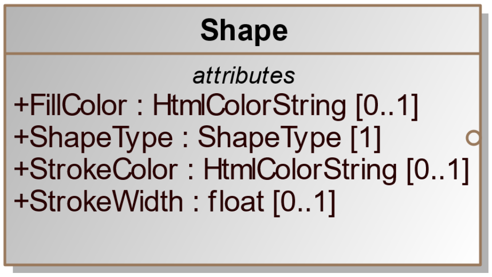

Shape
attributes
+FillColor : HtmlColorString [0..1]
+ShapeType : ShapeType [1]
+StrokeColor : HtmlColorString [0..1]
+StrokeWidth : float [0..1]

Figure 65 – Shape

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| FillColor | 0..1 | HtmlColorString | For fiducial marks only. If not specified, no fill is assumed. |
| ShapeType | 1 | ShapeType | The shape this Geometry represents. |
| StrokeColor | 0..1 | HtmlColorString | The color of the stroke. If no color is specified, assume black (what XFA does) |
| StrokeWidth | 0..1 | float | The width of the stroke in the measurement units of the BallotFormat associated with the PhysicalContest. |

Table 47 - Describes Shape properties

## 5.50. Class SpatialDimension

For defining the spatial layout of a GpUnit, e.g., a map or a spatial extent (a polygon that shows the related area) for various purposes, including to visualize election results, to understand the composition of districts, or to determine whether GpUnit instances are properly related. GpUnit includes SpatialDimension.

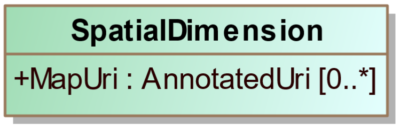


Figure 66 – SpatialDimension

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| MapUri | 0..* | AnnotatedUri | Typically a URI to a map of the GpUnit. |
| { SpatialExtent } | 0..1 | SpatialExtent | For associating a GpUnit element's spatial extent information. |

Table 48 - Describes SpatialDimension properties

## 5.51. Class SpatialExtent

SpatialDimension includes SpatialExtent for defining a GpUnit instance’s spatial extent data and the format used for the spatial extent.

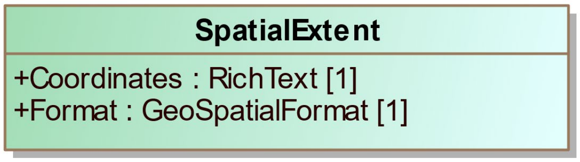


Figure 67 – SpatialExtent

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Coordinates | 1 | RichText | The data coordinates constituting the spatial extent. |
| Format | 1 | GeoSpatialFormat | Enumerated type for the format used, e.g., gml, kml, wkt, etc. |

Table 49 - Describes SpatialExtent properties

## 5.52. Class Term

For describing information about an office term. Term is included by Office.

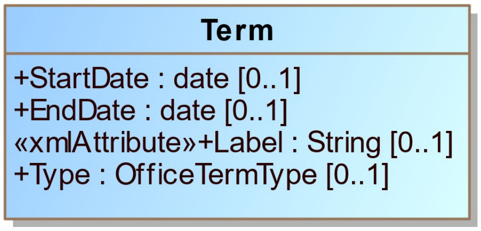

Term
+StartDate : date [0..1]
+EndDate : date [0..1]
«xmlAttribute»+Label : String [0..1]
+Type : OfficeTermType [0..1]

Figure 68 – Term

| Property | Multiplicity | Type | Property Description |
|---|---|---|---|
| Label | 0..1 | String | For use as needed and compatibility with the VIP schema. |
| StartDate | 0..1 | date | Start date for the current term of the office. |
| EndDate | 0..1 | date | End date for the current term of the office. |
| Type | 0..1 | OfficeTermType | Enumerated type of term, e.g., full-term, unexpired-term, etc. |

Table 50 - Describes Term properties

## 5.53. Enumeration BallotDefinitionVersion

«enumeration»
BallotDefinitionVersion
1.0.0

Figure 69 – BallotDefinitionVersion

| Value | Value Description |
|---|---|
| 1.0.0 | For version 1.0.0 of this specification. |

Table 51 - Describes BallotDefinitionVersion properties

## 5.54. Enumeration BallotMeasureType

Enumeration for types of ballot measures in the BallotMeasureContest element.

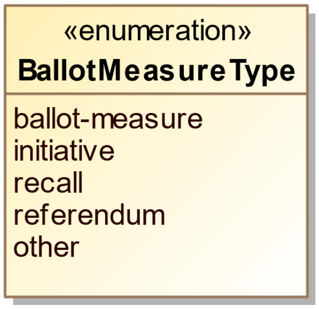


Figure 70 – BallotMeasureType

| Value | Value Description |
|---|---|
| ballot - measure | For a standard "yes" or "no" question on the ballot. |
| initiative | For an initiative. |
| recall | For a recall. |
| referendum | For a referendum. |
| other | Used when the type of ballot measure is not included in this enumeration. |

Table 52 - Describes BallotMeasureType properties

## 5.55. Enumeration BallotSideType

The side of a ballot sheet.

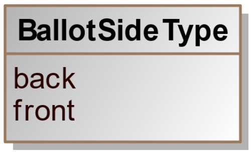


Figure 71 – BallotSideType

| Value | Value Description |
|---|---|
| back | The back of the ballot sheet. |
| front | The front of the ballot sheet |

Table 53 - Describes BallotSideType properties

## 5.56. Enumeration CandidatePreElectionStatus

Enumeration for various pre-election statuses applicable to a candidate in the Candidate class.

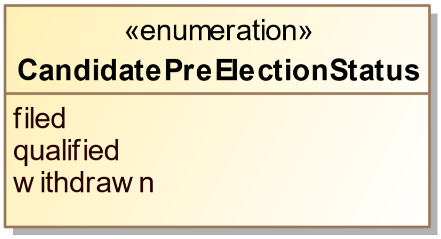


Figure 72 – CandidatePreElectionStatus

| Value | Value Description |
|---|---|
| filed | For candidates who have filed with the election authority but not necessarily qualified. |
| qualified | For candidates who are qualified by the election authority to be on the ballot for a contest. |
| withdrawn | For candidates who have withdrawn from the contest. |

Table 54 - Describes CandidatePreElectionStatus properties

## 5.57. Enumeration DayType

Enumeration for the day(s) in a schedule in the Schedule element.

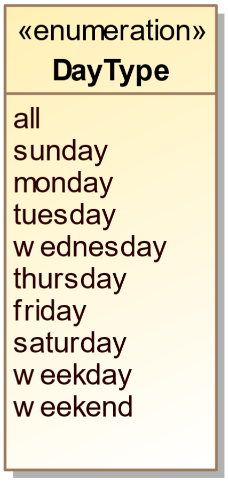


Figure 73 – DayType

| Value | Value Description |
|---|---|
| all | Used for all days of the week. |
| sunday | Used if day of week is Sunday. |
| monday | Used if day of week is Monday. |
| tuesday | Used if day of week is Tuesday. |
| wednesday | Used if day of week is Wednesday. |
| thursday | Used if day of week is Thursday. |
| friday | Used if day of week is Friday. |
| saturday | Used if day of week is Saturday. |
| weekday | Used for any day of the week except Saturday and Sunday. |
| weekend | Used for both Saturday and Sunday. |

Table 55 - Describes DayType properties

## 5.58. Enumeration ElectionType

Enumeration for the type of election in the BallotDefinition class.

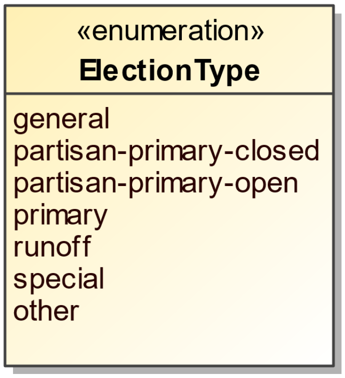


Figure 74 – ElectionType

| Value | Value Description |
|---|---|
| general | Election in which all eligible voters, regardless of party affiliation, are permitted to select candidates to fill public office and/or vote on ballot measures. |
| partisan - primary - closed | Primary election in which the voter receives a ballot containing only those party-specific contests pertaining to the political party with which the voter is affiliated, along with non-party-specific contests presented at the same election. Unaffiliated voters may be permitted to vote only on non-party-specific contests. |
| partisan - primary - open | Primary election in which the voter may choose a political party at the time of voting and vote in party-specific contests associated with that party, along with non-party-specific contests presented at the same election. Some states require voters to publicly declare their choice of party at the polling place, after which the election worker provides or activates the appropriate ballot. Other states allow the voters to make their choice of party within the privacy of the voting booth. |
| primary | Election held to determine which candidates qualify to appear as contest options in subsequent elections. |
| runoff | Election to select a winner following a primary or a general election, in which no candidate in the contest received the required minimum percentage of the votes cast. The two candidates receiving the most votes for the contest in question proceed to a runoff election. |
| special | Primary or general election that is not regularly scheduled. A special election may be combined with a scheduled election. |
| other | Used when the election type is not listed in this enumeration. |

Table 56 - Describes ElectionType properties

## 5.59. Enumeration GeoSpatialFormat

Enumeration for geospatial vector data formats used in Geographic Information System (GIS) software, used in the SpatialExtent class.

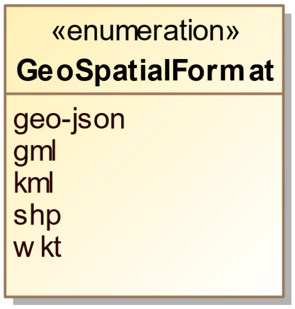


Figure 75 – GeoSpatialFormat

| Value | Value Description |
|---|---|
| geo-json | For GeoJSON open standard format. |
| gml | For Geography Markup Language format. |
| kml | For Keyhole Markup Language format. |
| shp | For the shape file format associated with Esri. |
| wkt | For Well-known Text format. |

Table 57 - Describes GeoSpatialFormat properties

## 5.60. Enumeration IdentifierType

Enumeration for election data-related codes in the ExternalIdentifier class .

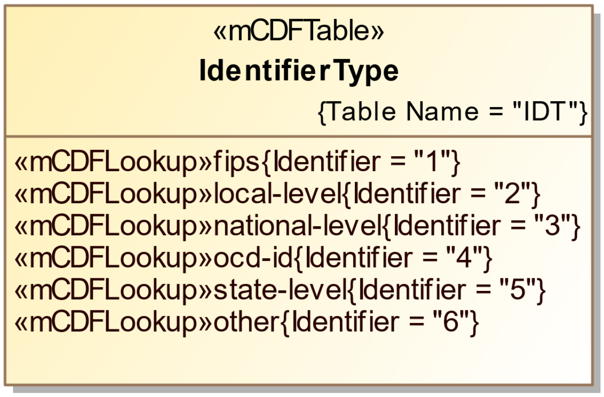

«mCDFTable»
IdentifierType
{Table Name = "IDT"}
«mCDFLookup»fips{Identifier = "1"}
«mCDFLookup»local-level{Identifier = "2"}
«mCDFLookup»national-level{Identifier = "3"}
«mCDFLookup»ocd-id{Identifier = "4"}
«mCDFLookup»state-level{Identifier = "5"}
«mCDFLookup»other{Identifier = "6"}

Figure 76 – IdentifierType

| Value | Value Description |
|---|---|
| fips | For FIPS codes. |
| local - level | For a code that is specific to a county or other similar locality. |
| national - level | For a code that is used at the national level other than “fips” or “ocd-id”. |
| ocd - id | For Open Civic Data identifiers. |
| state - level | For a code that is specific to a state. |
| other | Used when the type of code is not included in this enumeration. |

Table 58 - Describes IdentifierType properties

### 5.61. Enumeration MeasurementUnitType

The measurement unit for describing the location of content on a ballot.

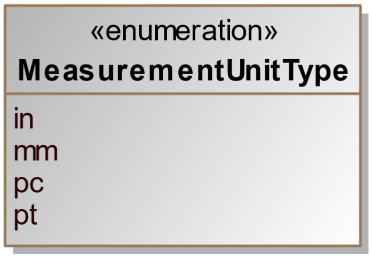


Figure 77 – MeasurementUnitType

| Value | Value Description |
|---|---|
| in | inches |
| mm | millimeters |
| pc | picas |
| pt | points (1/72 inch) |

Table 59 - Describes MeasurementUnitType properties

## 5.62. Enumeration OfficeTermType

Enumeration for the office term type in the Office class.

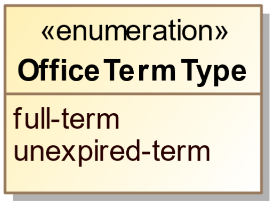


Figure 78 – OfficeTermType

| Value | Value Description |
|---|---|
| full - term | When the officeholder's term began at the beginning of the full term of the office, e.g., 6 years for U.S. Senate. |
| unexpired - term | When the officeholder's term began at some date after the beginning of the full term of the office, generally because the previous officeholder vacated the office before the full term expired. |

Table 60 - Describes OfficeTermType properties

## 5.63. Enumeration OrientationType

The orientation of a ballot sheet.

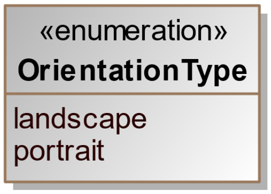


Figure 79 – OrientationType

| Value | Value Description |
|---|---|
| landscape | Orient the sheet so that the ballot content print across the long edge of the paper. |
| portrait | Orient the sheet so that the ballot content print across the short edge of the paper. |

Table 61 - Describes OrientationType properties

## 5.64. Enumeration ReportingUnitType

Enumeration for the type of geopolitical unit in the ReportingUnit class.

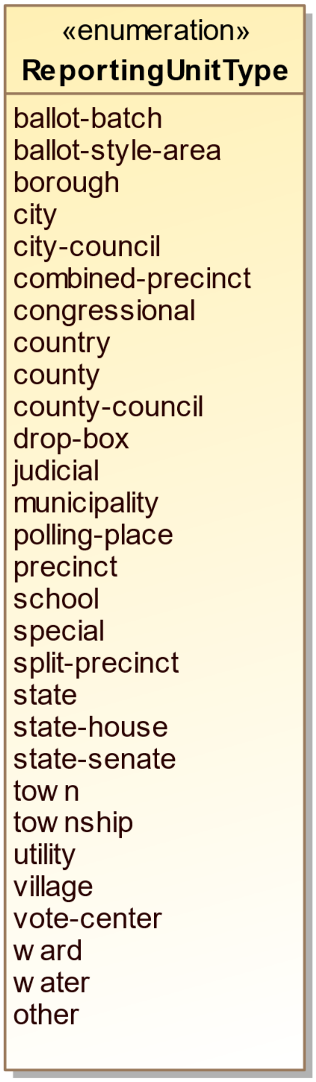


Figure 80 – ReportingUnitType

| Value | Value Description |
|---|---|
| ballot - batch | Used for reporting batches of ballots that may cross precinct boundaries. |
| ballot - style - area | Used for ballot style areas generally composed of precincts. |
| borough | Used in CT, NJ, PA, other states, and New York City for boroughs. For AK and LA, see county. |
| city | Used for a city that reports results and/or for the district that encompasses it. |
| city - council | Used for city council districts. |

| Value | Value Description |
|---|---|
| combined-precinct | Used for one or more precincts that have been combined for the purposes of reporting. Used for “Ward” if “Ward” is used interchangeably with “CombinedPrecinct”. |
| congressional | Used for U.S. Congressional districts. |
| country | Used for a reporting unit of type country. |
| county | Used for a county and/or for the district that encompasses it. In AK, used for counties that are called boroughs. In LA, used for parishes. |
| county-council | Used for county council districts. |
| drop-box | Used for a drop box for absentee ballots. |
| judicial | Used for judicial districts. |
| municipality | Used as applicable for various units such as towns, townships, villages that report votes and/or for the district that encompasses it. |
| polling-place | Used for a polling place. |
| precinct | Used also for “Ward” or “District” when these terms are used interchangeably with “Precinct”. |
| school | Used for a school district. |
| special | Used for a special district. |
| split-precinct | Used for splits of precincts. |
| state | Used for a state and/or for the district that encompasses it. |
| state-house | Used for a state house or assembly district. |
| state-senate | Used for a state senate district. |
| town | Used in some New England states as a type of municipality that reports votes and/or for the district that encompasses it. |
| township | Used in some mid-western states as a type of municipality that reports votes and/or for the district that encompasses it. |
| utility | Used for a utility district. |
| village | Used as a type of municipality that reports votes and/or for the district that encompasses it. |
| vote-center | Used for a vote center. |
| ward | Used for combinations or groupings of precincts or other units. |
| water | Used for a water district. |
| other | Used for other types of reporting units not included in this enumeration. |

Table 62 - Describes ReportingUnitType properties

## 5.65. Enumeration SelectionCaptureMethod

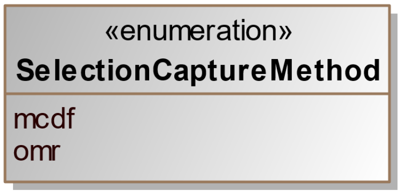


Figure 81 – SelectionCaptureMethod

| Value | Value Description |
|---|---|
| mcdf | For a read method that uses the mCDF encoded using a symbology. |
| omr | For a read method that uses optical mark recognition. |

Table 63 - Describes SelectionCaptureContent properties

## 5.66. Enumeration ShapeType

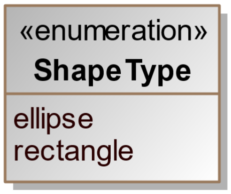


Figure 82 – ShapeType

| Value | Value Description |
|---|---|
| ellipse | For an ellipse shape |
| rectangle | For a rectangle shape |

Table 64 - Describes ShapeType properties

## 5.67. Enumeration StraightPartyRuleset

Specifies how to consider indications made in the controlled contests of a straight party controlling contest. These specify the consequences direct selections made by the voter in controlled contests.

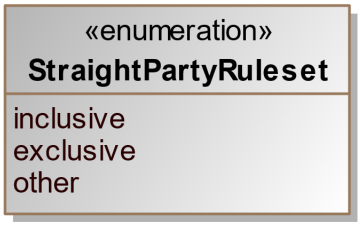


Figure 83 – StraightPartyRuleset

| Value | Value Description |
|---|---|
| inclusive | For a ruleset allowing 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 | For a ruleset that excludes any straight-party selections if there is a directly selected candidate in a contest. |
| other | Used when the straight party rule type is not included in this enumeration. |

Table 65 - Describes StraightPartyRuleset properties

## 5.68. Enumeration VoteVariation

Enumeration for contest decision algorithm or rules in the Contest element.

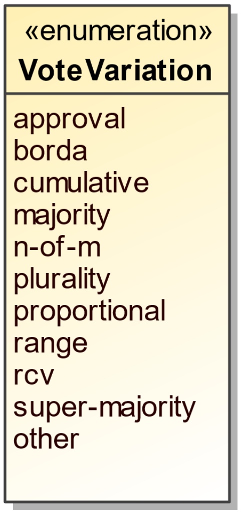


Figure 84 – VoteVariation

| Value | Value Description |
|---|---|
| approval | When voter can select as many candidates as desired in a contest up to a maximum number. |
| borda | For the Borda count voting. |
| cumulative | When voter can allocate more than one vote to a given candidate. |
| majority | For majority voting. |
| n- of - m | Includes vote for 1, i.e., 1-of-m. |
| plurality | For plurality voting. |
| proportional | For proportional voting. |
| range | For range voting. |
| rcv | For ranked choice voting. |
| super - majority | For super majority voting. |
| other | Used when the vote variation type is not included in this enumeration. |

Table 66 - Describes VoteVariation properties

# 6. Usage Examples

This section presents an overview of various aspects and usage of the Ballot Definition schema, including:

1. How geopolitical geography is implemented,
2. How contests and contest options are implemented,
3. How controlling contest are implemented,
4. How to physical ballot styles are implemented, and
5. How to define headers in ballot styles.

No attempt is made to provide a complete overview of usage, but more to describe in general how most elements are to be used.

## 6.1. Defining Geopolitical Geography

A primary feature of the schemas is the geopolitical geography element *<GpUnit >*, which can be used in a variety of ways to exactly mirror a jurisdiction's geopolitical geography. The UML model and JSON and XML schemas permit counts to be reported at a high aggregation level only if desired, thus *<GpUnit >* elements are required in two places:

1. The *<Contest>* element contains a required link to a *<GpUnit>* defined for the election district of the contest, so that vote counts can be linked to that *<GpUnit>* element. The *<Election DistrictId>* sub-element is used to link to the *<GpUnit>* element defined for the election district.
2. The *<Election>* element contains a required link to a *<GpUnit>* defined for the geographical scope of the election, again using the *<ElectionScopeId>* sub-element.

If counts need to be associated with the smaller geographies, i.e., precincts, additional *<GpUnit>* elements need to be defined for these geographies. *<GpUnit >* elements, as described in section can be used for the following major types of geographies:

- Geographies that report election results or that can have vote counts associated with them, including for states, counties, cities, towns, precincts, split precincts, ballot style areas, etc.;
- Geographies that serve as election districts, i.e., that have contests associated with them;
- Geographies that serve as districts that have no contests associated (e.g., tax districts); and
- Geographies that are for specific locations or specific objects including:
    - Voting devices;
    - Polling places or vote centers;
    - Drop boxes; and
    - Other miscellaneous objects.

Additionally, *<GpUnit >* elements can be defined for other types of objects not listed in the *<ReportingUnitType >* enumeration by using *other* in the *<Type>* sub-element and then listing the other type of object in the *<OtherType >* sub-element.

### 6.1.1. Defining <GpUnit> Elements for Governmental and Administrative geographies

Figure 85[1] shows an example of geopolitical geography in the town of Richmond, WI. Figure 86 shows the *<GpUnit >* definitions for the governmental and administrative geographies (in Wisconsin, a ward is the same as a precinct). The assumption here is that the election results will be reported by these geographies (county, town, and ward), thus accordingly there are *<GpUnit>* definitions for these geographies. The county needs to be linked with the state, the town needs to be linked with the county, the combined wards need to be linked with the town, the five precincts need to be linked with their parent combined precincts, and the two split precinct elements need to be linked with their parent Ward 1 precinct. Thus, the *<GpUnit>* definitions for the parent geographies include *<ComposingGpUnitId s>* to reference the geographies that are contained within the parent elements.

[1] This figure is repeated from Figure 7.

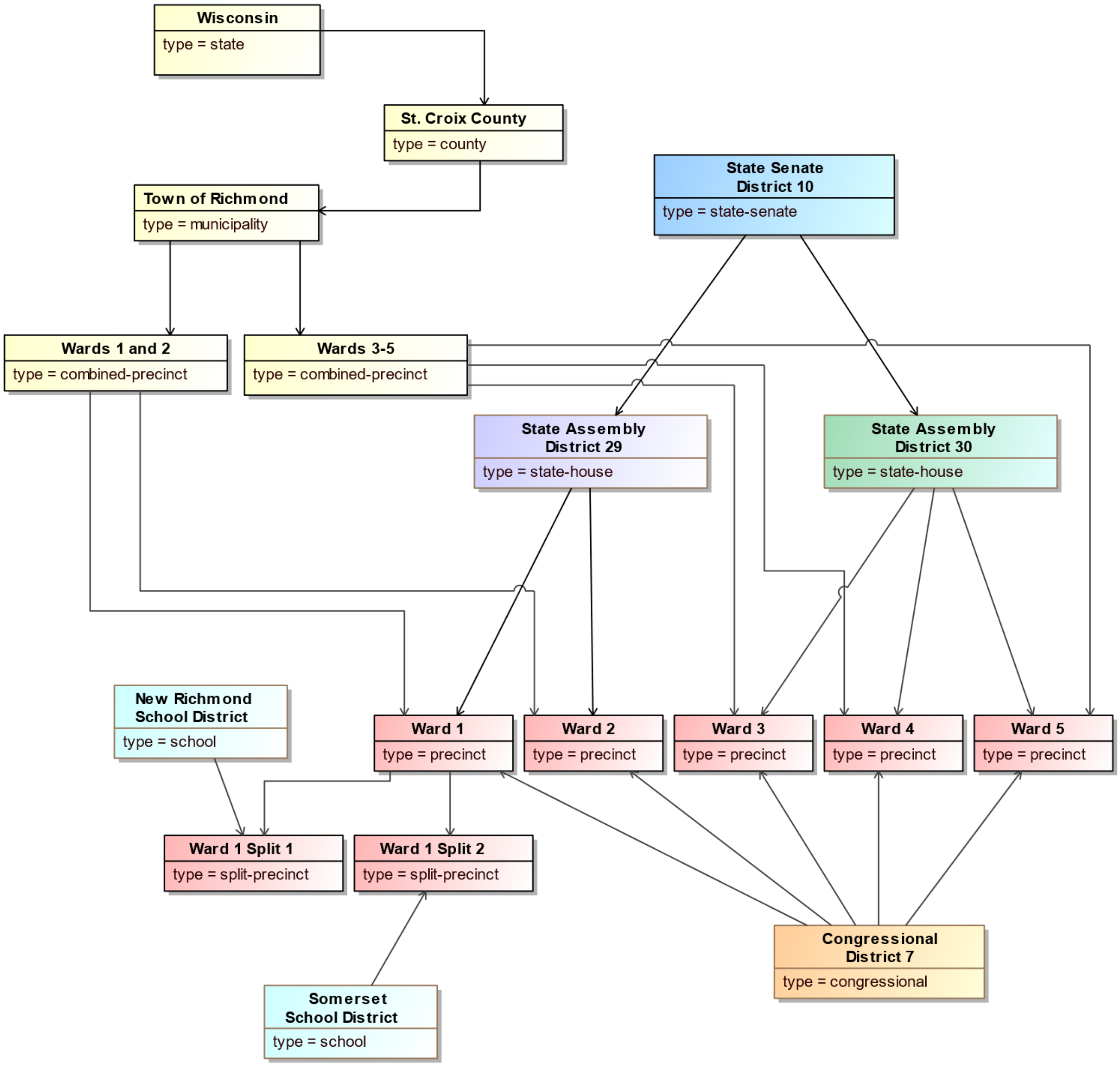


Figure 85 – Geopolitical Geography Example from Wisconsin

However, there is a rule that **must** be observed when defining these relationships:

> *A <GpUnit> element's <ComposingGpUnitIds> element must only reference those geopolitical units that are wholly contained within the scope of the <GpUnit> element.*

```
<! --    Def initions for the state, county, and city                  --  >
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID                 - State”>
        <Name>
            <Text language=”en”>State of Wisconsin</Text>
        </Name>
        <ComposingGpUnitIds>OID    - ST- CROIX</ComposingGpUnitIds>
        <Type >state</Type>
```

```
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-ST-CROIX”>
   <Name>
      <Text language=”en”>St. Croix County</Text>
   </Name>
   <ComposingGpUnitIds>OID-Richmond</ComposingGpUnitIds>
   <Type>county</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-Richmond”>
   <Name>
      <Text language=”en”>Town of Richmond</Text>
   </Name>
   <ComposingGpUnitIds>OID-WARDS1-2 OID-WARDS3-5</ComposingGpUnitIds>
   <Type>municipality</Type>
</GpUnit>
<!--Definitions for combined wards -->
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARDS1-2”>
   <Name>
      <Text language=”en”>Combined Wards 1 and 2</Text>
   </Name>
   <ComposingGpUnitIds>OID-WARD1 OID-WARD2</ComposingGpUnitIds>
   <Type>combined-precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARDS3-5”>
   <Name>
      <Text language=”en”>Combined Wards 3-5</Text>
   </Name>
   <ComposingGpUnitIds>OID-WARD3 OID-WARD4 OID-WARD5</ComposingGpUnitIds>
   <Type>combined-precinct</Type>
</GpUnit>
<!--Definitions for precincts and split precincts -->
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD1”>
   <Name>
      <Text language=”en”>Ward 1</Text>
   </Name>
   <ComposingGpUnitIds>OID-WARD1-1 OID-WARD1-2</ComposingGpUnitIds>
   <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD1-1”>
   <Name>
      <Text language=”en”>Ward 1-1</Text>
   </Name>
   <Type>split-precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD1-2”>
   <Name>
      <Text language=”en”>Ward 1-2</Text>
   </Name>
   <Type>split-precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD2”>
   <Name>
      <Text language=”en”>Ward 2</Text>
   </Name>
   <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD3”>
```

```
   <Name>
      <Text language=”en”>Ward 3</Text>
   </Name>
   <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD4”>
   <Name>
      <Text language=”en”>Ward 4</Text>
   </Name>
   <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-WARD5”>
   <Name>
      <Text language=”en”>Ward 5</Text>
   </Name>
   <Type>precinct</Type>
</GpUnit>
```

Figure 86 - <GpUnit> Definitions for Governmental and Administrative Geographies

Accordingly, split precincts are wholly contained within precincts, thus the proper way to structure the associated *<GpUnit>* elements is for the precinct element to contain a *<ComposingGpUnitIds>* element that references the split precinct elements – and not vice versa. If this rule is violated, an application would not be able to find information correctly within the instance file. Figure 87 shows a hierarchy of containment relationships, with precincts and splits being the foundational bricks for the geographies.

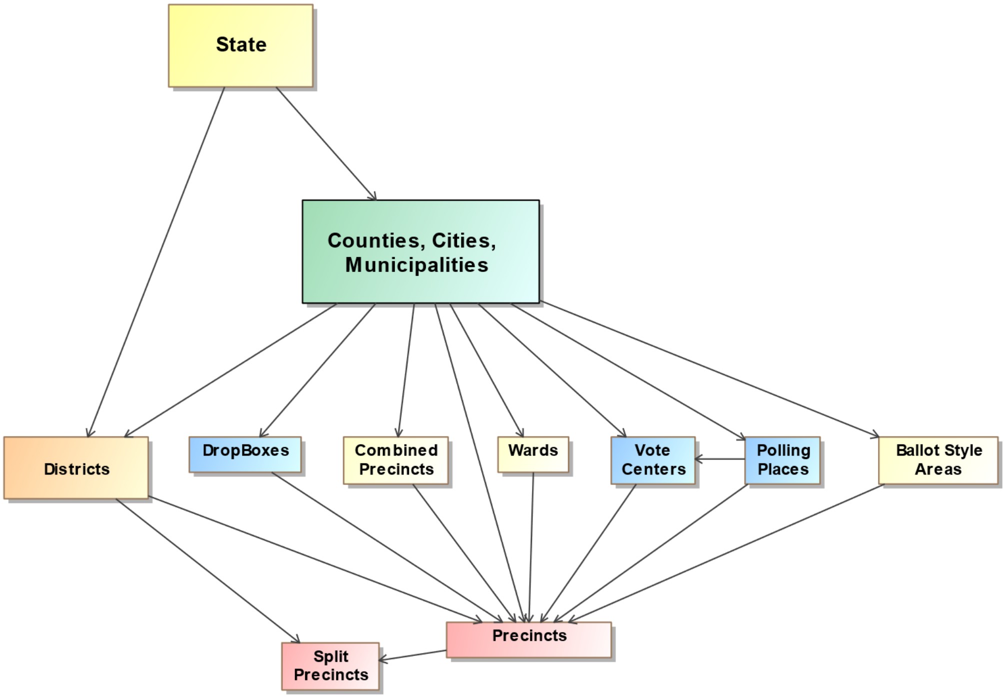


Figure 87 – Containment Relationships

### 6.1.2. Associating an Election Authority with the <GpUnit> Element Defined for the Scope of the Election

In Figure 85, the State of Wisconsin was assumed as the scope of the election, and in Figure 86, a *<GpUnit>* element was defined for the state. Figure 88Figure 88 shows the definition for the state, but this time with the election authority information added.

```
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-State">
    <ElectionAdministration>
        <ContactInformation>
            <Name>Wisconsin Election Authority</Name>
            <Uri>https://wisconsin-demo-election-url.wisconsin.gov</Uri>
        </ContactInformation>
        <ElectionOfficialPersonIds>OID</ElectionOfficialPersonIds>
        <Name>State of Wisconsin Demo Election</Name>
    </ElectionAdministration>
    <ComposingGpUnitIds>OID-ST-CROIX</ComposingGpUnitIds>
    <Name>
        <Text language="en">State of Wisconsin</Text>
    </Name>
    <Type>state</Type>
</GpUnit>
```

Figure 88 – Associating Election Authority Information

The *<Election>* element references a *<GpUnit>* element defined for the scope of the election, and that *<GpUnit>* element can then include the *<ElectionAdministration>* element for associating election authority information. *<ElectionAdministration>* includes *<ContactInformation>* for the election authority and, using *<ElectionOfficialPersonIds>*, references one or more *<Person>* elements defined for individuals/organizations associated with the election authority.

### 6.1.3. Defining <GpUnit> Elements for Political geographies

Figure 89 shows the districts and the wards that compose them from Figure 90. Figure 89 shows the *<GpUnit>* definitions for the districts as well as the use of *<ComposingGpUnitIds>* elements to link the districts to the wards that compose them.

The *<IsDistricted>* boolean is used in this example. It is not strictly necessary, as it is possible to identify districts by their type or by examining the *<Contest>* element's *<ElectionDistrictId>* sub-element, which links to the election district associated with the contest. However, if a district is defined but is not linked from a contest, or if the type of district is not listed in the *<ReportingUnitType>* enumeration and therefore *<OtherType>* is used, then *<IsDistricted>* is necessary to identify the *<GpUnit>* as a district. The *<IsDistricted>* boolean can also be used to signify that a *<GpUnit>* defined as a jurisdiction, e.g., a county, is also used as a district for, e.g., county-wide contests.

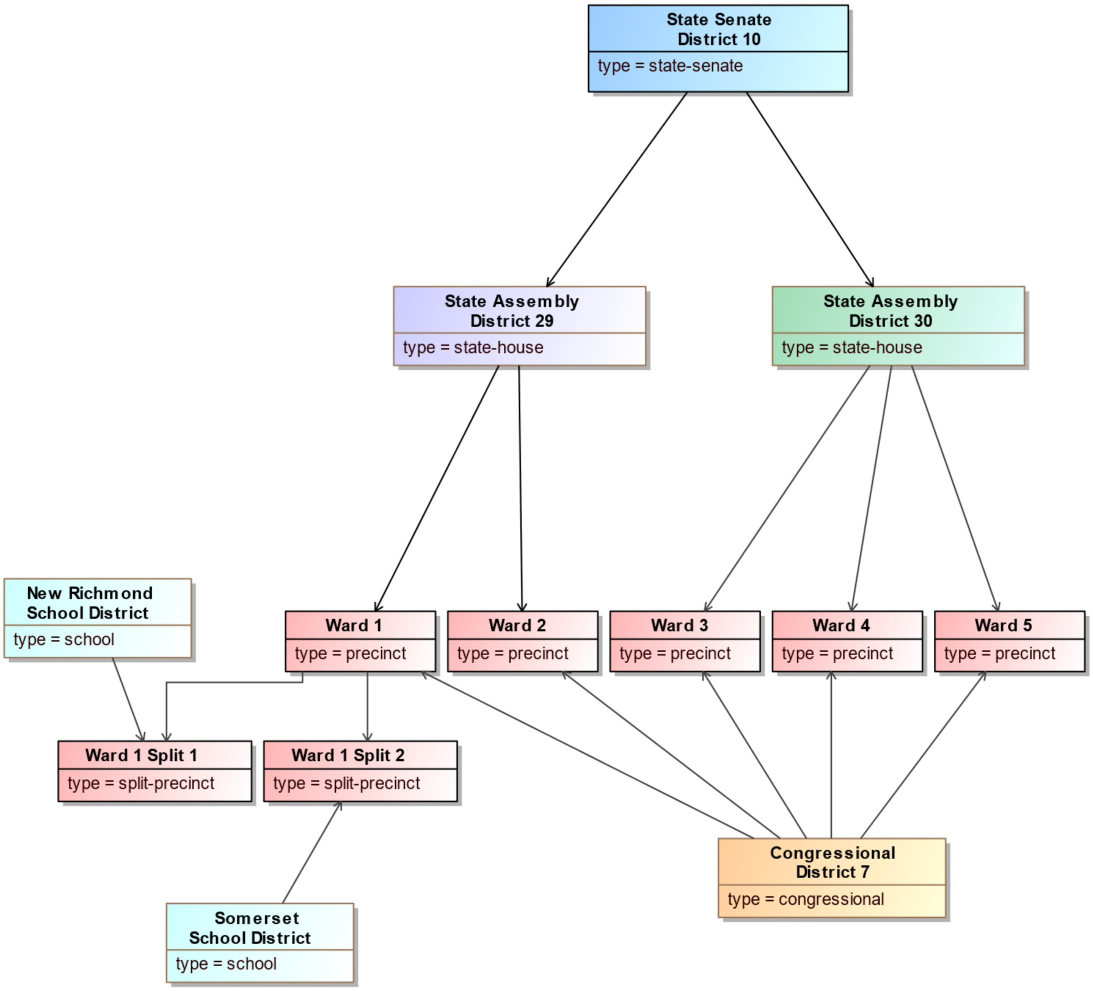


Figure 89 – District <GpUnit> Elements

```
    <!--Definition for State Senate District 10 -->
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-SD10”>
        <ComposingGpUnitIds>OID-HD29 OID-HD30</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>State Senate District 10</Text>
        </Name>
        <Type>state-senate</Type>
    </GpUnit>
    <!--Definition for House Districts 29 and 30 -->
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-HD29”>
        <ComposingGpUnitIds>OID-WARD1 OID-WARD2</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>House District 29</Text>
        </Name>
        <Type>state-house</Type>
    </GpUnit>
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-HD30”>
        <ComposingGpUnitIds>OID-WARD3 OID-WARD4 OID-WARD5</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>House District 30</Text>
        </Name>
        <Type>state-house</Type>
    </GpUnit>
    <!--Definition for Congressional District 7 -->
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-HD30”>
        <ComposingGpUnitIds>OID-WARD1 OID-WARD2 OID-WARD3 OID-WARD4 OID-
WARD5</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>Congressional District 7</Text>
        </Name>
        <Type>congressional</Type>
    </GpUnit>
    <!--Definition for school districts -->
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-NewRichmondSchool”>
        <ComposingGpUnitIds>OID-WARD1-1</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>New Richmond School District</Text>
        </Name>
        <Type>school</Type>
    </GpUnit>
    <GpUnit xsi:type=”ReportingUnit” ObjectId=”OID-SomersetSchool”>
        <ComposingGpUnitIds>OID-WARD1-2</ComposingGpUnitIds>
        <IsDistricted>true</IsDistricted>
        <Name>
            <Text language=”en”>New Somerset School District</Text>
        </Name>
        <Type>school</Type>
    </GpUnit>
```

Figure 90 – Defining Political Geographies and Linking to Precincts

### 6.1.4. Dealing with Duplicate Election Districts

There are, quite often, multiple jurisdictional-wide contests that use, for their respective election districts, the same physical geography. In a given county, for example, contests for county executive and at-large county council positions will all use the county as the election district. Because *<GpUnit >* elements defined for these geographies may have very large *<ComposingGpUnitIds>* sub-elements, instance files could grow unnecessarily large if different but essentially duplicate *<GpUnit >* elements are defined for each of the contests that share what is essentially the same election district.

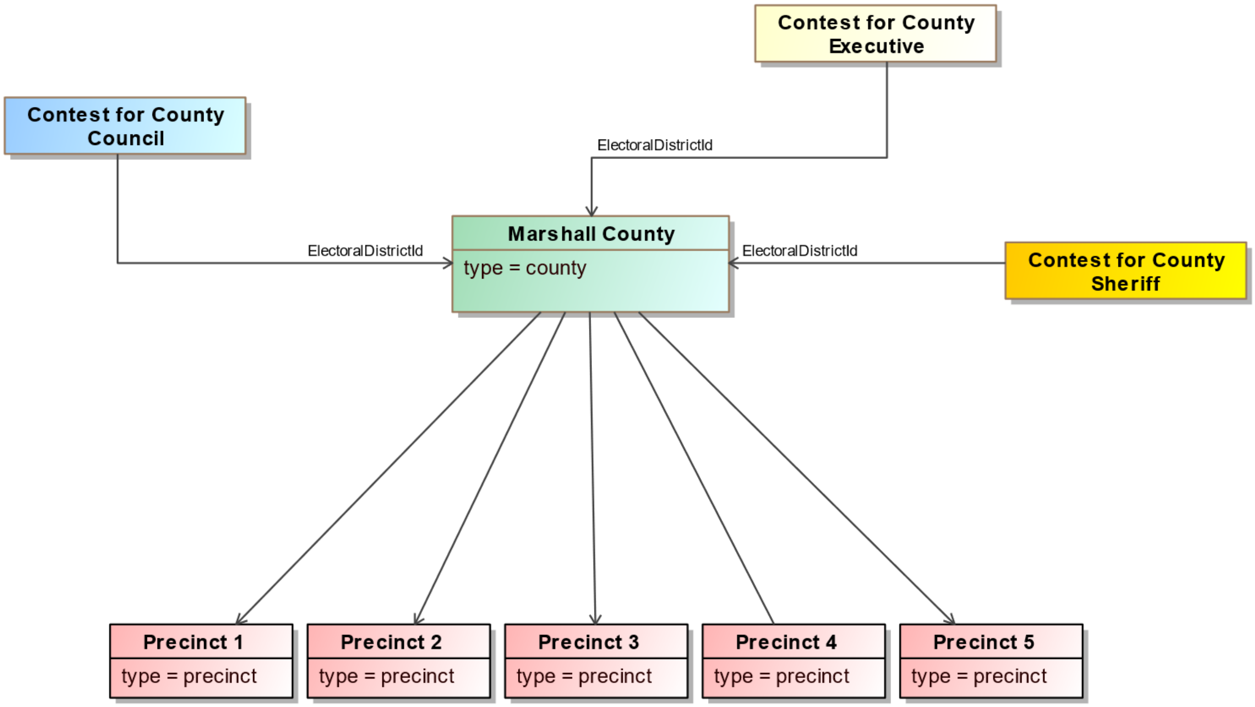


Figure 91 – Re-Using Election District Links

An option for reducing the file size is to re-use the <GpUnit> element defined for the county. Provided that this element is linked to its composing precincts, the contests that are county-wide can use this same <GpUnit> element for the election district. The generating application could, effectively, keep track of the ObjectId attribute of the county <GpUnit> element and then use the associated value for each contest's <ElectionDistrictId> sub-element – see Figure 91.

Thus, if all contest definitions for jurisdictional-wide contests use the same value for the *<Election DistrictId >* sub-element, duplication can be reduced and file sizes may be smaller, especially for larger jurisdictions.

## 6.2. Defining Contests

After the *<GpUnit >* elements have been defined, contests can then be defined using the *<Contest >* element. Defining contests involves the use of several elements, depending on what type of contest is involved and whether and what types of vote counts are desired. The elements needed for contest definitions are as follows:

1. The *<Contest >* abstract element, to identify the name of the contest and other attributes and its type:
    a. *CandidateContest*, for a contest involving candidates;
    b. *BallotMeasureContest*, for a contest involving a ballot measure;
    c. *PartyContest*, for a contest involving a straight party selection; and
2. *RetentionContest*, for a contest involving a judicial or other retention;
3. The *<ContestSelection >* abstract element, to identify a selection on the ballot for that corresponding contest and its type:
    a. *CandidateSelection*, if the selection is for a candidate;
    b. *BallotMeasureSelection*, if the selection is for a ballot measure or retention; and
    c. *PartySelection*, if the selection is for a straight party selection;
4. Within *<ContestSelection >*, *<VoteCounts>* to report the number of votes for the contest selection, for the contest as a whole or broken out by, e.g., precincts and/or types of ballots;
5. A link to a GpUnit element defined for the contest's election district, using the *<Election DistrictId>* sub-element; and
6. The *<Other Counts>* element, to report summary counts for the contest, e.g., overvotes, undervotes, write-ins.

It is possible to define contests without associated *<ContestSelection>* elements as well as without any of the elements for election results, for example, for a pre-election report. However, these elements are required if associating vote counts with contest selections.

```
  <Candidate ObjectId="OID - C1">
    <BallotName>
      <Text Language="en">John Brown</Text>
    </BallotName>
  </Candidate>
  <Candidate ObjectId="OID - C2">
    <BallotName>
      <Text Language="en">Thomas Paine</Text>
    </BallotName>
  </Candidate>
  ...
 <! -- Governor contest -- >
 <Contest xsi:type="CandidateContest" ObjectId="OID - GOV">
   <! -- Vote count for John Brown -- >
   <ContestSelection xsi:type="CandidateSe lection" ObjectId="OID - CS1">
     <VoteCounts>
        <Count>2716</Count>
     </VoteCounts>
     <CandidateIds>OID - C1</CandidateIds>
   </ContestSelection>
   <!--Vote count for Thomas Paine -->
   <ContestSelection xsi:type=”CandidateSelection” ObjectId=”OID-CS2”>
      <VoteCounts>
          <Count>2392</Count>
      </VoteCounts>
      <CandidateIds>OID-C2</CandidateIds>
   </ContestSelection>
   <ElectionDistrictId>OID-State</ElectionDistrictId>
   <Name>Governor</Name>
   <VotesAllowed>1</VotesAllowed>
</Contest>
```

Figure 92 – Basic Contest Definition and Linkage to Vote Counts

*PartyContest* is included as a contest type because a number of states keep track of the number of straight party selections made by voters.

*CandidateContest* is the type of contest likely to be used most often; it is for contests involving candidates and has several elements associated with it including *<PrimaryPartyIds>*, a link to the political party used when a primary election, *<NumberElected>* , for the number of seats associated with the office and *<VotesAllowed>* , for the number of selections a voter can make in the contest.

Figure 92 shows an example of a contest involving *CandidateContest* that illustrates using contest selections and counts, with the counts being aggregated, that is, associated with the election district of the contest as a whole. Contests for *BallotMeasureContest* and *PartyContest* are implemented as per this example.

## 6.3. Defining Controlling Contests

A controlling contest is ballot content structured as a contest that affects other contests controlled by it. The purpose of a controlling contest varies and includes the *selection*, *activation* or *deactivation* of other parts of the ballot based on the voter's selection.

The Ballot Definition (BD) Common Data Format (CDF) supports three of the most popular kinds of controlling contests:

- Activation
- Open Primary Party Preference
- Straight Party

The BD CDF supports these controlling contests via the ControllingContest abstract class and its subtypes. The following sections provide detailed examples showing how each controlling contest type is supported in the Ballot Definition CDF.

Restrictions on controlled contests:

- No two controlling contests can have the same controlled contests.

- Up to one selection is expected in a controlling contest. More than one selection is considered an overvote.

### 6.3.1. Activation defined

Both *Activation contest* and *Open Primary Party Preference* rely on the concept of "activation". Contests on a ballot without any control contests (and not otherwise an open primary ballot) are by default in in a "active", or votable state. The presence of a controlling contest and the selections made in it can modulate the "active" states of other, controlled contests. Activation contests and Open Primary Party Preference contests vary in the default state of their controlled contests. Contests controlled by an activation contest are inactive until a valid selection for the activation option is made. Conversely, contests controlled by an Open Primary Party Preference contest are active unless a valid vote in the controlling contest is made.

### 6.3.2. Activation Contest

An activation contest is a controlling contest containing an activation option, that when voted activates one or more controlled contests. Activation contests are often used for recalls, where an affirmative selection or, in other cases, any selection causes the controlling contest (normally a CandidateContest of replacement options) to become *active*.

Controlled contests to be conditionally activated by the ControllingContest are listed under ControlledContestIds.

An activation contest is represented using the ActivationContest subtype of Contest. A controlling contest of this type uses the ActivationSelection subtype of ContestSelection to list contest options. Each ActivationSelection must indicate whether that selection causes the controlled contests to activate. ActivationContest shares many of the properties of BallotMeasureContest in order to express referendum style language.

In the below example, an ActivationContest contains two ActivationSelection options, with the cs-activation-yes selection activating the controlled contest, cc-replacements. This is specified by the SelectionActivatatesControlled element.

```
<Contest  xsi:type= "ActivationContest"  ObjectId= "cc - activation  - contest" >
    <ContestOption  xsi:type= "A ctivationOption"  ObjectId= "cs - activation  - yes" >
        <CausesActivation>  true </CausesActivation>
        <Selection>
            <Text  Language= "en"  >YES</Text>
        </Selection>
    </ContestOption>
    <ContestOption  xsi:type= "ActivationOption"  ObjectI  d="cs - activation  - no" >
        <CausesActivation>  false  </CausesActivation>
        <Selection>
            <Text  Language= "en"  >NO</Text>
        </Selection>
    </ContestOption>
    <ElectionDistrictId>  ru - st - ignace  </ElectionDistrictId>
    <Name>St.  Ignace Mayoral Recall  </Name>
    <ControlledContestIds>  cc - replacements  </ControlledContestIds>
    <SummaryText>
        <Text Language="en">Shall Danielle Atron be recalled as mayor of St. Ignace?</
Text>
    </SummaryText>
</Contest>
```

### 6.3.3. Open Primary Party Preference

An Open Primary Party Preference controlling contest may appear on an open primary ballot and allows the voter to select which political party's primary they wish to vote. A valid selection in this contest protects the voter from voiding the partisan selection of the ballot by making selections in more than one party's contest.

Controlled contests (partisan CandidateContest instances) to be conditionally deactivated by the ControllingContest are listed under ControlledContestIds.

An Open Primary Party Preference contest is represented using the PartyPreferenceContest subtype of Contest. A control contest of this type uses the PartySelection subtype of ContestSelection to list contest options. Each PartySelection is associated with one or more political parties that have contests on the ballot.

The below example, the "Open Primary Party Preference" contest lists two ContestSelection options, one for each party on the ballot. The ControlledContestIds list the two contests affected by a selection in the ControllingContest.

```
<Contest xsi:type="PartyPreferenceContest" ObjectId="cc-party-preference">
    <ContestOption xsi:type="PartyOption" ObjectId="cs-party-preference-dem">
        <PartyIds>par-dem</PartyIds>
    </ContestOption>
    <ContestOption xsi:type="PartyOption" ObjectId="cs-party-preference-rep">
        <PartyIds>par-rep</PartyIds>
    </ContestOption>
    <ElectionDistrictId>ru-senate</ElectionDistrictId>
    <Name>Open Primary</Name>
    <ControlledContestIds>cc-senate-rep cc-senate-dem</ControlledContestIds>
</Contest>
```

The consequence of a voter selecting one of the PartySelection options is that all contests associated with parties other than the one selected have their contest options *deactivated*.

### 6.3.4. Straight Party

Straight Party is a controlling contest that allows voters to select multiple candidates of the same political party appearing in multiple contests through a single selection of a political party.

Controlled contests to receive indirect selections by the ControllingContest are listed under ControlledContestIds.

A Straight Party contest is represented using the StraightParty subtype of Contest. A control contest of this type uses the PartySelection subtype of ContestSelection to list contest options.

Each PartySelection is associated with one or more political parties that have candidates on the ballot.

The below example, the Party Preference contest lists two ContestSelection options, one for each party with candidates on the ballot. The ControlledContestIds list the contest affected by a selection in the ControllingContest.

The contest controlled by the straight party contest contains two ContestSelection elements.

```
<Contest xsi:type="StraightPartyContest" ObjectId="cc-straight-party">
    <ContestOption xsi:type="PartyOption" ObjectId="cs-straight-party-dem">
        <PartyIds>par-dem</PartyIds>
    </ContestOption>
    <ContestOption xsi:type="PartyOption" ObjectId="cs-straight-party-rep">
        <PartyIds>par-rep</PartyIds>
    </ContestOption>
    <ElectionDistrictId>ru-senate</ElectionDistrictId>
    <Name>Straight Party</Name>
    <ControlledContestIds>cc-senate</ControlledContestIds>
    <StraightPartyRuleset>inclusive</StraightPartyRuleset>
</Contest>
```

The candidates who receive *indirect selections* are those whose PartyId is the same as the PartyIds of the chosen PartySelection in the ControllingContest.

```
...
    <Candidate ObjectId="can-joe-smith">
        <BallotName>
            <Text Language="en">Joe Smith</Text>
        </BallotName>
        <PartyId>par-dem</PartyId>
    </Candidate>
    <Candidate ObjectId="can-alex-mack">
        <BallotName>
            <Text Language="en">Alex Mack</Text>
        </BallotName>
        <PartyId>par-rep</PartyId>
    </Candidate>
...
```

### 6.3.5. Inclusive and Exclusive Rules

StraightPartyContest supports two broad rule sets, designated as inclusive and exclusive. These specify the consequences direct selections made by the voter in controlled contests of a StraightPartyContest. In exclusive, when a valid PartySelection is made, selections made directly to controlled contests withdraws any indirect selections. In inclusive, both selections made directly and indirectly are considered.

## 6.4. Physical Ballot Styles Usages Examples

The following examples give step by step directions to represent physical ballot styles using the BD CDF. Each example consists of two parts, one describing the data required for proper ballot definition and another describing the mCDF instance appearing on the ballot. An OMR and BMD-style example are given.

### 6.4.1. OMR Example

This example gives instructions for producing interoperable ballots containing contest option positions where a voter can indicate their selection.

#### 6.4.1.1. Define a Ballot Format

Define the page size. Pages are defined in terms of their LongEdge, ShortEdge and Orientation. Page sizes cannot vary between ballot styles associated with a given BallotFormat. Units of measure for LongEdge, ShortEdge and all other measurements associated with the ballot format are given by the property MeasurementUnit.

Note: If page sizes need to vary between ballot styles generated by a ballot-producing system, create a separate BallotFormat for each size.

Define an ExternalIdentifier of Type local-level; this will serve as the link between the ballot style identifier mCDF instance appearing on the ballot and the ballot definition.

Define SelectionCaptureMethod as omr. This influences what message type the scanner should expect in the mCDFArea. OMR ballots use the Ballot Style Identification (BSI) message (further described in Appendix A).

It is recommended to define the locations of the mCDF areas using mCDFArea. Otherwise, a scanner will need to manually locate them. A ballot format may contain one or more mCDF areas. This is useful for when a space limitation in a symbology is reached before the logical mCDF message is complete. mCDF provides a feature called continuation that handles this, no special notation needs to be made in the BD to support it.

BD does not provide a particular list of supported symbologies to be used in the mCDFArea. It is anticipated that system integrators, and testing and certification programs specify the required Symbology to support.

The following code fragment shows a BallotFormat defined as 792ptx612pt (equivalent to 8.5in x 11in) with a portrait orientation.

```
<BallotFormat ObjectId="bf-1">
    <ExternalIdentifier>
        <Type>local-level</Type>
        <Value>bf-1</Value>
    </ExternalIdentifier>
    <LongEdge>792</LongEdge>
    <mCDFArea>
        <H>72</H>
        <W>189</W>
        <X>36</X>
        <Y>54</Y>
        <Symbology>DataMatrix</Symbology>
    </mCDFArea>
    <MeasurementUnit>pt</MeasurementUnit>
    <Orientation>portrait</Orientation>
    <SelectionCaptureMethod>omr</SelectionCaptureMethod>
    <ShortEdge>612</ShortEdge>
</BallotFormat>
```

### 6.4.1.2. Create Shapes

Shape primitives can be defined once in the ballot definition and reused throughout. Shape is used by OptionPosition to describe the visual cue for contest option positions, and by FidicualMark to describe the shapes used by points of reference. The ShapeType (e.g., ellipse, rectangle) is always required.

The following defines a Shape of a rectangle with a stroke of 1. The associated measurement unit is inferred from the BallotFormat referenced by an enclosing type (in this case pt).

```
<Shape ObjectId="geo-oval">
    <ShapeType>rectangle</ShapeType>
    <StrokeWidth>1</StrokeWidth>
</Shape>
```

### 6.4.1.3. Create Fiducial Marks (optional)

Fiducial marks are marks printed on the ballot used to calibrate the scanner so that other, non-fiducial positions can be more easily located. Fiducial marks can be shaped as ellipses or rectangles. The BD CDF supports unlimited fiducial marks and they may be defined at the ballot format or contest level.

For ballots using optical mark recognition, it is recommended to add three global fiducial marks, placed so they are non-reversable onto each ballot side in order to align and rotate the ballot. This is particularly important for applications where voters may directly scan their own ballots.

### 6.4.1.4. Define Ballot Styles

Create a BallotStyle for each logical ballot in the Election. BallotStyle instances contain a set of OrderedContest instances for listing the contests on the ballot. As the name implies, the order contests are emitted is significant.

A Physical property is used to store the physical details about how a given contest is represented using a particular ballot format. Each PhysicalContest associates with the BallotFormat and zero or more PhysicalContestOption instances. Each PhysicalContestOption links to the ContestSelection it represents (unless the option is for a write-in) and the positions on the ballot where selection can be indicated.

#### 6.4.1.5. Positions

Contest positions on the ballot use the same general structure (as they all inherit from BoundedObject). Areas on the ballot are described in terms of their bounding box, Sheet and Side.

##### Option Positions

Create an OptionPosition for each area where a voter can make a filling mark to select an option on the ballot. Each OptionPosition must specify the NumberVotes associated with an option, and if applicable, the Rank and FractionalVotes.

##### Write-In Positions

Create a WriteInPosition to designate the area where a voter can hand write a selection on the ballot.

WriteInPosition and OptionPosition are not mutually exclusive. If there is an oval or another shape for the voter to indicate a write-in selection, it should be handled via OptionPosition. WriteInPosition is reserved for the provided area where the voter may place a name.

If there is no indicator apart from the area provided to write-in a candidate's name, then the dimensions of the OptionPosition and WriteInPosition will be the same.

#### 6.4.1.6. Generate Ballot Style Identifiers

Interoperable OMR ballots must contain an mCDF instance of the Ballot Style Identification (BSI) message in order for scanners to properly identify the ballot style, sheet and side.

For omr ballots, a logical mCDF message could look like this:

```
NS1|^~&;|BSI|1|1;BAL|052001|1|1|bf-1|http://go.usa.gov/Tla9;ELE|20141104|39153;
```

### 6.4.2. Ballot Selection Records (e.g., BMD) Example

#### 6.4.2.1. Define a Ballot Format

Define the page size. Pages are defined in terms of their LongEdge, ShortEdge and Orientation. Page sizes cannot vary between ballot styles associated with a given BallotFormat. Units of measure for LongEdge, ShortEdge and all other measurements associated with the ballot format are given by the property MeasurementUnit.

Define an ExternalIdentifier of Type local-level; this will serve as a link between the contest selection capture mCDF instance appearing on the ballot and the ballot definition.

Define SelectionCaptureMethod as mcdf. This influences what message type the scanner should expect in the mCDFArea. When mcdf is specified, ballots use the Contest Selection Capture (CSC) message.

It is strongly recommended to define the locations of the mCDF areas using mCDFArea. Otherwise, a scanner will need to manually locate them. A ballot format may contain one or more mCDF areas. This is useful for when a space limitation in a symbology is reached before the logical mCDF message is complete. mCDF provides a feature called continuation that handles this and no special notation needs to be made in the BD to support it.

BD does not provide a particular list of supported symbologies to be used in the mCDFArea. It is anticipated that system integrators, and testing and certification programs specify the required Symbology to support.

The following code fragment shows a BallotFormat defined as 792ptx612pt (equivalent to 8.5in x 11in) with a portrait orientation.

```
<?xml version="1.0" encoding="UTF-8"?>
<BallotFormat ObjectId="bf-1">
    <ExternalIdentifier>
        <Type>local-level</Type>
        <Value>bf-1</Value>
    </ExternalIdentifier>
    <LongEdge>792</LongEdge>
    <mCDFArea>
        <H>72</H>
        <W>189</W>
        <X>36</X>
        <Y>54</Y>
        <Symbology>PDF417</Symbology>
    </mCDFArea>
    <MeasurementUnit>pt</MeasurementUnit>
    <Orientation>portrait</Orientation>
    <SelectionCaptureMethod>mcdf</SelectionCaptureMethod>
    <ShortEdge>612</ShortEdge>
</BallotFormat>
```

BallotStyle instances should be generated in their logical form (PhysicalContest current only applies to ballots using omr).

### 6.4.3. mCDF Examples

Both ballot styles using *SelectionCaptureMethod* of omr (BSI) and mcdf (CSC) use the mCDF to convey details about the ballot style to a scanner. This section describes the common segments between them.

BSI and CSC contain share two mCDF segments, BAL and ELE. Ultimately, the goal of both segments is to uniquely locate a ballot style across elections, ballot formats and jurisdictions. Each field's purpose is further described below:

- BallotStyle Code - locates the ballot style instance
- BallotFormat Code - locates the ballot format used by the containing ballot
- Election Code - locates the election the ballot style belongs to. Avoids a ballot style of the same BallotStyle Code being scanned for the wrong election.

- ElectionScope Code - locates the jurisdiction the ballot style belongs to. To uniquely locate ballot styles across jurisdictions.

In OMR ballots, it is additionally necessary to locate the page of the CDF instance to that target areas on the ballot can be correctly placed.

Table 67 - Describes the required fields for BSI and CSC.

| SEG | Required Segments and Fields | BSI | CSC |
|---|---|---|---|
| BAL | Sheet | R | O |
| BAL | Side | R | O |
| BAL | DefinitionUri | O | O |
| BAL | BallotFormat Code | R | O |
| BAL | BallotStyle Code | R | R |
| ELE | Election Code | O | O |
| ELE | Election Scope | O | O |

Interoperable ballots require, at a minimum, the use of the BAL segment, and specifically *BallotStyle Code* and *BallotFormat Code*. These codes are expressed in the Ballot Definition as ExternalIdentifier instances.

### 6.4.3.1. ExternalIdentifiers

ExternalIdentifier is used throughout NIST CDF classes to describe identifiers that come from various systems and are unlikely to change frequently. These identifiers contrast with the internal identifiers (called ObjectIds) which are used to wire the file together, and may change more often.

Each ExternalIdentifier has at least a Type (e.g., fips, ocd-id, local-level, etc.) and a Value.

mCDF segments reference these ExternalIdentifier instances in order to locate applicable parts of the ballot definition. In the context of interoperable ballot styles, the BSI and CSC mCDF messages reference ExternalIdentifier instances associated with particular classes to locate the right ballot style, the system that produced it (i.e., its BallotFormat), and optionally, to locate the right election and jurisdiction.

ExternalIdentifiers appearing in mCDF segments vary slightly from those represented in JSON or XML. Namely, all ExternalIdentifiers default to Type of local-level (2) if no type is provided.

local-level represents an identifier originating from a local election technology system, such as a voting system EMS.

## 6.5. Defining and Using Re-usable Ballot and Contest Headers

Headings are used on ballots to delineate divisions or sections of a ballot, for example:

- For Contests: Partisan Section, Proposal Section
- For Jurisdictions: City, County, State, National
- For Offices: Judicial, Legislative

Headings can appear without an explicit relationship to other content (i.e., for a contest). This could be used for contest-independent headers (e.g., "GENERAL ELECTION", "OFFICIAL BALLOT", etc.) as well as for footers.

A contest listed on the ballot has heading information that can be included in the *<BallotTitle>* and *<BallotSubTitle>* sub-elements of *<Contest>* . By using the *<BallotStyle>* element, it is also possible to create election-independent contest headings as well as other election-independent ballot headings that can be defined once and then re-used across multiple ballot styles, the advantage being that less space is used and the generation and handling of the XML and JSON content is simplified.

Under *<Election>* , each heading is defined once, as an instance of *<Header>* , and accessible through the use of directed associations from *<OrderedHeader>* .

Contests can also be associated with headers. This is accomplished by nesting *<OrderedContest>* under *<OrderedHeader>* .

A code example follows, showing several headers defined for the ballot style and for contests within the ballot style. The headers are then re-used in subsequent ballot style definitions.

```
<BallotDefinition xmlns="http://itl.nist.gov/ns/voting/1500-20/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Election>
        <BallotStyle>
            <GpUnitIds>ru-beaverton-p1</GpUnitIds>
            <OrderedContent xsi:type="OrderedHeader">
                <HeaderId>he-official</HeaderId>
            </OrderedContent>
            <OrderedContent xsi:type="OrderedHeader">
                <HeaderId>he-nonpartisan</HeaderId>
                <OrderedContent xsi:type="OrderedContest">
                    <ContestId>cc1</ContestId>
                </OrderedContent>
                <OrderedContent xsi:type="OrderedContest">
                    <ContestId>cc2</ContestId>
                </OrderedContent>
            </OrderedContent>
            <OrderedContent xsi:type="OrderedContest">
                <ContestId>cc3</ContestId>
            </OrderedContent>
        </BallotStyle>
        <BallotStyle>
            <GpUnitIds>ru-bentley-p1</GpUnitIds>
            <OrderedContent xsi:type="OrderedHeader">
                <HeaderId>he-official</HeaderId>
            </OrderedContent>
            <OrderedContent xsi:type="OrderedContest">
                <ContestId>cc3</ContestId>
            </OrderedContent>
            <OrderedContent xsi:type="OrderedContest">
                <ContestId>cc4</ContestId>
            </OrderedContent>
        </BallotStyle>
        <Contest ObjectId="cc1" xsi:type="CandidateContest">
            <BallotTitle>
                <Text Language="en">Council Member</Text>
            </BallotTitle>
            <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
            <Name>City Council Member</Name>
            <VotesAllowed>3</VotesAllowed>
        </Contest>
        <Contest ObjectId="cc2" xsi:type="CandidateContest">
            <BallotTitle>
                <Text Language="en">Mayor</Text>
            </BallotTitle>
            <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
            <Name>Mayor</Name>
            <VotesAllowed>1</VotesAllowed>
        </Contest>
        <Contest ObjectId="cc3" xsi:type="BallotMeasureContest">
            <BallotTitle>
                <Text Language="en">GALDWIN COUNTY ANIMAL SHELTER</Text>
            </BallotTitle>
            <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
            <Name>Animal Shelter</Name>
            <SummaryText>
                <Text Language="en">Shall the tax limiation...</Text>
                </SummaryText>
            </Contest>
            <Contest ObjectId="cc4" xsi:type="BallotMeasureContest">
                <BallotTitle>
                    <Text Language="en">FILE MILAGE PROPOSAL</Text>
                </BallotTitle>
                <ElectionDistrictId>ru-bentley</ElectionDistrictId>
                <Name>File Milage Proposal</Name>
                <SummaryText>
                    <Text Language="en">Shall the tax limiation...</Text>
                </SummaryText>
            </Contest>
            <ElectionScopeId>ru-michigan</ElectionScopeId>
            <Name>
                <Text Language="en">General Election</Text>
            </Name>
            <StartDate>2018-01-29</StartDate>
            <EndDate>2018-01-29</EndDate>
            <Type>general</Type>
        </Election>
        <GeneratedDate>2018-01-29T00:00:00Z</GeneratedDate>
        <GpUnit xsi:type="ReportingUnit" ObjectId="ru-michigan">
            <ComposingGpUnitIds>ru-beaverton</ComposingGpUnitIds>
            <Name>
                <Text Language="en">Michigan</Text>
            </Name>
            <Type>state</Type>
        </GpUnit>
        <GpUnit xsi:type="ReportingUnit" ObjectId="ru-beaverton">
            <ComposingGpUnitIds>ru-beaverton-p1</ComposingGpUnitIds>
            <Name>
                <Text Language="en">Beaverton</Text>
            </Name>
            <Type>city</Type>
        </GpUnit>
        <GpUnit xsi:type="ReportingUnit" ObjectId="ru-beaverton-p1">
            <Name>
                <Text Language="en">City of Beaverton, Precinct 1</Text>
            </Name>
            <Type>precinct</Type>
        </GpUnit>
        <GpUnit xsi:type="ReportingUnit" ObjectId="ru-bentley">
            <ComposingGpUnitIds>ru-bentley-p1</ComposingGpUnitIds>
            <Name>
                <Text Language="en">Bentley</Text>
            </Name>
            <Type>township</Type>
        </GpUnit>
        <GpUnit xsi:type="ReportingUnit" ObjectId="ru-bentley-p1">
            <Name>
                <Text Language="en">Bentley Township, Precinct 1</Text>
            </Name>
            <Type>precinct</Type>
        </GpUnit>
        <Header ObjectId="he-official">
            <Name>
                <Text Language="en">OFFICIAL BALLOT</Text>
            </Name>
        </Header>
        <Header ObjectId="he-nonpartisan">
            <Name>
                <Text Language="en">NONPARTISAN SECTION</Text>
            </Name>
        </Header>
        <Header ObjectId="he-city">
            <Name>
                <Text Language="en">CITY</Text>
            </Name>
        </Header>
        <Header ObjectId="he-proposal">
            <Name>
                <Text Language="en">PROPOSAL SECTION</Text>
            </Name>
        </Header>
        <Header ObjectId="he-township">
            <Name>
                <Text Language="en">TOWNSHIP</Text>
            </Name>
        </Header>
        <Header ObjectId="he-county">
            <Name>
                <Text Language="en">COUNTY</Text>
            </Name>
        </Header>
        <Issuer>State of Michigan</Issuer>
        <IssuerAbbreviation>MI</IssuerAbbreviation>
        <SequenceStart>0</SequenceStart>
        <SequenceEnd>0</SequenceEnd>
        <VendorApplicationId>MI POC</VendorApplicationId>
        <Version>1.0.0</Version>
</BallotDefinition>
```

## 7. Processing of interoperable ballots

This section provides some details of the ballot definition format relevant for manufacturers of ballot-aware scanners. Manufacturers of ballot-producing systems should also read this section so they may better understand how interoperable ballot styles can be achieved.

**Coordinate System –** Origin is top left corner of the page. When using portrait orientation, X-coordinate values increase along the short edge, and Y-coordinate values increase along the long edge. Conversely, when using landscape orientation, X-coordinate values increase along the long edge, and Y-coordinate values increase along the short edge.

Calculations on coordinate values should be performed with at least single floating-point precision.

**Ballot Style Detection** – Scan each ballot side for a mCDF instance. Optionally, check for symbology defined within the bounding boxes defined in each BallotFormat. For each ballot format:

1. Check for defined locations of mCDFArea (if no mCDFArea is defined manual detection techniques must be used)
2. Attempt to decode the content in the given symbology

**Detecting side of sheet (OMR only)–** Side of sheet is encoded into the mCDF instance appearing on one or more sides of the sheet. If there is one mCDF instance per side, then the side is determined by decoding the mCDF instance on that side. If a single mCDF instance appears on the sheet, then the side lacking a mCDF instances takes the opposite side. For example, if the side containing the mCDF instance reports the side as front, then the opposite side is inferred to as the back.

**Rotation –** Check for defined fiducial marks. Incorrectly placed or inadequate quantity of fiducial mark cannot accurately determine the orientation of a page.

**Deskewing –** Check for defined fiducial marks. Global fiducial marks are defined under BallotFormat. Local fiducial marks are defined under PhysicalContest. Note the sheet and side associated with the fiducial!

**Image distortion** – Minor image distortion (e.g., elongation) caused by the scanning process can be resolved by the use of global and local fiducials. Check for defined fiducial marks. Global fiducial marks are defined under BallotFormat. Local fiducial marks are defined under PhysicalContest. Note the sheet and side associated with the fiducial!

## References

[1] Object Management Group (OMG), UML Specification version 1.1 (OMG document ad/97-08-11) September 22, 2011, https://omg.org/ [accessed 12/20/2022].

[2] W3C, Extensible *Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation, November 26, 2008, https://www.w3.org/TR/xml/ [accessed 12/20/2022].

[3] JavaScript Object Notation, https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf [accessed 12/20/2022].

[4] Dziurłaj J (2022) Recommendations for Voting System Interoperability. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Grant/Contractor Report (GCR) NIST GCR 22-034. https://doi.org/10.6028/NIST.GCR.22-034

[5] Dziurłaj J (2023) Micro Common Data Format Specification. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication 1500 (SP) NIST SP 1500-19. https://doi.org/10.6028/NIST.SP.1500-19

[6] mCDF Profile for Contest Selection Capture, Available at https://github.com/usnistgov/CastVoteRecords/tree/feature/mCDF/mCDF%20Profile%20for%20Contest%20Selection%20Capture [accessed 12/19/2022].

[7] Wack J (2019) Election Results Common Data Format Specification Revision 2.0. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication 1500 (SP) NIST SP 1500-100r2. https://doi.org/10.6028/NIST.SP.1500-100r2

[8] City of Cambridge, Massachusetts, *Geographic Information Systems: Map Gallery* [Web site], https://www.cambridgema.gov/GIS/mapgallery [accessed 2/1/2019].

[9] Hilkert, Scott, "Application of Lessons Learned in the Illinois EDC Project Toward Emerging Election Data Standards and System Guidelines," submitted to *Workshop for a Common Data Format for Electronic Voting Systems*, National Institute of Standards and Technology, Gaithersburg, Maryland, October 29-30, 2009, https://www.nist.gov/system/files/documents/itl/vote/hilkert-Lessons-Learned-in-Illinois-EDC-1-1.pdf [accessed 11/25/2020].

[10] Voluntary Voting Systems Guidelines, Version 2.0, February 10, 2021, Available at https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf [accessed 01/05/2023]

## Appendix A. mCDF Profile for Ballot Style Identification

This profile of the NIST Ballot Definition Common Data Format 1.0 is designed to support the identification of ballot styles encoded onto paper ballots using various machine readable symbologies. The profile includes only the BD classes and properties required to convey the following:

- The election associated with the ballot;
- The ballot style associated with the ballot;
- The ballot format associated with the ballot; and
- The election authority who created the ballot style

## A.1. Reading the profile

This profile is a subset of the NIST Ballot Definition CDF Data Model. Each field in a segment (detailed below) contains a “CDF Mapping” column, which points back to the CDF property profiled. It is strongly recommended to look up these cross-references to understand the correct use of each property.

The profile is structured in terms of mCDF messages, segments, data types and enumerations. Each segment is accompanied by a table listing each field, if it is required, whether it has a default value, and its name. Messages, segments, and data types are referenced using three character codes (e.g., ELE for the Election segment).

## A.2. Messages

### A.2.1. BSI (Ballot Style Identification)

The mCDF Profile for Ballot Style Identification describes a single message, Ballot Style Identification (BSI) to allow scanners to identify an OMR style ballot. Its structure is given by the following abstract syntax: NS1 BAL ELE?

That is, a valid BSI message consists of a NS1 segment, an BAL segment and optionally an ELE segment. Note: The NS1 Segment is defined in the mCDF specification.

To use this message, set the following NS1 field values:

| Field | Value |
|---|---|
| NS1-3 | BSI |
| NS1-4 | 1 |

Table 68 - Describes NS1 field values

### A.2.1.1. Segments

#### BAL (BallotStyle)

The BAL segment consists of one or more ExternalIdentifiers(s), for associating a mCDF instance with a BallotStyle. A uri may be provided for associating a mCDF instance with a Ballot Definition CDF instance. The Uri can be used by downstream processors to lookup names and other data associated with ExternalIdentifiers, such as names of contests, contest options, etc.

| Order | Datatype | Multiplicity | Default | Attr Name | CDF Mapping |
|---|---|---|---|---|---|
| 1 | ExternalIdentifier (EID) | 1..1 | | BallotStyle Code | BallotDefinition::BallotStyle::ExternalIdentifier |
| 2 | BallotSideType (BST) | 1..1 | 1 | Side | |
| 3 | Integer | 1..1 | 1 | Sheet | |
| 4 | ExternalIdentifier (EID) | 1..1 | | BallotFormat Code | BallotFormat::ExternalIdentifier |
| 5 | anyURI | 0..1 | | DefinitionUri | |

Table 69 - Describes BAL (BallotStyle) CDF mappings

**BAL-1 BallotStyle Code**

For associating the mCDF instance with the BallotStyle containing it.

**BAL-2 Side**

The side of the ballot style that the mCDF instance appears.

**BAL-3 Sheet**

The sheet of the ballot style that the mCDF instance appears.

**BAL-4 BallotFormat Code**

The ballot format associated with the ballot containing the mCDF instance.

**BAL-5 DefinitionUri**

The uri to the ballot definition instance containing the ballot style.

#### ELE (Election)

For associating an mCDF message with an Election. ElectionScope Code is used to convey the authority conducting the election.

| Order | Datatype | Multiplicity | Default | Attr Name | CDF Mapping |
|---|---|---|---|---|---|
| 1 | ExternalIdentifier (EID) | 1..1 | | Election Code | BallotDefinition::Election::ExternalIdentifier |
| 2 | ExternalIdentifier (EID) | 0..1 | | ElectionScope Code | BallotDefinition::GpUnit::ExternalIdentifier |

Table 70 - Describes ELE (Election) CDF mappings

### ELE-1 Election Code

For associating the mCDF instance with a particular Election.

### ELE-2 ElectionScope

For associating the mCDF instance with the jurisdiction conducting the Election.

## A.2.1.2. Abstract Data Types

### EID (ExternalIdentifier)

| Order | Datatype | Multiplicity | Default | Attr Name | CDF Mapping |
|---|---|---|---|---|---|
| 1 | String | 0..1 | | Value | BallotDefinition::ExternalIdentifier::Value |
| 2 | IdentifierType (IDT) | 0..1 | 2 | Type | BallotDefinition::ExternalIdentifier::Type |
| 3 | String | 0..1 | | OtherType | BallotDefinition::ExternalIdentifier::OtherType |

Table 71 - Describes EID (ExternalIdentifier) CDF mappings

### A.2.1.3. Enumerations

Enumerations in the profile convey identical data as their CDF counterparts. However, they use integer literals instead of text values in order to save space.

#### BST (BallotSideType)

The following table maps the BallotSideType enumeration literals into numeric values.

| Name | Value |
|---|---|
| back | 2 |
| front | 1 |

Table 72 - Describes BST (BallotSideType) field values

#### IDT (IdentifierType)

The following table maps the IdentifierType enumeration literals into numeric values.

| Name | Value |
|---|---|
| fips | 1 |
| local-level | 2 |
| national-level | 3 |
| ocd-id | 4 |
| state-level | 5 |
| other | 6 |

Table 73 - Describes IDT (IdentifierType) field values

## A.3. Examples

### A.3.1. Simple Usage

The following example demonstrates basic use of the mCDF syntax. A mCDF instance containing the “BSI” message is constructed. The BSI instance is encoded on the front of a single sheet ballot style, identified using a `local-level` identifier. The ballot format identifier (`local-level`) is also given to associate the ballot with the correct physical details in the ballot definition. The election code is given as a `local-level` identifier. Finally, the election jurisdiction is given as a `fips` code.

```
NS1|^~\&;|BSI|1|1;BAL|112115|1|1|1-
ess|http://go.usa.gov/ballotdef;ELE|331332219|26-37520^1;
```

The physical message begins with the NS1 header and declaration of delimiters (^~\&; ). The semicolon (;) character is used to separate segments while | separates fields. BAL (BallotStyle) is composed of five fields: BallotStyle Code (112115), Side (1), Sheet (1), BallotFormatCode (1-ess), and DefinitionUri (http://go.usa.gov/ballotdef) where the ballot definition may be located.

Information about the election (ELE) with which the BallotStyle is associated concludes the message and is composed of the Election Code (331332219) and the ElectionScopeCode (26-37520) with an explicitly specified Type indicating that the scope is represented with a FIPS code (1). ^ is used to delimit a component in a field. All segments end with a semicolon (; ).

Note that both the BallotStyle Code and Election Code use the data type ExternalIdentifier. The Type of ExternalIdentifier is "2", indicating a 'local-level' identifier. As this is the default value for IdentifierType, it may be omitted.

## A.4. Mapping from mCDF to Ballot Definition

The BSI message is used to on OMR ballots in order to identify the ballot style and activate the correct ballot. A simplified process flow is shown below:

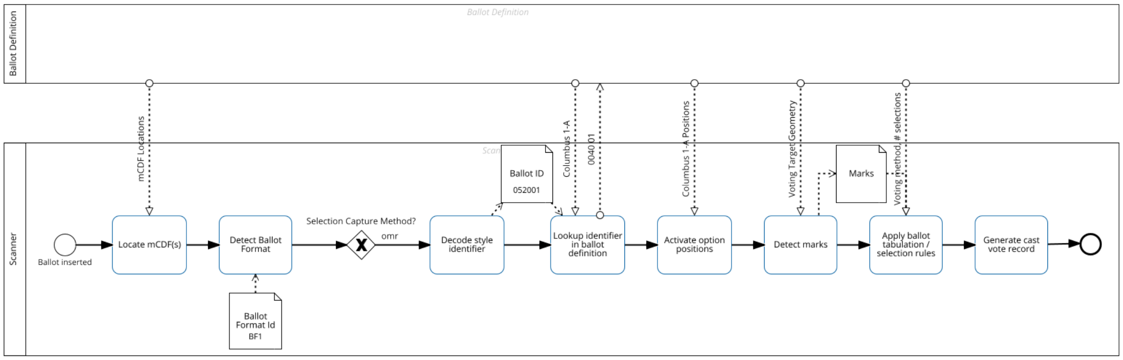


Figure 93 - Process of scanning an interoperable OMR ballot

This example will show how structures in the BSI message map back to the NIST Ballot Definition.

This section references fragments from code examples mapping_bsi_mcdf_to_bd.xml.

Consider the following mCDF BSI message:

```
NS1|^~&;|BSI|1|1;BAL|052001|1|1|BF1|http://go.usa.gov/Tla9;ELE|20141104|39153;
```

Every mCDF segment in a message has a table in the documentation that lists its fields includes a CDF mapping column. This column can be used to understand the relationship between the data points collected via mCDF and its representation in a larger JSON/XML CDF instance.

Note: Some fields may have no equivalent in the profiled CDF, such fields are specific to use-cases of the mCDF such as identifying the sheet or side a mCDF message appears. For example, the NS1 segment is used to envelope the contents of an mCDF message. Nothing in the header is mapped to a CDF.

For the BAL segment, BAL-1 maps to BallotDefinition::BallotStyle::ExternalIdentifier, which is illustrated as XML in this fragment:

```
<BallotStyle>
    <ExternalIdentifier>
        <Type> local   - level   </Type>
        <Value>  052001 </Value>
    </ExternalIdentif          ier>
    <GpUnitIds>   prec  - 52</GpUnitIds>
</BallotStyle>
```

BAL-2 and BAL-3 represent the side and sheet the mCDF message appears and are used by BoundedObjects (e.g., OptionPosition, WriteInPosition) to locate and activate the positions for the correct sheet and side of a ballot style. The following XML fragment shows a physical contest with an option position described in terms of sheet and side.

```
<OrderedContent xsi:type="OrderedContest">
    <ContestId>_9CC</ContestId>
    <Physical>
        <BallotFormatId>bf-1</BallotFormatId>
        <PhysicalContestOption>
            <ContestOptionId>_CS1HBR</ContestOptionId>
            <OptionPosition>
                <H>10</H>
                <Sheet>1</Sheet>
                <Side>front</Side>
                <W>10</W>
                <X>414.6800316</X>
                <Y>120.94517325</Y>
                <IndicatorId>geo-oval</IndicatorId>
                <NumberVotes>1</NumberVotes>
            </OptionPosition>
        </PhysicalContestOption>
        ...
    </Physical>
</OrderedContent>
```

BAL-4 maps to BallotDefinition::BallotFormat::ExternalIdentifier, represented as the following XML fragment:

```
...
<BallotFormat ObjectId="bf-1">
    <ExternalIdentifier>
        <Type>local-level</Type>
        <Value>BF1</Value>
    </ExternalIdentifier>
    <LongEdge>1224</LongEdge>
    <MeasurementUnit>pt</MeasurementUnit>
    <Orientation>portrait</Orientation>
    <SelectionCaptureMethod>omr</SelectionCaptureMethod>
    <ShortEdge>612</ShortEdge>
</BallotFormat>
...
```

Finally, BAL-5 represents the Definition Uri, which is not mapped to any one thing in the ballot definition, but is a reference to the entire ballot definition instance that various parties can use to decode mCDF instances.

For ELE, ELE-1 maps to BallotDefinition::Election::ExternalIdentifier and ELE-2 maps to BallotDefinition::GpUnit::ExternalIdentifier.

ELE-1 maps to the following BD XML fragment:

```
...
<Election>
    ...
    <ExternalIdentifier>
```

```
        <Type>local-level</Type>
        <Value>2014-11-04G</Value>
    </ExternalIdentifier>
    ...
</Election>
...
```

Likewise, ELE-2 maps to the following BD XML fragment:

```
<BallotDefinition>
    <Election>
        ...
        <ElectionScopeId>gpu-39153</ElectionScopeId>
    </Election>
    <GpUnit xsi:type="ReportingUnit" ObjectId="gpu-39153">
        <ExternalIdentifier>
            <Type>local-level</Type>
            <Value>39153</Value>
        </ExternalIdentifier>
        <Type>county</Type>
    </GpUnit>
</BallotDefinition>
```

That is, ELE-2 represents the GpUnit that represents the scope of the election.

## A.5. Retrieval with XPath

Retrieval of correct parts of the ballot definition instance can be achieved through the use of structured XPath expressions (for ballot definitions in XML). The following example represents the BallotFormat whose local-level typed ExternalIdentifier Value is set to the variable $targetBallotFormat:

```
/BallotDefinition/BallotFormat[ExternalIdentifier[Type = 'local-level'
and Value = $targetBallotFormat]]
```

## Appendix B. Acronyms

Selected acronyms and abbreviations used in this document are defined below.

| | |
|---|---|
| BMD | Ballot Marking Device |
| CDF | Common Data Format |
| EMS | Election Management System |
| ID | Identifier |
| JSON | JavaScript Object Notation |
| RCV | Ranked Choice Voting |
| UML | Unified Modeling Language |
| URI | Uniform Resource Identifier |
| XML | eXtensible Markup Language |
| QR Code | Quick Response Code |
| PDF417 | PDF417 |

## Appendix C. File Download Locations

The files associated with this specification are available for download from a NIST repository, whose address is:

https://github.com/usnistgov/BallotDefinition

These files are also available from:

http://vote.nist.gov

The files include:

- This specification,
- microCDF profile,
- UML model,
- XML and JSON schemas, and
- Example files.