

Draft NIST Special Publication 500-XXX

A Common Data Format for Election Results Reporting

John P. Wack, NIST

Kim Brace, EDS

Sam Dana, Prometheus Computing

Herb Deutsch, ES&S

John Dziurlaj, Ohio SoS Office

Ian Piper, Dominion Voting

Don Rehill, AP

Richard Rivello, NIST

Sarah Whitt, Wisconsin GAB

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.SP.XXX>

NIST
**National Institute of
Standards and Technology**
U.S. Department of Commerce

DRAFT NIST Special Publication 500-XXX

A Common Data Format for Election Results Reporting

John P. Wack, NIST
Kim Brace, EDS
Sam Dana, Prometheus Computing
Herb Deutsch, ES&S
John Dziurlaj, Ohio SoS Office
Ian Piper, Dominion Voting
Don Rehill, AP
Richard Rivello, NIST
Sarah Whitt, Wisconsin GAB

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.SP.XXX>

Month and Year of Publication



U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

Authority

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 500-XXX
Natl. Inst. Stand. Technol. Spec. Publ. 500-XXX, NNN pages (Month YYYY)
CODEN: NSPUE2

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.SP.XXX>

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

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

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST.

Public comment period: *Month Day, YYYY* through *Month Day, YYYY*

All comments are subject to release under the Freedom of Information Act (FOIA).

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

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. This Special Publication 500-series reports on ITL's research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

This publication describes the NIST ElectionResultsReporting common data format for election-related data and for election results reporting. It provides the XML schema for this format and the UML model that the format was derived from, as well as background information about election results reporting and how geo-political geography is structured and used in the model and schema. The XML format is comprehensive and at the same time very flexible, able to accommodate most/all election scenarios used throughout the U.S. It is part of a series of planned common data format specifications for voting equipment.

Keywords

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

Acknowledgements

The authors John P. Wack of the National Institute of Standards and Technology (NIST), Kim Brace of Election Data Services, Sam Dana of Prometheus Computing, Herb Deutsch of Election Systems and Software, John Dziurlaj of the Ohio Secretary of State's Office, Ian Piper of Dominion Voting, Don Rehill of the Associated Press, Richard Rivello of NIST, and Sarah Whitt of the Wisconsin Government Accountability Board, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge the following contributors for their keen and insightful assistance with developing this specification: Kenneth Bennett, Thomas Connolly, Art Greisser, Chris Jerdonik, Arthur Keller, Jared Marcotte, Lauren Massa-Lochridge, Neal McBurnett, John McCarthy, Janet Modrow, Justin Moore, Tammy Patrick, Benjamin Rice, John Sebes, Paul Stenbjorn, Beth Ann Surber, David Tarrent, and David Webber, as well as other members of the IEEE Voting System Standards Committee.

Note to Reviewers

This will include information about how to review this specification.

Executive Summary

This publication is a specification for a common data format (CDF) for election-related data and election results reporting. The format, known as the NIST ElectionResultsReporting CDF, is comprehensive and detailed in its coverage of election results data and at the same time very flexible, able to accommodate most/all election scenarios used throughout the U.S. This publication contains the XML schema specification, usage guidance, and background information about election results reporting and how geo-political geography is structured and used in election results.

The ElectionResultsReporting CDF specification was written so that manufacturers of election management systems (EMS) and election jurisdictions could implement it in their systems and databases. By doing so, the election data can be imported or exported among voting devices from different manufacturers, thereby increasing interoperability and saving election jurisdictions from having to create software to translate between vendor proprietary formats. Interoperable data will also reduce costs to election jurisdictions as well as reduce the complexity in election management.

This specification is geared towards the following audiences:

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

The format accommodates three different election scenarios:

Pre-election. The period prior to an election, in which various reports can be produced that may include basic information about the reporting jurisdiction but not yet complete information about the election.

Election. The period during which an election is being conducted and election results reports are produced. The reports will likely include only aggregated results data but, depending on the capabilities of the reporting jurisdiction, could include more detailed, precinct-level reporting.

Post-election. The period after the close of polls when more detailed election results reports can be produced that can be very detailed, down to the level of precincts and split precincts reporting and possibly reporting of results from specific voting devices.

The XML schema associated with the ElectionResultsReporting CDF is derived from a UML model that was designed to accommodate different types of geo-political geography and how the geographies interrelate and are used throughout the U.S and territories. The model was also designed to accommodate different types of contests and their many variations, and to provide the capability to report on these contests at very high levels of aggregation as well as to very fine levels of detail. The detail can include combinations of the following:

- Reporting by precinct and splits of precincts
- Reporting by ballot type, e.g., absentee, election day
- Reporting by device type and specific voting device

The UML model can be re-used and modified to meet the needs of other planned common data format specifications for voting devices such as electronic pollbooks and ballot marking devices.

It is envisioned that this specification will be required in a future version of the EAC's (Election Assistance Commission) VVSG (Voluntary Voting System Guidelines) [1].

Table of Contents

Executive Summary	vi
1 Introduction	1
1.1 Purpose	1
1.2 Audience	2
1.3 Motivation and methodology	2
1.4 Document Structure	3
2 Background: Geo-Political Geography and Election Results Reporting	4
2.1 The Primary Types of Geo-Political Geography	4
2.1.1 Governmental-based Geography	5
2.1.2 Political-based Geography	6
2.1.3 Administrative-based Geography	6
2.2 Linking the Geo-Political Geographies Together	6
2.3 Geo-Political Geography in the UML Model and XML Schema	10
3 Use Cases and ElectionResultsReporting UML Model	12
3.1 Use Cases for Election Results Reporting	12
3.1.1 Pre-Election Use Case	12
3.1.2 Election Night Reporting Use Case	13
3.1.3 Post-Election Reporting Use Case	14
3.2 ElectionResultsReporting UML Model	14
3.2.1 Major Classes in the Model	15
3.2.2 Relationships between Classes	17
4 ElectionResultsReporting XML Schema	19
4.1 Schema Stylistic Conventions	19
4.2 Elements and Types	19
4.2.1 The <AnnotatedString> Type	20
4.2.2 The <BallotSelection> Element and Extension Base	21
4.2.3 The <BallotStyle> and <OrderedContest> Elements	27
4.2.4 The <Candidate> Element	30
4.2.5 The <ContactInformation> Element	32
4.2.6 The <Contest> Element and Extension Base	34

4.2.7	The <CountStatus> Element	43
4.2.8	The <Counts> Extension Base Element.....	44
4.2.9	The <Device> Element	46
4.2.10	The <Election> Element	47
4.2.11	The <ElectionAdministration> Element.....	50
4.2.12	The <ElectionReport> Element	51
4.2.13	The <ExternalIdentifiers Element>	54
4.2.14	The <GpUnit> Element and Extension Base	56
4.2.15	The <InternationalizedText> Element and LanguageString Type.....	62
4.2.16	The <Office> Element.....	64
4.2.17	The <OfficeGroup> Element.....	66
4.2.18	The <Party> Element and Extension Base.....	68
4.2.19	The <PartyRegistration> Element.....	71
4.2.20	The <Person> Element.....	72
4.2.21	The <Schedule> and <Hours> Elements.....	74
4.2.22	The <SpatialDimension> and <SpatialExtent> Elements	76
4.2.23	The <SummaryCounts> (extension base <Counts>) Element	78
4.2.24	The <VoteCounts> (extension base <Counts>) Element	80
4.3	Enumerations.....	81
4.3.1	The BallotMeasureType Enumeration	81
4.3.2	The CandidatePostElectionStatus Enumeration.....	82
4.3.3	The CandidatePreElectionStatus Enumeration	83
4.3.4	The CountItemStatus Enumeration	84
4.3.5	The CountItemType Enumeration.....	85
4.3.6	The DayType Enumeration.....	86
4.3.7	The DeviceType Enumeration	87
4.3.8	The ElectionType Enumeration	88
4.3.9	The GeoSpatialFormat Enumeration	89
4.3.10	The IdentifierType Enumeration	90
4.3.11	The OfficeTermType Enumeration	91
4.3.12	The ReportDetailLevel Enumeration.....	92
4.3.13	The ReportingUnitType Enumeration	93

4.3.14 The ResultsStatus Enumeration	95
4.3.15 The VoteVariation Enumeration	96
4.4 Primitives	97
4.4.1 The HTMLColorString Primitive	97
4.4.2 The Primitive ShortString Primitive	97
4.4.3 The Primitive TimeWithZone Primitive.....	97
5 ElectionResultsReporting Schema Aspects and Usage.....	98
5.1 How UML Relationships Are Implemented in the XML Schema	98
5.1.1 Directed Association Relationships and the <i>objectId</i> Attribute.....	99
5.2 Wrapper Elements	100
5.3 General Layout of XML Instance Files	100
5.4 Defining Geo-Political Geography Using the <GpUnit> Element	101
5.4.1 Defining and Associating <GpUnit> Elements	102
5.4.2 Using One <GpUnit> Element for Duplicate Electoral Districts	104
5.5 Defining Contests Using the <Contest> Element.....	105
5.6 Associating Vote Counts with Contests	107
5.6.1 Associating Aggregated Vote Counts	107
5.6.2 Associating Vote Counts Broken Down by Precincts plus Ballot and Device Type	108
5.6.3 Associating Summary Vote and Ballot Counts	109
5.7 Associating Summary Ballot Counts with Geo-political Units.....	110

List of Appendices

Appendix A— Acronyms	112
Appendix B— Glossary	113
Appendix C— References	118
Appendix D— File Download Locations.....	119
Appendix E— ElectionResultsReporting XML Schema	120

List of Figures

Figure 1 – Governmental Geographies	5
---	---

Figure 2 - Overlapping Non-hierarchical Electoral Districts	7
Figure 3 - Ward and Precincts in Cambridge, MA.	8
Figure 4 - Districts Overlaying Wards and Precincts	9
Figure 5 - Simplified ElectionResultsReporting Class Diagram.....	15
Figure 6 - Directed Composition Example	17
Figure 7 - Is a Type of Example	17
Figure 8 - Directed Association Example	17
Figure 9 – Is a Type of Relationship in XML.....	99
Figure 10 – Directed Association in XML	99
Figure 11 - Use of Wrapper Elements	100
Figure 12 - Geo-political Geography Example from Wisconsin	102
Figure 13 - <GpUnit> Definitions.....	103
Figure 14 - Linking <GpUnit> Elements to Form Relationships.....	104
Figure 15 - Basic Contest Definition and Linkage to Vote Counts	106
Figure 16 - Linking Aggregated Vote Counts to Contests	108
Figure 17 - Linking Vote Counts to Precincts	108
Figure 18 - Filtering Vote Counts by Device and Ballot Type.....	109
Figure 19 - Associating Summary Counts with Contests.....	110
Figure 20 - Associating Summary Ballot Counts with <GpUnit> Elements.....	111

List of Tables

Table 4.1 – Attributes for <AnnotatedString>	20
Table 4.2 – Attributes for <BallotSelection>	21
Table 4.3 – Elements for <BallotSelection>	21
Table 4.4 – Elements for <BallotSelection xsi:type="BallotMeasureSelection">	23
Table 4.5 – Elements for <BallotSelection xsi:type="CandidateSelection">	24
Table 4.6 – Elements for <BallotSelection xsi:type="PartySelection">	26
Table 4.7 – Attributes for <BallotStyle>	27
Table 4.8 – Elements for <BallotStyle>	27
Table 4.9 – Elements for <OrderedContest>.....	29
Table 4.10 - Attributes for <Candidate>	30

Table 4.11 - Elements for <Candidate>.....	30
Table 4.12 - Attributes for <ExternalIdentifiers>	32
Table 4.13 - Elements for <ContactInformation>.....	32
Table 4.14 – Attributes for <Contest>.....	35
Table 4.15 – Elements for <Contest>.....	35
Table 4.16 – Elements for <Contest xsi:type="BallotMeasureContest">	38
Table 4.17 – Elements for <Contest xsi:type="CandidateContest">.....	40
Table 4.18 – Elements for <Contest xsi:type="RetentionContest">.....	42
Table 4.19 – Elements for <CountStatus>	43
Table 4.20 – Elements for <Counts>.....	44
Table 4.21 – Elements for <Device>	46
Table 4.22 – Elements for <Election>	47
Table 4.23 – Elements for <Device>	50
Table 4.24 – Elements for <ElectionReport>.....	51
Table 4.25 - Attributes for <ExternalIdentifiers>	54
Table 4.26 - Elements for <ExternalIdentifiers>	54
Table 4.27 – Attributes for <GpUnit>.....	57
Table 4.28 – Elements for <GpUnit>.....	57
Table 4.29 – Elements for <GpUnit xsi:type="ReportingDevice">	58
Table 4.30 – Elements for <GpUnit xsi:type="ReportingUnit">.....	59
Table 4.31 – Attributes for <InternationalizedText>.....	62
Table 4.32 – Elements for <InternationalizedText>	62
Table 4.33 – Attributes for <Text>	63
Table 4.34 – Attributes for <Office>.....	64
Table 4.35 – Elements for <Office>.....	64
Table 4.36 – Attributes for <Term>.....	65
Table 4.37 – Elements for <Term>.....	65
Table 4.38 – Attributes for <OfficeGroup>.....	66
Table 4.39 – Elements for <OfficeGroup>.....	66
Table 4.40 – Attributes for <Party>.....	68
Table 4.41 – Elements for <Party>.....	68
Table 4.42 – Elements for <Party xsi:type="Coalition">	70

Table 4.43 – Elements for <PartyRegistration>	71
Table 4.44 – Attributes for <Person>	72
Table 4.45 – Elements for <Person>	72
Table 4.46 – Attributes for <Schedule>	74
Table 4.47 – Elements for <Schedule>	74
Table 4.48 – Attributes for <Schedule>	75
Table 4.49 – Elements for <Hours>	75
Table 4.50 – Elements for <SpatialDimension>	76
Table 4.51 – Elements for <SpatialExtent>	77
Table 4.52 – Elements for <SummaryCounts>	78
Table 4.53 – Elements for <VoteCounts>	80
Table 4.54 – Values for <BallotMeasureType>	81
Table 4.55 – Values for <CandidatePostElectionStatus>	82
Table 4.56 – Values for <CandidatePreElectionStatus>	83
Table 4.57 – Values for <CountItemStatus>	84
Table 4.58 – Values for <CountItemType>	85
Table 4.59 – Values for <DayType>	86
Table 4.60 – Values for <DeviceType>	87
Table 4.61 – Values for <ElectionType>	88
Table 4.62 – Values for <GeoSpatialFormat>	89
Table 4.63 – Values for <IdentifierType>	90
Table 4.64 – Values for <OfficeTermType>	91
Table 4.65 – Values for <ReportDetailLevel>	92
Table 4.66 – Values for <ReportingUnitType>	93
Table 4.67 – Values for <ResultsStatus>	95
Table 4.68 – Values for <VoteVariationType>	96

1 Introduction

This publication is a specification for an XML-based (eXtensible Markup Language) [2] common data format (CDF) for exporting election and election results data from voting systems used for managing elections and tabulating election results across states and territories of the United States. The format, known as the NIST ElectionResultsReporting CDF, allows for reporting of election data known in advance about elections, for reporting election night results, and for reporting on updates and certified results from the post-election canvass. It can be used for reporting election results from distributed voting places to central offices of the county or state, and from county or state offices to news media and the general public. The electronic formats can also be used as a basic export of election information from an election management system (EMS) and for combining election data from different EMSs or transferring election data between EMSs.

This specification includes a data model in UML (Unified Modeling Language) [3] that itemizes and defines the data involved in election management and election results reporting. The XML format is derived from the UML model.

The primary features of this specification include:

- Major data elements and their attributes and associations are fully defined in a UML data model
- The data model can be used to generate data formats for today's election systems as well as for election systems to be developed
- Election data and results can be reported at flexible levels from highly aggregated to very detailed
- Detailed reporting includes by device type, by type of ballot, and by geo-political geographies including precinct and split precinct
- Geo-political units of geography can be specified in a flexible manner that will accommodate most or all states and jurisdictions reporting structures
- Major elements such as for contests, geo-political units, and parties include the capability for multiple types of identifiers
- Detailed instructions for implementation and use of the XML schema is included

1.1 Purpose

The purpose of this specification is to provide an election results reporting XML format that manufacturers of voting equipment can integrate into their products and that election offices, the media and other groups can use to facilitate election management and reporting and analysis of election results.

Advantages of using this specification include:

- Election results can be reported directly from election offices in this format regardless of voting systems manufacturer, thus enabling (or providing) interoperability

- The need for custom software and custom reporting formats can be greatly reduced
- Jurisdictions that use multiple versions of EMSs and tabulators can more easily combine the results into one report and transfer information between systems
- Potential improvements in the usability and accuracy of election reconciliations, audits, and analysis can be realized
- Consistency in election results reports across different voting systems will make reporting on election performance, e.g., for the EAC (Election Assistance Commission) election survey [4] and other election analyses, easier and more accurate

1.2 Audience

The intended audience of this specification includes election officials, manufacturers and developers, as well as others in the election community including the general public. Election results reporting is deceptively complex, thus some background in election administration is useful in understanding the material in this specification. .

1.3 Motivation and methodology

This document was motivated primarily to reduce the inherent complexity for U.S. election officials in collecting election results and publishing them, especially on election night when time frames are tight and there are more opportunities for error and a greater need for automation. The process of reporting election results is a highly complicated activity that occurs over several different time frames and in multiple scenarios. The equipment involved and data produced often do not interoperate, adding more complexity to the process. Additionally, there are sometimes significant variations among different jurisdictions within a state as well among the states themselves in the way they perform election results reporting.

NIST worked with a community of U.S. election officials, analysts, and voting system manufacturers to analyze different reporting scenarios and their associated geo-political geographies throughout the United States, to analyze existing voting systems and how equipment in the future may operate, as well as to analyze how election results are themselves typically studied and used post-election. NIST also utilized, in its analysis, several other XML schemas associated with voting:

- The State of Florida XML schema for election results reporting [5]
- The Pew Voting Information Project XML schema version 3.0 [6]
- The OASIS Election Markup Language (EML) XML schema version 7.0 [7]

This resulted in the following three-step process:

1. Development of three use cases for election reporting activities that correspond to the different time frames and reporting scenarios:
 - **Pre-election** – describes election data that is known ahead of the election; basically an export from an EMS of the contests, candidates, ballot initiatives,

- information about offices, and the geo-political geographies associated with the reporting jurisdiction
 - **Election night** – addresses reporting of election results to varying levels of detail. The results can be summarized by contest and jurisdiction or broken down by individual reporting units such as precincts. The reports can be formatted in various ways, including as updates or corrections to previous reports, or as internal intermediate reports within a state or county
 - **Post-election** – addresses updates and the final results compiled during the post-election canvass
2. Development of an election results UML data model that represents the data of the use cases and defines how the data are associated and organized,
 3. Lastly, generation of the XML data formats from the UML model.

The advantages of using a UML data model include that the model is independent of the concrete XML format (or other potential formats that could be derived); if changes are needed to the XML format, one can make changes to the model and generate or derive a new version of the format using commercial products.

Note that this specification addresses U.S. elections and is not intended for use “as is” in elections in other countries. However, the specification was written with the intention that it be adaptable to other election environments. Terminology used in this specification would naturally differ from that used in other countries.

1.4 Document Structure

Chapters 1 and 2 deal with introductory and background material. One of the most complicated aspects of election results reporting is understanding and dealing with geo-political units of geography and how they are structured differently across states and jurisdictions, therefore Chapter 2 includes an overview of geo-political geographies, describing how they overlay each other and how results are tied to them. Chapter 3 contains an overview of the three use cases for election results reporting and the UML data model that implements the use cases. Chapter 4 contains documentation for the XML schema that is derived from the UML model. Chapter 5 describes how to use the major features of the schema.

The appendices include references, definitions, acronyms, instructions for downloading the files associated with this specification, and the XML schema listing.

2 Background: Geo-Political Geography and Election Results Reporting

This chapter provides an overview of the geo-political geography in the United States as it relates to elections and election reporting, and serves to provide background for how geo-political geography is implemented in the ElectionResultsReporting model and schema described in chapters 3-5. Geo-political geography is both very simple and very complex, but it is the underpinning for understanding election setup and election results reporting. Election results are tied to geo-political geographies, thus understanding how the geographies are categorized and how they interrelate will be helpful in understanding the model and schema.

2.1 The Primary Types of Geo-Political Geography

The primary types of geo-political geography include jurisdictions (that is, geographies that run elections) such as states, counties, and cities, as well as many types of electoral districts (tied to contests), precincts, and various other geographical units associated with political boundaries. Generally, the media and election analysts wish to obtain voting results broken out by these units, thus the process of running an election includes associating vote counts with these units so that they can be ultimately reported.

Precincts are generally the smallest unit and building block of geo-political geography, and as in many states, there is generally one polling place per precinct. Ballot counts and vote counts for contests can be associated with a variety of different types of geo-political geography, ranging from aggregated counts associated with a county or state down to more detailed counts associated with a precinct and breakdowns of precinct.

However, this is changing. Many states now use vote centers or absentee voting that reduces, eliminates, or combines the number of districts. Still, the notion of a precinct continues to exist, and precincts can be thought of as the bricks that compose geo-political geography.

Geo-political geography can be quite complex; some of the geo-political geographies form hierarchical, but others overlap, and some can change their boundaries quite often such as every few years or even every few days. Cities and districts in particular can divide precinct lines, splitting them into multiple parts (called split precincts), with each part requiring separate ballots. Quite often, different states or counties within the same state may report elections in different ways, thus associating results with geo-political units differently in each case.

The following sections break down geo-political geography into three primary types and show how the geographies are related. These three types are:

1. Governmental-based Geography
2. Political-based Geography
3. Administrative-based Geography

2.1.1 Governmental-based Geography

Governmental-based geography refers to entities that have generally been set up from the beginning of time within the state and that tend to not change over time, with the exception of some cities. For many states, the governmental-based geography is hierarchical, as shown in Figure 1. These can be categorized as follows:

- States
- Counties and parishes
- Cities
- Indian Reservations
- Towns and Townships
- Other Civil Divisions

Nearly all states have counties, although some use different words to describe them, e.g., parishes for Louisiana and boroughs for Alaska. Within the counties in 20 states are townships, which observe their 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. Townships in Michigan, Minnesota and Wisconsin also run their elections, but report their information to the county, which then reports to the state.

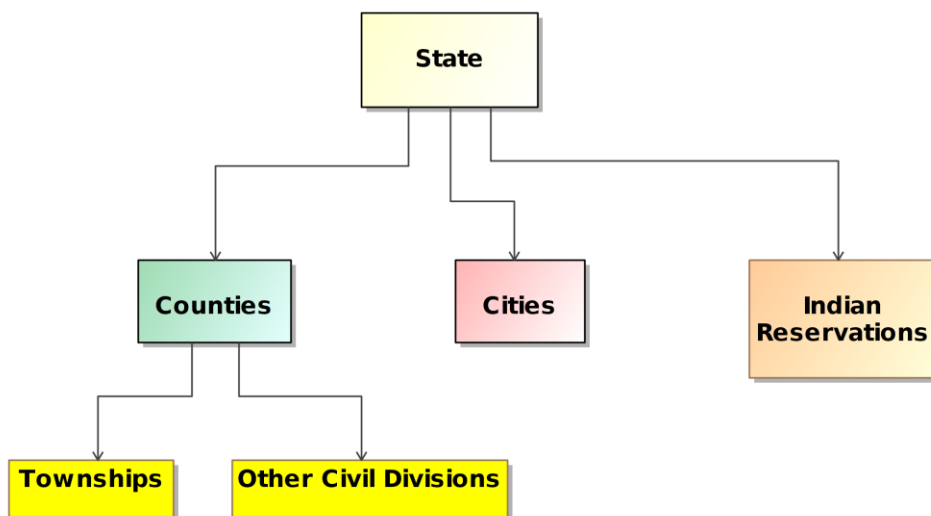


Figure 1 – Governmental Geographies

Reporting on election results for a governmental-based geography requires knowledge of which precincts compose the geography. Governmental-based geographies generally 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. Accordingly, changes to city boundaries may result in crossing the boundaries of one or more precincts, creating split precincts and requiring a distinct ballot style per split precinct.

Generally, governmental-based geographies are associated with offices that are elected

jurisdiction-wide (such as for Governor, County Clerk, Supervisor, Treasurer, Assessor, Highway Commissioner, etc.) and thus do not require different ballot style areas within the geography for those offices, i.e., all voters in the jurisdiction vote for the office.

2.1.2 Political-based Geography

Political-based geographies are those that tend to be population-based and therefore may change with each U.S. Census every 10 years in a process known as re-districting. Political-based geographies are generally known as electoral districts, where people are elected to an office that has jurisdiction within a specific geography, e.g., a U.S. Congressional district. These political-based 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 districts
- City districts (sometimes called Wards)
- Numerous other forms of electoral districts

2.1.3 Administrative-based Geography

Administrative-based geographies are called thus because their boundaries are determined through election or civil administration. Administrative-based geographies include precincts and their various types such as wards, combined precincts, and split precincts. They can sometimes be very small, sometimes only applying to a several or a single house along a street, and thus difficult to track and maintain correct boundary information. They can involve territory that is non-contiguous in itself, e.g., for some of the taxing and special districts. They can change a number of times throughout a given year, even daily in some cases. These administrative-based geographies 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 e.g., fire, water, sewer, transit, school, police, hospital, utilities
- Special districts, e.g., unique areas brought together for a referendum

2.2 Linking the Geo-Political Geographies Together

As an example of administrative-based geographies and their relationship to political-based and governmental-based geographies, Figure 3 shows the wards and precincts (administrative-based) that make up the city of Cambridge, MA, (governmental-based), and Figure 4 shows how the wards and precincts in the city compose the U.S. Congressional electoral districts [8]. The wards are implemented as collections of precincts. As a general rule, it is preferred that electoral

districts are composed of whole precincts, as in these examples. In many states, the boundaries of electoral districts may overlay the precincts in less orderly ways as a result of re-districting, creating one or more split precincts, with a distinct ballot style per split precinct. Depending on the number of districts and how they overlap the precincts and overlay each other, the resultant number of ballot styles created may grow substantially beyond the number of whole precincts.

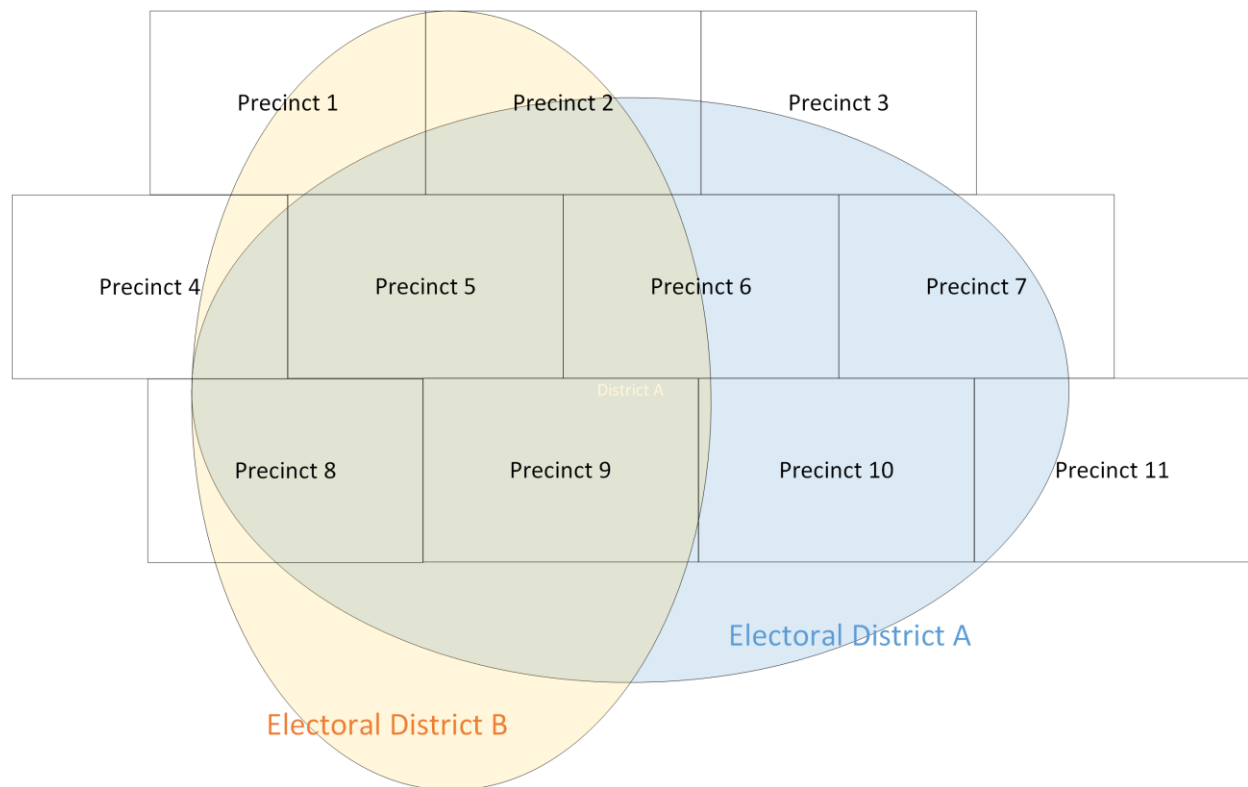


Figure 2 - Overlapping Non-hierarchical Electoral Districts

As an example, Figure 2 shows two electoral districts that overlay a series of precincts. Every time a precinct is not wholly contained within either of the districts, the precinct is split into however many pieces are necessary. Figure 2 shows that a number of the precincts are split in different ways, e.g., Precinct 6 is split into two pieces.

Precinct 6 will thus require 2 distinct ballot styles, one for each of the splits. To correctly tabulate the votes in the different electoral districts, it will be necessary to know which split of Precinct 6 is contained within each of the two electoral districts.

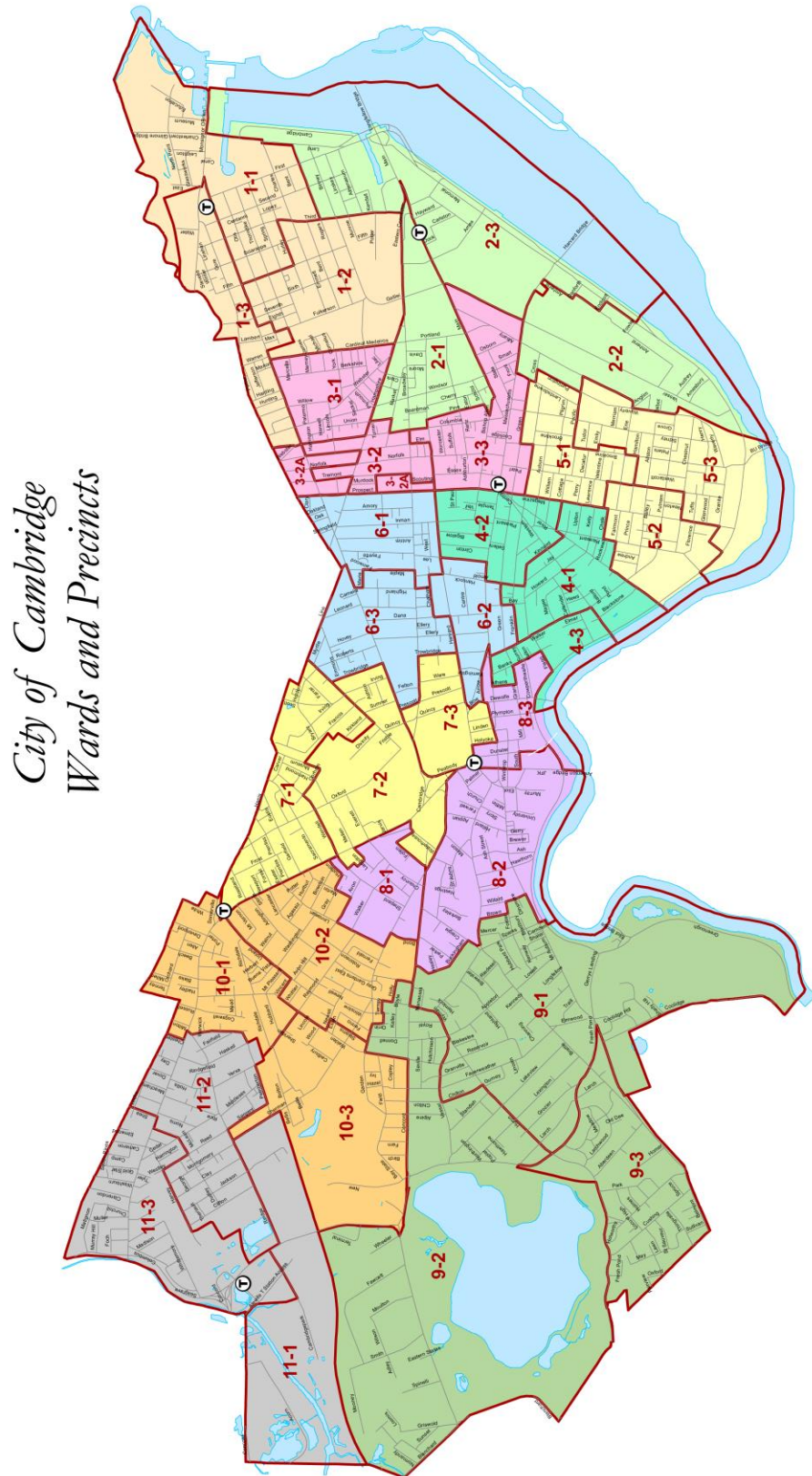


Figure 3 - Ward and Precincts in Cambridge, MA.

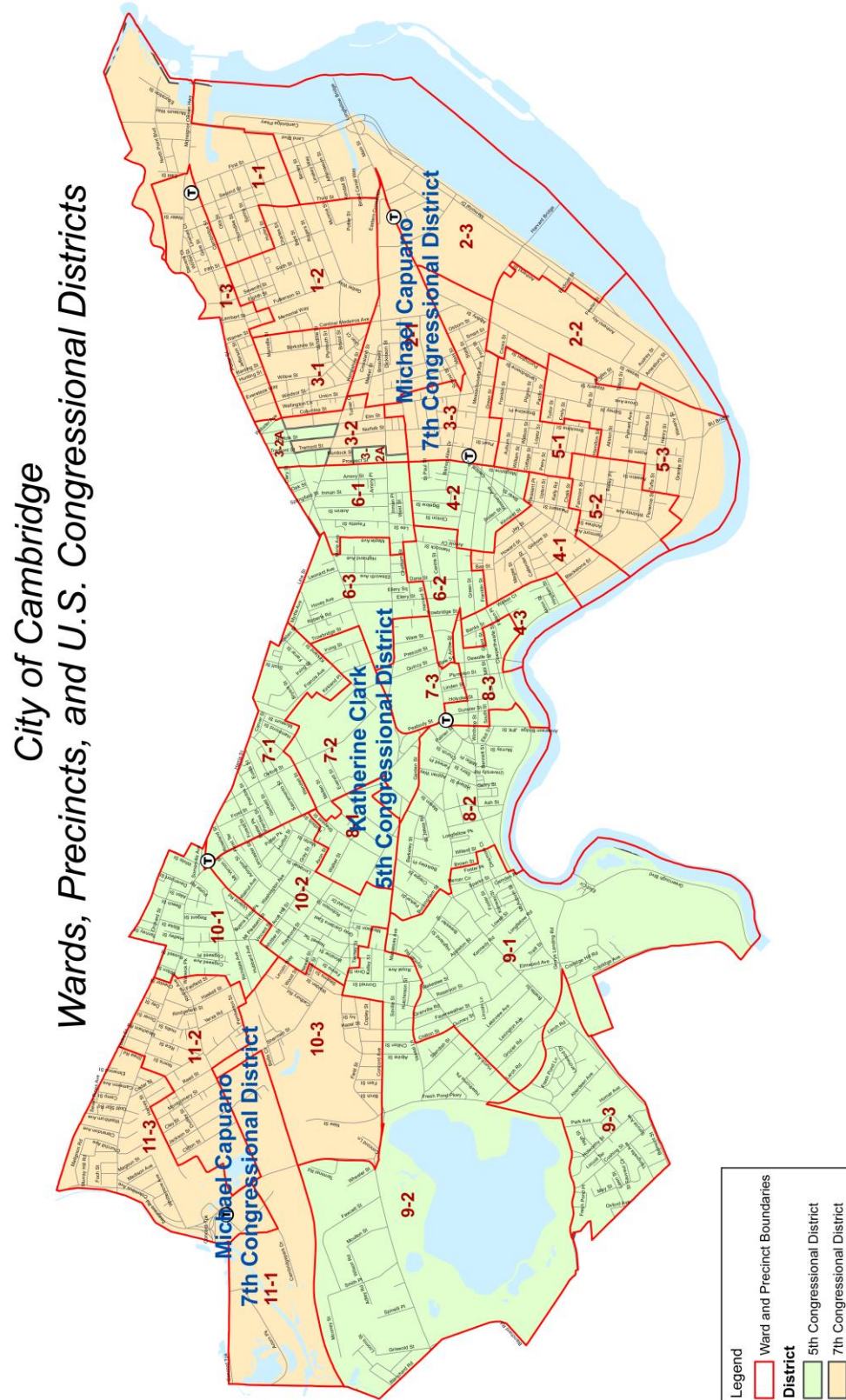


Figure 4 - Districts Overlaying Wards and Precincts

Precincts can be split as well by changes to the other administrative-based 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 any one of the ballots. This adds further complexity and places additional demands on election jurisdictions on their ability to manage and report details of votes on election night and post-election.

To make this situation more manageable, some states/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 Geo-Political Geography in the UML Model and XML Schema

The previous discussion served to show that there are different types of geo-political geography that may change often, or never, and that they can often overlap each other, or behave hierarchically, resulting sometimes in very complex maps and many small geo-political units that require distinct, different ballot styles. Election officials may spend considerable time in managing this complexity.

Furthermore, each state and sometimes county or city will do certain things differently, using different combinations of units such as combined precincts or wards, and having different rules about how the associated contests operate. When one combines (a) the complexities of geo-political geography with (b) the different election rules employed in the U.S. states and territories, one sees that running an election can be an extremely complicated endeavor. Election results reporting mirrors this complexity.

It is important to note that the different geographies form relationships much as a lattice, in which objects can be related in non-hierarchical ways. The ElectionResultsReporting model and schema models geo-political geography in this way using a object that can be linked with other units depending on the type of geo-political geography it is modeling. In the UML model described in Chapter 3, this unit is referred to as the GeoPoliticalUnit class, and in the XML schema described in Chapters 4-5 it is called the `<GpUnit>` element. GeoPoliticalUnits have a type value, that is, whether they are a type of district, or county, or precinct, etc. Depending on the type, the GeoPoliticalUnits are defined and related to each other in ways that can mirror the real-world geo-political geography of the reporting jurisdiction.

Broadly speaking, GeoPoliticalUnits can be structured hierarchically when modeling jurisdictional geographies. If a GeoPoliticalUnit models a jurisdiction that runs/reports on elections, then the GeoPoliticalUnits can be related hierarchically in that the lowest-level objects, i.e., precincts, will be children of the election-running object, say the county or city.

If a GeoPoliticalUnit models a district, there is no inherent hierarchy as they relate to jurisdictions, except that the districts can be related to the precincts and/or split precincts that compose them.

Chapter 3 provides more information about the ElectionResultsReporting UML model, and chapters 4 and 5 describe the generated XML schema.

3 Use Cases and ElectionResultsReporting UML Model

This chapter describes three use cases for election results reporting covered by this specification, and then describes the ElectionResultsReporting UML model that implements the use cases and is used to generate the XML schema. This chapter is provided for background and to assist in understanding how the model and schema are structured and used.

3.1 Use Cases for Election Results Reporting

This section describes three use cases for the following phases of election results reporting:

- **Pre-election** – for reports of election data that is known ahead of elections including any election-specific information, the contests, candidates, ballot measures, information about offices, political parties, and the geo-political geographies associated with the reporting jurisdiction
- **Election night** – for reporting of election results to the public during the election or on election night and for internal state rolls-ups of election results between and among election management systems to varying levels of detail
- **Post-election** – for reporting the final results compiled during the post-election canvass

3.1.1 Pre-Election Use Case

The pre-election 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. Pre-election reports are useful for a variety of purposes, including 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.

Pre-election 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 pre-election 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
 - As corrections to previous reports
- All jurisdictions, districts, offices, 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, candidates, office holders, etc.

- Any specific election information about an election such as election type, date, place, jurisdiction, authority information, registered voters. The type of election could include
 - Open and closed partisan primaries
 - Runoffs
 - Special elections
 - General elections
- Any defined ballot measures and contests, including retention contests
- Candidates on the ballot and their associated party affiliations and contact information
- Offices associated with the contests and electoral districts
- Any vote-capture device types or the specific devices associated with polling places.
- Ballot styles containing contest and candidate information in the order as they appear on the ballot at specific precincts or split precincts.

3.1.2 Election Night Reporting Use Case

The election night reporting use case generally addresses aggregated, contest-level reporting of election results. Depending on a jurisdiction's capabilities, it may include further details such as precinct-level vote and ballot counts and breakdowns of counts by device type and ballot type. Thus, there is a continuum between the election night reporting and post-election use cases that will be jurisdiction-specific. For simplicity's sake, the election night reporting use case includes only aggregated reporting of elections by contest totals or by precinct totals; the post-election use case includes finer details of reporting data.

The election night reporting use case involves election reporting managed in two primary ways:

1. By the states or counties (and sometimes cities) directly, with the counties reporting outward to the public/media and possibly reporting to the state, or
2. As part of county-wide (or sometimes city-wide) processes, with the counties reporting upward to the state and the state reporting outward to the public/media.

When reporting upward to the state, election results are reported from local jurisdictions upward to the city or county or state level, where the results continue to be aggregated in a process referred to as rolling up the votes and creating state or county roll-ups. As part of this process, internal, intermediate reports from different EMSs or databases may need to be combined. The roll-ups are then formatted and released to the media and public during election night until all results from the local jurisdictions are accounted for. The results are, at this stage, considered as unofficial.

The data for election night reporting includes the following:

- Any and all information from the pre-election use case
- Aggregated contest results for offices and ballot measures, including
 - Votes reported in the contest
 - Votes reported for each candidate(s) or selection(s) in the contest
 - Overvotes reported for the contest

- Undervotes reported for the contest
- Further vote details as possible to report, including:
 - Breakdowns of votes by various geo-political geographies such as precinct or split precinct.
 - Vote counts and ballot counts broken down by ballot types (absentee, early, etc.) and type of device (electronic, optical scan, etc.).
 - Additional counts as available, e.g., the number of votes cast by party or the number of straight party selections.

3.1.3 Post-Election Reporting Use Case

The post-election reporting use case addresses the updates to the election night unofficial results. The post-election results contain greater detail than election night results, including results by reporting unit, by counts of different types of ballots (early, absentee, provisional, etc.), and by the type of vote-capture device used by voters to cast their ballots. They may also include results that were not available for election night reporting, such as absentee ballots received too close to the election day, or provisional ballots. This use case is of interest especially by analysts and media performing detailed analysis of election results and voting trends.

The data for post-election night reporting includes:

- Any and all information from the pre-election and election night reporting use cases
- Ballot measure and contest vote counts, including:
 - By geo-political units of geography (e.g., county, township, city, precinct, etc.)
 - By type of ballot/voting (e.g., absentee, early, in-person, provisional)
 - By type of voting equipment used (e.g., electronic, optical scan, etc.)
 - By overvotes and undervotes for each contest
- Ballot measure and contest summaries broken down as per the vote counts, but also including the number of ballots on which the ballot measure or contest appeared
- Precinct summaries broken down as per the vote counts

3.2 ElectionResultsReporting UML Model

This section presents the ElectionResultsReporting 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 UML relationships so as to understand how the model and XML schema are structured.

The UML model represents a format-independent description of the data involved in election results reporting (as required by the three 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 such as 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 5 shows a high-level view of the class diagram, minus its attributes and certain ancillary classes (the complete diagram is available for download, see File Download Locations).

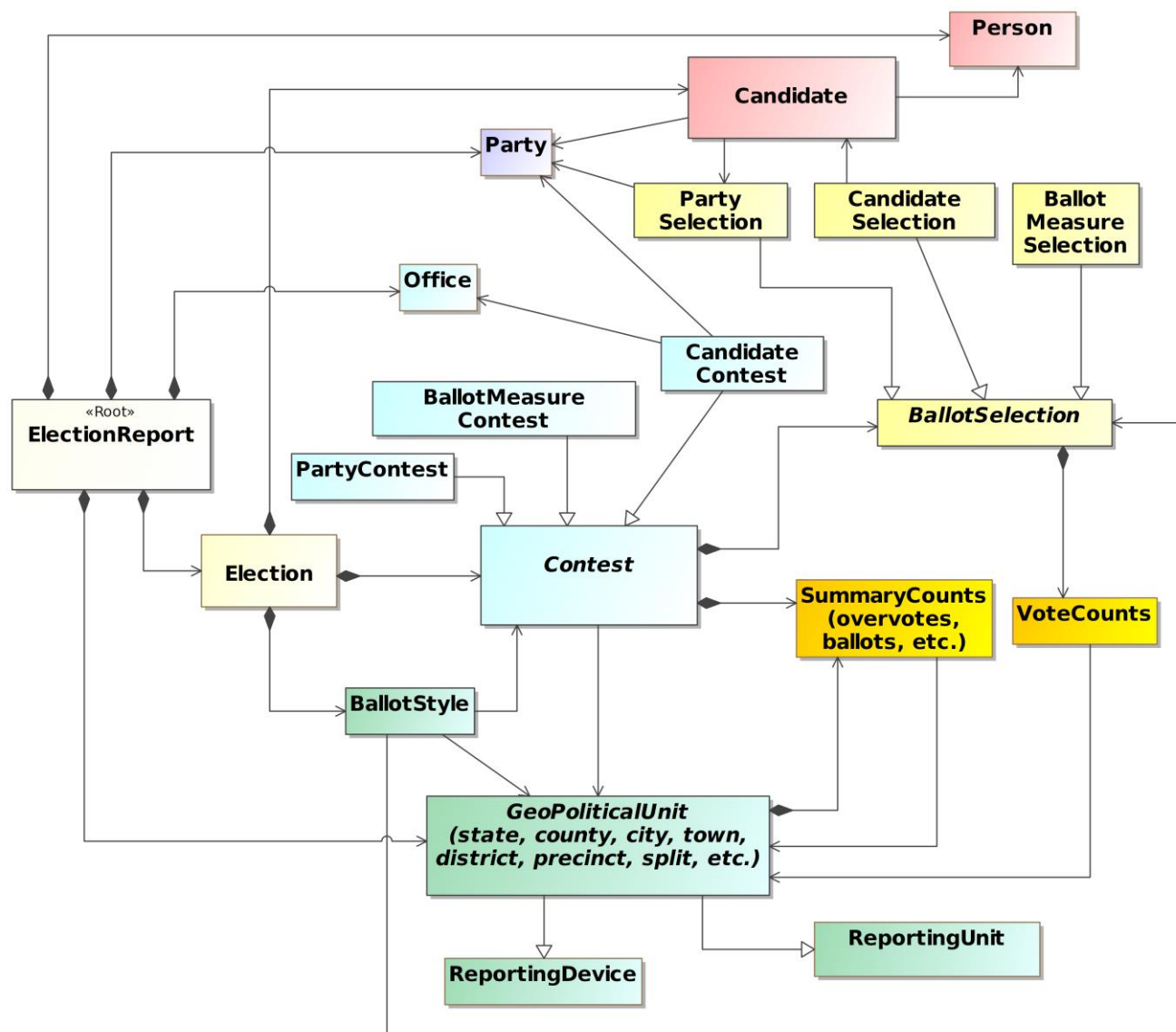


Figure 5 - Simplified ElectionResultsReporting Class Diagram

3.2.1 Major Classes in the Model

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. When XML is generated from the UML model, the classes in the model get generated as major XML elements, e.g., the UML

Candidate class results in the XML *<Candidate>* element. The major classes shown in Figure 5 are:

- **ElectionReport** – the root class, includes attributes describing the type of report, when generated, etc.
- **Election** – for a specific election, includes attributes describing the type of election, date, etc.
- **GeoPoliticalUnit** – for describing units of geo-political geography so that they can be associated with contests, offices, ballot styles, and election results (note that in the XML schema, GeoPoliticalUnit is called *GpUnit* for brevity). There are two major types of GeoPoliticalUnits:
 1. **ReportingUnit** – for describing jurisdictions, districts, precincts, and other units for which election results can be associated.
 2. **ReportingDevice** – for describing types of voting devices or specific voting devices for which election results can be associated.
- **Office** – for describing political offices that are associated with contests and electoral districts and current office holders
- **Party** – for describing political parties associated with the reporting jurisdiction and for associating the parties with candidates and contests
- **Person** – for including address and contact information for persons associated with the reporting 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 ballot selections and election results. There are four types of Contests:
 1. **CandidateContest** – for contests involving candidates
 2. **BallotMeasureContest** – for contests involving ballot measures
 3. **PartyContest** – for straight-party selection
 4. **RetentionContest** – for judicial or other types of retention contests (not shown in Figure 5)
- **Candidate** – for describing candidates so that they can be associated with contests
- **BallotSelection** – for describing the types of ballot selections in a contest and associating them with election results
 1. **BallotMeasureSelection** – used if the ballot selection is for a ballot measure
 2. **CandidateSelection** – used if the ballot selection is for a candidate
 3. **PartySelection** – used if the ballot selection is for a party as in straight party selections
- **VoteCounts** – used for the vote counts associated with a ballot selection
- **SummaryCounts** – for describing summaries of vote and ballot counts associated with a contest or with geo-political units such as a precinct
- **BallotStyle** – for describing the contests and ballot selections on a ballot and linking them to geo-political units such as precincts

The attributes associated with the classes are described above but not in detail, since the attributes correspond exactly to the XML schema's attributes and sub-elements associated with the major elements. Chapter 4 describes in detail the major XML elements and their associated

attributes and elements.

3.2.2 Relationships between Classes

From the previous section, the major classes in the UML model result in major elements in the XML schema, and the different types of relationships between the UML classes determine how the XML elements are structured in the schema. There are 3 types of relationships between classes in the class diagram:

Directed Composition: see Figure 6, ElectionReport and Election, read as, an election report is composed of elections. In the XML schema, the `<Election>` element will be generated as a sub-element of the `<ElectionReport>` element. A directed composition relationship has a closed diamond at one end and an arrow pointing to the composing class:



Figure 6 - Directed Composition Example

Is a Type of or Instance of: see Figure 7, 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; the names of abstract classes are shown in italics, e.g., *Contest*. In the XML schema, Contest will be generated as an abstract element and serve as an extension base to the `<CandidateContest>` element. An instance relationship has an open triangle at one end, pointing from the concrete class to the abstract class:



Figure 7 - Is a Type of Example

Directed Association: see Figure 8, 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 (this is explained in more detail in section **Error! Reference source not found.**). 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 8 - Directed Association Example

The generation of XML elements from the UML model is also discussed more specifically as it relates to XML in section 5.1.

For the directed composition associations in Figure 5, one can see that ElectionReport is composed of Election, GeoPoliticalUnit, Office, Party, and Person. Election is composed of BallotStyle, Candidate, and Contest. Contest is composed of BallotSelection and SummaryCounts. BallotSelection is composed of VoteCounts. GeoPoliticalUnit is also composed of SummaryCounts.

For the “is a type of” relationships in Figure 5, one can see that BallotMeasureContest, CandidateContest, and PartyContest are all types of Contest. ReportingDevice and ReportingUnit are types of GeoPoliticalUnits. BallotMeasureSelection, CandidateSelection, and PartySelection are all types of BallotSelection.

Lastly, for the directed associations in Figure 5, 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. The XML generation is more complicated for directed associations and is discussed in greater detail in section 5.1.1.

4 ElectionResultsReporting XML Schema

This chapter contains documentation and discussion of the features included in the ElectionResultsReporting XML schema.

In the sections below, an element or an enumeration name is denoted using italics and angle brackets, e.g., *<ElectionReport>* or *<ReportingUnitType>*. Attributes and enumeration values are in italics, e.g., *label* or *geo-json*. An element is sometimes referred to as a “sub-element” when it is included in another element, e.g., *<Election>* is a sub-element of *<ElectionReport>*. “Includes” is used to denote that an element contains another element as a sub-element, e.g., *<ElectionReport>* includes *<Election>*. “References” is used to denote one element linking to a second element via the second element’s *objectId* attribute, e.g., *<Candidate>* references *<Party>* using the *<PartyId>* sub-element.

4.1 Schema Stylistic Conventions

The ElectionResultsReporting schema was written observing the following stylistic conventions:

- Element, attribute, enumeration, and primitive names observe variations of “CamelCode” conventions, that is:
 - Element and enumeration and primitive names begin with a capital letter and names that consist of multiple words are concatenated and each word begins with a capital letter, thus “CamelCode”. For example, *<ElectionReport>* or *<BallotStyle>*.
 - Attribute names begin with a non-capital (lower-case) letter and names that consist of multiple words are concatenated, with the first word beginning in a non-capital letter and subsequent words beginning in a capital letter, thus “camelCode”. For example, *objectId*.
 - Enumeration value names are in non-capital letters, and names that consist of multiple words are separated by hyphens. For example, *1-of-n* or *absentee-in-person*.
- Element, attribute, and enumeration value ordering is generally alphabetical, with the following exceptions:
 - Element names such as *<StartDate>* are followed by *<EndDate>*, or *<Type>* is followed by *<OtherType>*.
 - The *objectId* attribute comes first, followed by the other attributes in alphabetical order.
 - If there is an enumeration value of *other*, it comes last in the list of values.

4.2 Elements and Types

The following sections deal with major elements and types in the schema. Those types in the schema that are used as elements are referred to as elements, whereas types that are used only as types are referred to as types.

4.2.1 The <AnnotatedString> Type

<AnnotatedString> is used as a type for character strings; it adds a 16-character annotation to a character string, for example:

```
<Email annotation="Work">john.a.smith@26003.org</Email>
<Phone annotation="Cell">304-243-1234</Phone>
```

Table 4.1 – Attributes for <AnnotatedString>

Attribute	Required	Type	Attribute Description
annotation	no	ShortString	An annotation of up to 16 characters associated with a character string.

Schema Definition:

```
<xsd:complexType name="AnnotatedString">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="annotation" type="ShortString"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

4.2.2 The <BallotSelection> Element and Extension Base

Used for the ballot selections in a contest (e.g., for candidates, for ballot measures) and to generally link them to vote counts. <Contest> includes <BallotSelection>.

<BallotSelection> is an abstract element with three xsi:types that get used according to the type of contest:

- <BallotSelection xsi:type="BallotMeasureSelection">, used if the contest type is for a ballot measure, including for retentions (see section 4.2.2.1)
- <BallotSelection xsi:type="CandidateSelection">, used if the contest type is for one or more candidates, to link the ballot selection to the candidate elements and endorsement parties (see section 4.2.2.2)
- <BallotSelection xsi:type="PartySelection">, used if the contest type is for a party, e.g., for a straight party contest (see section 4.2.2.3)

<BallotSelection> includes <VoteCounts> for associating vote counts with the ballot selection. <BallotSelection> “wraps” occurrences of <VoteCounts> elements in a container element for the purpose of making large instance files easier to manipulate in XML viewers and editors.

<SequenceOrder> is included to specify an ordering for the ballot selections for purposes of display. However, this ordering cannot be guaranteed to be the same order as is on *all* ballots originally if ballot rotation was used. The original ballot ordering can be preserved, however, by using the <BallotStyle> element (see section 4.2.3).

Table 4.2 – Attributes for <BallotSelection>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.3 – Elements for <BallotSelection>

Element	Multiplicity	Type	Element Description
<SequenceOrder>	0 or 1	xsd:integer	Order in which the candidate is listed on the ballot for purposes of results display. If not present, no order is assumed.
<VoteCounts>	1 or more	VoteCounts	For associating votes with a ballot selection in a contest.
<VoteCountsCollection>	0 or more		Wrapper for collections of <VoteCounts> elements.

Schema definition:

```
<xsd:complexType name="BallotSelection" abstract="true">
  <xsd:sequence>
    <xsd:element name="SequenceOrder" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="VoteCountsCollection" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="VoteCounts" type="VoteCounts"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
```

4.2.2.1 The *xsi:type* **BallotMeasureSelection**

For a ballot 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 is an *xsi:type* of *<BallotSelection>* and therefore inherits its attributes and elements (see Table 4.2 and Table 4.3). Its syntax is:

```
<BallotSelection xsi:type="BallotMeasureSelection" ... />
```

Table 4.4 – Elements for *<BallotSelection xsi:type="BallotMeasureSelection">*

Element	Multiplicity	Type	Element Description
<i><Selection></i>	1	InternationalizedText	Contains the text used to indicate a vote for or against the ballot measure, e.g., "yes", "no".

Schema definition:

```
<xsd:complexType name="BallotMeasureSelection">
  <xsd:complexContent>
    <xsd:extension base="BallotSelection">
      <xsd:sequence>
        <xsd:element name="Selection" type="InternationalizedText"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.2.2 The *xsi:type* **CandidateSelection**

For the ballot selections in a candidate contest, including for write-ins. It is an *xsi:type* of *<BallotSelection>* and therefore inherits its attributes and elements (see Table 4.2 and Table 4.3). References to multiple *<Candidate>* elements can be included if necessary, e.g., when the ballot selection would be for a ticket of candidates (unless the ticket itself is defined as a candidate). Its syntax is:

```
<BallotSelection xsi:type="CandidateSelection" ... />
```

<EndorsementPartyId> 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 *<EndorsementPartyId>* 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 occurrences of *<CandidateId>* according to the ordering scheme in place.

Table 4.5 – Elements for *<BallotSelection xsi:type="CandidateSelection">*

Element	Multiplicity	Type	Element Description
<i><CandidateId></i>	1 or more	xsd:IDREF	Unique identifier for a <i><Candidate></i> element. For associating a candidate with the candidate selection on the ballot. The multiplicity is unlimited for cases where the ballot selection is for multiple candidates, e.g., a ticket.
<i><EndorsementPartyId></i>	0 or more	xsd:IDREF	Unique identifier for a <i><Party></i> element. For associating one or more endorsing parties with the candidate selection.
IsWriteIn	0 or 1	xsd:boolean	Indicates whether the candidate is a write-in, e.g., “yes” or “no”. Assumed to be “no” if not present.

Schema definition:

```
<xsd:complexType name="CandidateSelection">
  <xsd:complexContent>
    <xsd:extension base="BallotSelection">
      <xsd:sequence>
        <xsd:element name="CandidateId" type="xsd:IDREF"
maxOccurs="unbounded"/>
        <xsd:element name="EndorsementPartyId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
        <xsd:element name="IsWriteIn" type="xsd:boolean" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.2.3 The xsi:type *PartySelection*

For a ballot selection involving a party such as for a straight party selection on the ballot. It is an xsi:type of *<BallotSelection>* and therefore inherits its attributes and elements (see Table 4.2 and Table 4.3). Its syntax is:

```
<BallotSelection xsi:type="PartySelection" ... />
```

Table 4.6 – Elements for *<BallotSelection xsi:type="PartySelection">*

Element	Multiplicity	Type	Element Description
<i><PartyId></i>	1 or more	xsd:IDREF	Unique identifier for a <i><Party></i> element. For associating one or more parties with the party selection.

Schema definition:

```
<xsd:complexType name="PartySelection">
  <xsd:complexContent>
    <xsd:extension base="BallotSelection">
      <xsd:sequence>
        <xsd:element name="PartyId" type="xsd:IDREF" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.3 The <BallotStyle> and <OrderedContest> Elements

For defining a ballot style composed of contests and their ballot selections, and associating the ballot style with a political party, a reference to an image of the ballot, and a reference to the a precinct or other geo-political unit that the ballot is unique to. <Election> includes <BallotStyle>.

<BallotStyle> references <OrderedContest> to include contests on that ballot style and the ballot selections associated with each contest (see section 4.2.3.1). 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 geo-political unit.

<BallotStyle> includes <GpUnitId> to reference a <GpUnit> element defined for the associated precinct or split precinct. If the ballot style is associated with multiple precincts (or other geographies), multiple references can be used, e.g.,

```
<BallotStyle>
  <GpUnitId>Precinct5</GpUnitId>
  <GpUnitId>Precinct6</GpUnitId>
  <GpUnitId>Precinct7</GpUnitId>
  <OrderedContest ... />
  ...
</BallotStyle>
```

When including <ExternalIdentifiers>, if the type is not listed in enumeration <IdentifierType>, use *other* and include the type (that is not listed in the enumeration) in *OtherType*.

Table 4.7 – Attributes for <BallotStyle>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.8 – Elements for <BallotStyle>

Element	Multiplicity	Type	Element Description
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID with the ballot style.
<GpUnitId>	1 or more	xsd:IDREF	Unique identifier for a <GpUnit> element. For associating a specific geo-political unit with the ballot style.
<ImageUri>	0 or more	xsd:anyURI	URI for a ballot image.

Element	Multiplicity	Type	Element Description
<OrderedContest>	0 or more	OrderedContest	For associating contests in the order as listed on ballot.
<PartyId>	0 or more	xsd:IDREF	Unique identifier for a <Party> element. For associating a party with the ballot style.

Schema definition:

```

<xsd:complexType name="BallotStyle">
  <xsd:sequence>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
    <xsd:element name="GpUnitId" type="xsd:IDREF" maxOccurs="unbounded"/>
    <xsd:element name="ImageUri" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="OrderedContest" type="OrderedContest" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>

```

4.2.3.1 The <OrderedContest> Element

<OrderedContest> associates a reference to a contest with references to the contest's ballot selections. <BallotStyle> includes <OrderedContest> for each contest on the ballot. To preserve any rotation associated with the ballot, it is expected that the generating application will list the occurrences of <OrderedBallotSelectionId> in the order as on the ballot for the associated geo-political unit.

Table 4.9 – Elements for <OrderedContest>

Element	Multiplicity	Type	Element Description
<ContestId>	1	xsd:IDREF	Unique identifier for a <Contest> element. For associating a contest on the ballot.
<OrderedBallotSelectionId>	0 or more	xsd:IDREF	Unique identifier for a <BallotSelection> element. For associating ballot selections with the contest.

Schema definition:

```
<xsd:complexType name="OrderedContest">
  <xsd:sequence>
    <xsd:element name="ContestId" type="xsd:IDREF"/>
    <xsd:element name="OrderedBallotSelectionId" type="xsd:IDREF" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

4.2.4 The <Candidate> Element

For defining information about a candidate in a contest. `<BallotSelection xsi:type="CandidateSelection">` references `<Candidate>` elements to associate one or more candidates with a ballot selection. `<Election>` includes `<Candidate>`.

`<Candidate>` uses `<PartyId>` 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 `<BallotSelection xsi:type="Candidate">` sub-element (see section 4.2.2.2).

`<ExternalIdentifiers>` can be used to associate an ID 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*.

Table 4.10 - Attributes for <Candidate>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.11 - Elements for <Candidate>

Element	Multiplicity	Type	Element Description
<BallotName>	1	InternationalizedText	For the candidate's name as listed on the ballot.
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating codes with the candidate.
<FileDate>	0 or 1	xsd:date	Date when the candidate filed for the contest.
<IsIncumbent>	0 or 1	xsd:boolean	Boolean to indicate whether the candidate is the incumbent for the office associated with the contest. Assumed to be "no" if not present.
<IsTopTicket>	0 or 1	xsd:boolean	Boolean to indicate whether the

Element	Multiplicity	Type	Element Description
			candidate is the top of a ticket that includes multiple candidates. Assumed to be “no” if not present.
<PartyId>	0 or 1	xsd:IDREF	Unique identifier for a <Party> element. For associating a party with the candidate.
<PersonId>	0 or 1	xsd:IDREF	Unique identifier for a <Person> element. For associating more detailed information about the candidate.
<PostElectionStatus>	0 or 1	CandidatePostElectionStatus	Final status of the candidate, e.g., winner, withdrawn, etc.
<PreElectionStatus>	0 or 1	CandidatePreElectionStatus	Registration status of the candidate, e.g., filed, qualified, etc.

Schema definition:

```

<xsd:complexType name="Candidate">
  <xsd:sequence>
    <xsd:element name="BallotName" type="InternationalizedText"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
    <xsd:element name="FileDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="IsIncumbent" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="IsTopTicket" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"/>
    <xsd:element name="PersonId" type="xsd:IDREF" minOccurs="0"/>
    <xsd:element name="PostElectionStatus" type="CandidatePostElectionStatus"
minOccurs="0"/>
    <xsd:element name="PreElectionStatus" type="CandidatePreElectionStatus"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>

```

4.2.5 The <ContactInformation> Element

For defining contact information about objects such as persons, boards of authorities, organizations, etc. <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.

<Email>, <Fax>, and <Phone> are of type *AnnotatedString*, which permits up to a 16-character annotation to be associated with the data, for example:

```
<Email annotation="Work">john.a.smith@26003.org</Email>
<Email annotation="Home">john.a.smith@26003.org</Email>
```

Table 4.12 - Attributes for <ExternalIdentifiers>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema [6].

Table 4.13 - Elements for <ContactInformation>

Element	Multiplicity	Type	Element Description
<AddressLine>	0 or more	xsd:string	For associating an address with the contact.
<Email>	0 or more	AnnotatedString	Email address associated with the contact.
<Fax>	0 or more	AnnotatedString	Fax number associated with the contact.
<Name>	0 or 1	xsd:string	Name associated with the contact.
<Phone>	0 or more	AnnotatedString	Phone number associated with the contact.
<Schedule>	0 or more	Schedule	For associating a schedule with the contact.
<Uri>	0 or more	xsd:anyURI	URI associated with the contact.

Schema definition:

```
<xsd:complexType name="ContactInformation">
```

```
<xsd:sequence>
  <xsd:element name="AddressLine" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Email" type="AnnotatedString" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Fax" type="AnnotatedString" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
  <xsd:element name="Phone" type="AnnotatedString" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Schedule" type="Schedule" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Uri" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
```

4.2.6 The <Contest> Element and Extension Base

For defining a contest and linking the contest to the associated candidates or ballot measures. <Election> includes <Contest>.

<Contest> is an abstract element with four xsi:types that get used according to the type of contest:

- <Contest xsi:type="BallotMeasureContest">, used for a contest involving a ballot measure (see section 4.2.6.1)
- <Contest xsi:type="CandidateContest">, used for a contest involving one or more candidates for an office (see section 4.2.6.2)
- <Contest xsi:type="PartyContest">, used for a contest for a straight party selection on the ballot (see section 4.2.6.3)
- <Contest xsi:type="RetentionContest">, used for a judicial or other type of retention contest (see section 4.2.6.4)

Each xsi:type is used to include information for that specified type of contest, for example, <Contest xsi:type="CandidateContest"> is used to include information about a candidate contest including the associated office and the number of candidates that can be voted for.

<Contest> includes <BallotSelection> to link the selections on the ballot to the contest, e.g., to link one or more candidates to a candidate contest. Like <Contest>, <BallotSelection> is also an abstract element and has xsi types that essentially correspond to those of <Contest>, as follows:

- <Contest xsi:type="BallotMeasureContest"> includes <BallotSelection xsi:type="BallotMeasureSelection"> (see section 4.2.2.1)
- <Contest xsi:type="CandidateContest"> includes <BallotSelection xsi:type="CandidateSelection"> (see section 4.2.2.2)
- <Contest xsi:type="PartyContest"> includes <BallotSelection xsi:type="PartySelection"> (see section 4.2.2.3)
- <Contest xsi:type="RetentionContest"> includes <BallotSelection xsi:type="BallotMeasureSelection"> (see section 4.2.2.1)

<Contest> includes a required <ElectoralDistrictId> reference to a <GpUnit> defined for the geographical scope of the contest. For example, in a state senate contest, <ElectoralDistrictId> would reference a <GpUnit xsi:type="ReportingUnit"> element defined for the district associated with the contest. <Office> also includes an optional reference that serves the same purpose.

<Contest> includes <SummaryCounts> for providing a summary of miscellaneous counts associated with the contest, including total number of ballots cast containing the contest, total number of overvotes, undervotes, or write-ins. The summary counts can be associated with the contest as a whole, or with precincts or other lower-level reporting units by using multiple occurrences of <SummaryCounts> (see sections 4.2.8 and 4.2.23).

`<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 geo-political units (see section 4.2.3).

When including `<ExternalIdentifiers>`, if the type is not listed in enumeration `<IdentifierType>`, use *other* and include the type (that is not listed in the enumeration) in `<OtherType>`.

Table 4.14 – Attributes for `<Contest>`

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.15 – Elements for `<Contest>`

Element	Multiplicity	Type	Element Description
<code><Abbreviation></code>	0 or 1	xsd:string	Abbreviation for the contest.
<code><BallotSelection></code>	0 or more	BallotSelection	For associating a ballot selection for the contest, i.e., a candidate, a ballot measure.
<code><BallotSubTitle></code>	0 or 1	InternationalizedText	Subtitle of the contest as it appears on the ballot.
<code><BallotTitle></code>	0 or 1	InternationalizedText	Title of the contest as it appears on the ballot.
<code><CountStatus></code>	0 or more	CountStatus	For providing various counting status associated with the contest.
<code><ElectoralDistrictId></code>	1	xsd:IDREF	Unique identifier for a <code><GpUnit></code> element. For associating the contest with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc.
<code><ExternalIdentifiers></code>	0 or 1	ExternalIdentifiers	For associating an ID with the contest.
<code><HasRotation></code>	0 or 1	xsd:boolean	Boolean to indicate whether

Element	Multiplicity	Type	Element Description
			the selections in the contest are rotated. Assumed to be “no” if not present.
<Name>	1	xsd:string	Name of the contest, not necessarily as it appears on the ballot (use <BallotTitle> for this).
<SequenceOrder>	0 or 1	xsd:integer	Ordering for listing the contest for purposes of results display. If not present, no order is assumed.
<SubUnitsReported>	0 or 1	xsd:integer	Number of subunits, e.g., precincts, that have completed reporting votes for this contest.
<SummaryCounts>	0 or more	SummaryCounts	For providing a summary of miscellaneous counts associated with the contest.
<TotalSubUnits>	0 or 1	xsd:integer	Total number of subunits, e.g., precincts that have this contest on the ballot.
<VoteVariation>	0 or 1	VoteVariation	Vote variation associated with the contest, e.g., <i>n-of-m</i> .
<OtherVoteVariation>	0 or 1	xsd:string	For use when <VoteVariation> is <i>other</i> .

Schema Definition:

```

<xsd:complexType name="Contest" abstract="true">
  <xsd:sequence>
    <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
    <xsd:element name="BallotSelection" type="BallotSelection" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element name="BallotSubTitle" type="InternationalizedText"
minOccurs="0"/>
    <xsd:element name="BallotTitle" type="InternationalizedText" minOccurs="0"/>
    <xsd:element name="CountStatus" type="CountStatus" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ElectoralDistrictId" type="xsd:IDREF"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
    <xsd:element name="HasRotation" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="SequenceOrder" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="SubUnitsReported" type="xsd:integer" minOccurs="0"/>

```

```
<xsd:element name="SummaryCounts" type="SummaryCounts" minOccurs="0"
  maxOccurs="unbounded"/>
<xsd:element name="TotalSubUnits" type="xsd:integer" minOccurs="0"/>
<xsd:element name="VoteVariation" type="VoteVariation" minOccurs="0"/>
<xsd:element name="OtherVoteVariation" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
```

4.2.6.1 The *xsi:type* **BallotMeasureContest**

For ballot measure (i.e., referenda or a tax measure) and judicial retention contests. It is an *xsi:type* of *<Contest>* and therefore inherits its attributes and elements (see Table 4.14 and Table 4.15). Its syntax is:

```
<Contest xsi:type="BallotMeasureContest" ... />
```

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>*.

Table 4.16 – Elements for *<Contest xsi:type="BallotMeasureContest">*

Element	Multiplicity	Type	Element Description
<i><ConStatement></i>	0 or 1	InternationalizedText	For a statement on the ballot associated with a “no” vote.
<i><EffectOfAbstain></i>	0 or 1	InternationalizedText	For a statement on the ballot detailing the effect of abstaining from voting on the ballot measure.
<i><FullText></i>	0 or 1	InternationalizedText	For full text on the ballot of the ballot measure.
<i><InfoUri></i>	0 or 1	xsd:anyURI	For associating a URI with the ballot measure contest.
<i><PassageThreshold></i>	0 or 1	InternationalizedText	For a statement on the ballot of the number or percentage of votes needed to approve or pass the ballot measure.
<i><ProStatement></i>	0 or 1	InternationalizedText	For a statement on the ballot associated with a “yes” vote.
<i><SummaryText></i>	0 or 1	InternationalizedText	For a summary on the ballot of the ballot measure.
<i><Type></i>	0 or 1	BallotMeasureType	For indicating the type of ballot measure.
<i><OtherType></i>	0 or 1	xsd:string	Used when <i>BallotMeasureType</i> is <i>other</i> .

Schema Definition:

```
<xsd:complexType name="BallotMeasureContest">
  <xsd:complexContent>
    <xsd:extension base="Contest">
      <xsd:sequence>
        <xsd:element name="ConStatement" type="InternationalizedText"
minOccurs="0"/>

```

```

        <xsd:element name="EffectOfAbstain" type="InternationalizedText"
minOccurs="0"/>
        <xsd:element name="FullText" type="InternationalizedText"
minOccurs="0"/>
        <xsd:element name="InfoUri" type="xsd:anyURI" minOccurs="0"/>
        <xsd:element name="PassageThreshold" type="InternationalizedText"
minOccurs="0"/>
        <xsd:element name="ProStatement" type="InternationalizedText"
minOccurs="0"/>
        <xsd:element name="SummaryText" type="InternationalizedText"
minOccurs="0"/>
        <xsd:element name="Type" type="BallotMeasureType" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

4.2.6.2 The *xsi:type* **CandidateContest**

For a contest that involves selecting one or more candidates. It is an *xsi:type* of *<Contest>* and therefore inherits its attributes and elements (see Table 4.14 and Table 4.15). Its syntax is:

```
<Contest xsi:type="CandidateContest" ... />
```

This element optionally references *<Office>* and *<Party>*. For cases when the contest is associated with multiple offices, e.g., if Governor and Lt. Governor are both separate offices, it is expected that the generating application will list the multiple references to *<Office>* according to any ordering scheme in place.

Table 4.17 – Elements for *<Contest xsi:type="CandidateContest">*

Element	Multiplicity	Type	Element Description
<i><NumberElected></i>	0 or 1	<i>xsd:integer</i>	Number of candidates that are elected in the contest ("N" of N-of-M).
<i><OfficeId></i>	0 or more	<i>xsd:IDREF</i>	Unique identifier for an <i><Office></i> element. For associating an office description with the contest.
<i><PrimaryPartyId></i>	0 or 1	<i>xsd:IDREF</i>	Unique identifier for a <i><Party></i> element. For associating a party with the contest.
<i><VotesAllowed></i>	1	<i>xsd:integer</i>	Maximum number of votes/write-ins per voter in this contest.

Schema Definition:

```
<xsd:complexType name="CandidateContest">
  <xsd:complexContent>
    <xsd:extension base="Contest">
      <xsd:sequence>
        <xsd:element name="NumberElected" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="PrimaryPartyId" type="xsd:IDREF" minOccurs="0"/>
        <xsd:element name="VotesAllowed" type="xsd:integer"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.6.3 The *xsi:type PartyContest*

For a contest that involves choosing a party, typically for a straight party selection on the ballot. It is an *xsi:type* of *<Contest>*. Its syntax is:

```
<Contest xsi:type="PartyContest" ... />
```

Schema definition:

```
<xsd:complexType name="PartyContest">  
  <xsd:complexContent>  
    <xsd:extension base="Contest"/>  
  </xsd:complexContent>  
</xsd:complexType>
```

4.2.6.4 The *xsi:type 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 `<Contest xsi:type="BallotMeasureContest">` in that it can include a reference to a candidate or the associated office.

Besides being an *xsi:type* of `<Contest>`, this element uses `<BallotMeasureContest>` as an extension base. Therefore, it inherits attributes and elements from BOTH elements (see Table 4.14 and Table 4.15 for `<Contest>` and Table 4.16 for `<Contest xsi:type="BallotMeasure">`). Its syntax is:

```
<Contest xsi:type="RetentionContest" ... />
```

Table 4.18 – Elements for `<Contest xsi:type="RetentionContest">`

Element	Multiplicity	Type	Element Description
<code><CandidateId></code>	1	<code>xsd:IDREF</code>	Unique identifier for a <code><Candidate></code> element. For associating a candidate with the retention contest.
<code><OfficeId></code>	0 or 1	<code>xsd:IDREF</code>	Unique identifier for an <code><Office></code> element. For associating an office description with the retention contest.

Schema Definition:

```
<xsd:complexType name="RetentionContest">
  <xsd:complexContent>
    <xsd:extension base="BallotMeasureContest">
      <xsd:sequence>
        <xsd:element name="CandidateId" type="xsd:IDREF"/>
        <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.7 The <CountStatus> Element

For reporting on the counting status for various items such as ballot types or write-ins, e.g., whether for a certain type of ballot, the counts are in progress, not yet started, complete, etc. <Contest>, <Election>, and <GpUnit> include <CountStatus>.

If the type of count item is not listed in enumeration <CountItemType>, use *other* and include the type (that is not listed in the enumeration) in <OtherType>.

Table 4.19 – Elements for <CountStatus>

Element	Multiplicity	Type	Element Description
<Status>	1	CountItemStatus	The status of the count, from the <CountItemStatus> enumeration.
<Type>	1	CountItemType	The type of item, from the <CountItemType> enumeration.
<OtherType>	0 or 1	xsd:string	Used when <Type> is <i>other</i> .

Schema Definition:

```
<xsd:complexType name="CountStatus">
  <xsd:sequence>
    <xsd:element name="Status" type="CountItemStatus"/>
    <xsd:element name="Type" type="CountItemType"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```


4.2.8 The <Counts> Extension Base Element

<Counts> is not used or referenced directly; it is used only as an extension base for the following elements:

- <SummaryCounts>, for reporting on contest and geo-political unit summary vote counts (see section 4.2.23)
- <VoteCounts>, for reporting on contest vote counts (see section 4.2.24)

These elements inherit the attributes and elements of <Counts>.

<Counts> provides the capability to filter vote counts by device characteristics or by various types of ballots or write-ins, or to link the counts to a <GpUnit> element defined for, e.g., a precinct.

The <IsSuppressedForPrivacy> boolean is used for cases when a jurisdiction has a requirement for suppressing a contest's vote count when associated with a precinct and the count is so small as to potentially show how specific voters at the precinct's polling place may have voted (typically under 10 votes). This is also a potential concern when filtering on device type or ballot type for a precinct; the accessible voting device will typically have relatively low counts. In some cases, jurisdictions suppress the votes for that contest/precinct/device/ballot type combination, but add them back to the contest totals, thus meeting the requirement for voter privacy. However, this results in the precinct counts not adding up to the aggregated totals, thus this boolean can be used to indicate which of the counts were suppressed.

If the type of count item is not listed in enumeration <CountItemType>, use *other* and include the type (that is not listed in the enumeration) in <OtherType>.

Table 4.20 – Elements for <Counts>

Element	Multiplicity	Type	Element Description
<Device>	0 or 1	Device	For filtering the vote counts by device type.
<GpUnitId>	0 or 1	xsd:IDREF	Unique identifier for a <GpUnit> element. For associating counts with a geo-political unit, e.g., a precinct, a county, a township, etc.
<IsSuppressedForPrivacy>	0 or 1	xsd:boolean	Boolean to indicate if votes are suppressed for voter privacy, e.g., "yes" or "no". Assumed to be "no" if not present.
<Type>	0 or 1	CountItemType	The type of count being used as a filter on the vote counts, e.g., election day, early voting, etc.
<OtherType>	0 or 1	xsd:string	Used when <Type> is <i>other</i> .

Schema Definition:

```
<xsd:complexType name="Counts" abstract="true">
  <xsd:sequence>
    <xsd:element name="Device" type="Device" minOccurs="0"/>
    <xsd:element name="GpUnitId" type="xsd:IDREF" minOccurs="0"/>
    <xsd:element name="IsSuppressedForPrivacy" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="Type" type="CountItemType" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

4.2.9 The <Device> Element

For filtering vote counts by various attributes associated with devices. <GpUnit> includes <Device>, and <SummaryCounts> and <VoteCounts> include <Device> as a result of it being included from extension base <Counts>.

If manual counting of ballots is being employed, use the <DeviceType> enumeration value *manual-count*.

Table 4.21 – Elements for <Device>

Element	Multiplicity	Type	Element Description
<Manufacturer>	0 or 1	xsd:string	Manufacturer of the device.
<Model>	0 or 1	xsd:string	Manufacturer's device model, used to filter on, e.g., a specific model of DRE or other device type.
<Type>	0 or 1	DeviceType	Enumerated type of device, e.g., DRE, opscan-precinct, etc.
<OtherType>	0 or 1	xsd:string	Used when <Type> is <i>other</i> .

Schema Definition:

```

<xsd:complexType name="Device">
  <xsd:sequence>
    <xsd:element name="Manufacturer" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Model" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Type" type="DeviceType" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

4.2.10 The <Election> Element

For defining the status of the election and associated information such as candidates, contests, and vote counts.

<Election> includes the major elements that are specific to an election: <BallotStyle>, <Candidate>, and <Contest>. <Election> “wraps” occurrences of these elements in container elements for the purpose of making large instance files easier to manipulate in XML viewers and editors.

<Election> includes a required <ElectionScopeId> reference to a <GpUnit> element for the purpose of identifying the geographical scope of the election. For example, for an election within a county, <ElectionScopeId> would include a reference to a <GpUnit> defined for the county. If it is desired to include election authority information, the <GpUnit> can include <ElectionAdministration>, see sections 4.2.14.2 and 4.2.11.

Table 4.22 – Elements for <Election>

Element	Multiplicity	Type	Element Description
<BallotStyle>	1 or more	BallotStyle	For defining ballot styles associated with the election.
<BallotStyleCollection>	0 or 1		Wrapper element for <GpUnit> definitions.
<Candidate>	1 or more	Candidate	For defining candidates associated with the election.
<CandidateCollection>	0 or 1		Wrapper element for <GpUnit> definitions.
<ContactInformation>	0 or 1	ContactInformation	For associating various contact information with the election.
<Contest>	1 or more	Contest	For defining contests associated with the election.
<ContestCollection>	0 or 1		Wrapper element for <GpUnit> definitions.
<CountStatus>	0 or more	CountStatus	For providing various counting status on types of ballots or other items.
<ElectionScopeId>	1	xsd:IDREF	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.

Element	Multiplicity	Type	Element Description
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID 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	xsd:date	Calendar start date of the election, e.g., "November 4, 2014".
<EndDate>	0 or 1	xsd:date	For an election that spans a period of days.
<Type>	1	ElectionType	Enumerated type of election, e.g., <i>partisan-primary</i> , <i>open-primary</i> , etc.

Schema Definition:

```

<xsd:complexType name="Election">
  <xsd:sequence>
    <xsd:element name="BallotStyleCollection" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="BallotStyle" type="BallotStyle"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="CandidateCollection" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Candidate" type="Candidate"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ContactInformation" type="ContactInformation"
minOccurs="0"/>
    <xsd:element name="ContestCollection" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Contest" type="Contest" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="CountStatus" type="CountStatus" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ElectionScopeId" type="xsd:IDREF"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

```
<xsd:element name="Name" type="InternationalizedText"/>
<xsd:element name="StartDate" type="xsd:date"/>
<xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
<xsd:element name="Type" type="ElectionType"/>
</xsd:sequence>
</xsd:complexType>
```

4.2.11 The <ElectionAdministration> Element

Used to provide various information about an election authority. <GpUnit xsi:type="ReportingUnit"> includes <ElectionAdministration>.

The <Electon> element references the <GpUnit> defined as the scope of the election, e.g., if the scope of the election is for a county, <Election> references the <GpUnit> element defined for that county. <ElectionAdministration> would presumably be included from that <GpUnit> element defined as the scope of the election.

<ElectionAdministration> includes <ContactInformation> for the election authority and, using <ElectionOfficialPersonId>, references one or more <Person> elements defined for individuals/organizations associated with the election authority.

Table 4.23 – Elements for <Device>

Element	Multiplicity	Type	Element Description
<ContactInformation>	0 or 1	ContactInformation	For including various contact information.
<ElectionOfficialPersonId>	0 or more	xsd:IDREF	Unique identifier for a <Person> element defined for the election authority.
<Name>	0 or 1	xsd:string	Name of the election authority.

Schema Definition:

```

<xsd:complexType name="ElectionAdministration">
  <xsd:sequence>
    <xsd:element name="ContactInformation" type="ContactInformation"
minOccurs="0"/>
    <xsd:element name="ElectionOfficialPersonId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

4.2.12 The <ElectionReport> Element

The root element; for defining items pertaining to the status and format of the report and when it was generated. The optional <Signature> element is used for an XML digital signature [10]. <Signature> *must* be the last element of <ElectionReport>.

<ElectionReport> includes the major elements that are not necessarily specific to an election and that therefore can exist in a pre-election report: <GpUnit>, <Office> and <OfficeGroup>, <Party>, <Person>, and <Election>. As with <Election>, <ElectionReport> “wraps” occurrences of the elements (excepting <Election>) in container elements for the purpose of making the instance files easier to manipulate in XML viewers and editors.

When a particular ordering of items such as offices or political parties are important to preserve, it is expected that the generating application will define those elements according to any ordering scheme in place.

Table 4.24 – Elements for <ElectionReport>

Element	Multiplicity	Type	Element Description
<Election>	0 or more	Election	For associating elections with the report.
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID with the report.
<Format>	1	ReportDetailLevel	Detail level of the report, e.g., contest summary, precinct level results, etc.
<GeneratedDate>	1	xsd:dateTime	Identifies the date and time that the election report was generated.
<GpUnit>	1 or more	GpUnit	For associating geo-political units with the report.
<GpUnitCollection>	0 or 1		Wrapper element for <GpUnit> definitions.
<Issuer>	1	xsd:string	Identification of the report issuer.
<IssuerAbbreviation>	1	xsd:string	An abbreviation of the report issuer such as the 2-character U.S. Census Bureau abbreviation of the state whose results are being reported, e.g., AL, TX, MN, etc.
<IsTest>	0 or 1	xsd:boolean	Used to indicate whether the report is a test report. Assumed to be “no” if not present.

Element	Multiplicity	Type	Element Description
<Notes>	0 or 1	xsd:string	For including an arbitrary message with the report.
<Office>	1 or more	Office	For associating offices with the report.
<OfficeCollection>	0 or 1		Wrapper element for <Office> definitions.
<OfficeGroup>	0 or more	OfficeGroup	For associating a name for a grouping of offices with the report.
<Party>	1 or more	Party	For associating parties with the report.
<PartyCollection>	0 or 1		Wrapper element for <Party> definitions.
<Person>	1 or more	Person	For associating persons with the report.
<PersonCollection>	0 or 1		Wrapper element for <Person> definitions.
<SequenceStart>	1	xsd:integer	If this report is part of a sequence of files, its number in the sequence.
<SequenceEnd>	1	xsd:integer	Indicates the upper bound of the sequence.
<Status>	1	ResultsStatus	Status of the election report, e.g., test mode, unofficial, etc.
<TestType>	0 or 1	xsd:string	A description of the type of test, e.g., pre-election, logic and accuracy, etc.
<VendorApplicationID>	1	xsd:string	An identifier of the vendor application generating the election report, e.g., X-EMS version 3.1.a.
<Signature>	0 or 1	Signature	Reference to the <Signature> element of the W3C digital signature schema imported into this schema.

Schema Definition:

```

<xsd:complexType name="ElectionReport">
  <xsd:sequence>
    <xsd:element name="Election" type="Election" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="Format" type="ReportDetailLevel"/>
<xsd:element name="GeneratedDate" type="xsd:dateTime"/>
<xsd:element name="GpUnitCollection" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="GpUnit" type="GpUnit" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Issuer" type="xsd:string"/>
<xsd:element name="IssuerAbbreviation" type="xsd:string"/>
<xsd:element name="IsTest" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="Notes" type="xsd:string" minOccurs="0"/>
<xsd:element name="OfficeCollection" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Office" type="Office" maxOccurs="unbounded"/>
      <xsd:element name="OfficeGroup" type="OfficeGroup" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="PartyCollection" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Party" type="Party" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="PersonCollection" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Person" type="Person" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="SequenceStart" type="xsd:integer"/>
<xsd:element name="SequenceEnd" type="xsd:integer"/>
<xsd:element name="Status" type="ResultsStatus"/>
<xsd:element name="TestType" type="xsd:string" minOccurs="0"/>
<xsd:element name="VendorApplicationId" type="xsd:string"/>
<xsd:element ref="ds:Signature" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>

```

4.2.13 The <ExternalIdentifiers Element>

For associating a jurisdiction's codes, i.e., identifiers, with objects such as candidates, offices, or geo-political units such as counties, towns, precincts, etc.¹ Multiple occurrences of the <ExternalIdentifier> sub-element 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 [11]) codes as well as OCD-IDs (Open Civic Data Identifiers [12]), as follows:

```
<ExternalIdentifiers >
  <ExternalIdentifier>
    <Type>fips</Type>
    <Value>54</Value>
  <ExternalIdentifier>
    <Type>ocd-id</Type>
    <Value>ocd-division/country:us/state:mv</Value>
  <ExternalIdentifier>
  </ExternalIdentifiers>
```

For elements that include <ExternalIdentifiers>, if the type is not listed in enumeration <IdentifierType>, use *other* and include the type (that is not listed in the enumeration) in *OtherType*, e.g.,

```
<ExternalIdentifier>
  <Type>other</Type>
  <Value>101-A</Value>
  <OtherType>Ohio County Precinct Numbers</OtherType>
</ExternalIdentifier>
```

Table 4.25 - Attributes for <ExternalIdentifiers>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.26 - Elements for <ExternalIdentifiers>

Element	Multiplicity	Type	Element Description
<Type>	1	IdentifierType	An identifier type, e.g., FIPS.
<OtherType>	0 or 1	xsd:string	Used when <IdentifierType> value is <i>other</i> .
<Value>	1	xsd:string	The identifier used by the jurisdiction.

¹ A future version of this standard may reference a registry containing election data-related codes, thus this element may change.

Schema definition:

```
<xsd:complexType name="ExternalIdentifiers">
  <xsd:sequence>
    <xsd:element name="ExternalIdentifier" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Type" type="IdentifierType"/>
          <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
          <xsd:element name="Value" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="label" type="xsd:string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
```

4.2.14 The <GpUnit> Element and Extension Base

For defining geo-political units such as cities, districts, jurisdictions, precincts or split precincts, for the purpose of associating contests, offices, vote counts, or other information with the geographies. <ElectionReport> includes <GpUnit>.

<GpUnit> is an abstract element with two xsi:types:

- <GpUnit xsi:type="ReportingUnit">, for associating vote counts with a specific vote-capture device (see section 4.2.13)
- <GpUnit xsi:type="ReportingDevice">, for associating vote counts with geo-political units such as cities, districts, counties, precincts, etc. (see section 4.2.13)

<Election> and <Contest> contain a required reference to a <GpUnit> element defined as the jurisdiction of the election or contest respectively; <Office> contains a similar reference that is optional.

<SummaryCounts> and <VoteCounts> reference <GpUnit> to link vote or summary counts to <GpUnit> elements defined for precincts or other types of geo-political units. <GpUnit> also includes <SummaryCounts> so as to provide summary counts for the corresponding geo-political unit. <BallotStyle> references <GpUnit> to link a ballot style to its corresponding geo-political unit.

<ComposingGpUnitId> is used to reference other <GpUnit> elements and associate them with the parent <GpUnit>, with the rule that each referenced element **must** be defined for a geo-political unit that is contained within the parent <GpUnit> element. For example, precincts are often contained within counties, thus within a <GpUnit> element defined for a county, it is correct to reference elements defined for precincts. As well, within a <GpUnit> element defined for a district, it is correct to reference those elements defined for precincts and split precincts that are contained within the district element.

If this above rule of containment is not observed, then applications will not know how to navigate within instance files and find the needed information. Applications need to know that, e.g., state elements can reference county elements, or precinct elements can reference split-precinct elements. If this rule is violated, for example, split-precinct elements are referencing precinct elements, an application would not be able to find information correctly within the instance file.

When including <ExternalIdentifiers>, if the type is not listed in enumeration <IdentifierType>, use *other* and include the type (that is not listed in the enumeration) in <OtherType>.

Table 4.27 – Attributes for <GpUnit>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.28 – Elements for <GpUnit>

Element	Multiplicity	Type	Element Description
<ComposingGpUnitId>	0 or more	xsd:IDREF	Unique identifier for a <GpUnit> element. For creating a reference to another <GpUnit> that is contained with the parent <GpUnit>.
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID with the <GpUnit>, e.g., a district's or county's code.
<Name>	0 or 1	xsd:string	Name of the geo-political unit.
<SummaryCounts>	0 or more	SummaryCounts	Ballot summary counts (overvotes, undervotes, total ballots, etc.) optionally broken down by device type and ballot class.

Schema Definition:

```

<xsd:complexType name="GpUnit" abstract="true">
  <xsd:sequence>
    <xsd:element name="ComposingGpUnitId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="SummaryCounts" type="SummaryCounts" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>

```

4.2.14.1 The *xsi:type ReportingDevice*

For reporting counts associated with a specific vote-capture device. It is an *xsi:type* of *<GpUnit>* and therefore inherits its attributes and elements (see Table 4.27 and Table 4.28). Its syntax is:

```
<GpUnit xsi:type="ReportingDevice" ... />
```

It identifies a specific vote-capture device by using the *<SerialNumber>* element, and then includes *<Device>* to identify other characteristics of the device such as manufacturer and model.

Table 4.29 – Elements for *<GpUnit xsi:type="ReportingDevice">*

Element	Multiplicity	Type	Element Description
<i><Device></i>	0 or 1	Device	For filtering the vote counts by device type, model, and manufacturer.
<i><SerialNumber></i>	0 or 1	xsd:string	A serial number or otherwise identifier associated with the device.

Schema Definition:

```
<xsd:complexType name="ReportingDevice">
  <xsd:complexContent>
    <xsd:extension base="GpUnit">
      <xsd:sequence>
        <xsd:element name="Device" type="Device" minOccurs="0"/>
        <xsd:element name="SerialNumber" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.14.2 The *xsi:type ReportingUnit*

For defining a geo-political unit such as state, county, township, precinct, etc., using the *<ReportingUnitType>* enumeration. It is an *xsi:type* of *<GpUnit>* and therefore inherits its attributes and elements (see Table 4.27 and Table 4.28). Its syntax is:

```
<GpUnit xsi:type="ReportingUnit" ... />
```

This element optionally references *<Person>* to associate one or more 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 element 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>* so as to include election authority-related information.

The *<Type>* sub-element uses the *<ReportingUnitType>* enumeration to specify the type of geo-political geography being defined. *<ReportingUnitType>* contains the most common types of geographies, e.g., state, county, precinct, and so forth². 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>* sub-element or by examining the *<Contest>* element's *<ElectoralDistrictId>* sub-element, which links to the electoral district associated with the contest (see section 4.2.6). 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.

Table 4.30 – Elements for *<GpUnit xsi:type="ReportingUnit">*

Element	Multiplicity	Type	Element Description
<i><AuthorityId></i>	0 or more	xsd:IDREF	The <i>objectId</i> of one or more <i><Person></i> elements describing an authority responsible for the reporting unit.
<i><ContactInformation></i>	0 or 1	ContactInformation	For associating contact information with the reporting unit.

² A future version of this schema may specify geographies differently using a structured registry of defined types.

Element	Multiplicity	Type	Element Description
<CountStatus>	0 or more	CountStatus	For providing various counting status on types of ballots or other items.
<ElectionAdministration>	0 or 1	ElectionAdministration	For use when the reporting unit serves as the authority in the election.
<IsDistricted>	0 or 1	xsd:boolean	Boolean to indicate whether the reporting unit is a district; assumed to be “no” if not present.
<Number>	0 or 1	xsd:string	A number associated with the reporting unit; for compatibility with VIP.
<PartyRegistration>	0 or more	PartyRegistration	For associating a count of registered voters per party with the geo-political unit.
<SpatialDimension>	0 or 1	SpatialDimension	For describing the reporting unit’s spatial extent (a polygon that shows the related area).
<SubUnitsReported>	0 or 1	xsd:integer	Number of associated subunits such as precincts that have completed reporting.
<TotalSubUnits>	0 or 1	xsd: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 or 1	xsd:string	For use when <ReportingUnitType> value is <i>other</i> .
<VotersParticipated>	0 or 1	xsd:integer	Number of voters who have participated in the election, i.e., shown up at the polls, including those who did not cast ballots.
<VotersRegistered>	0 or 1	xsd:integer	Number of registered voters residing within the boundaries of the geo-

Element	Multiplicity	Type	Element Description
			political unit.

Schema Definition:

```

<xsd:complexType name="ReportingUnit">
  <xsd:complexContent>
    <xsd:extension base="GpUnit">
      <xsd:sequence>
        <xsd:element name="AuthorityId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="ContactInformation" type="ContactInformation"
minOccurs="0"/>
        <xsd:element name="CountStatus" type="CountStatus" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="ElectionAdministration"
type="ElectionAdministration" minOccurs="0"/>
        <xsd:element name="IsDistricted" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="Number" type="xsd:string" minOccurs="0"/>
        <xsd:element name="PartyRegistration" type="PartyRegistration"
minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="SpatialDimension" type="SpatialDimension"
minOccurs="0"/>
        <xsd:element name="SubUnitsReported" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="TotalSubUnits" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="Type" type="ReportingUnitType"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="VotersParticipated" type="xsd:integer"
minOccurs="0"/>
        <xsd:element name="VotersRegistered" type="xsd:integer" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

4.2.15 The <InternationalizedText> Element and LanguageString Type

For strings that can contain text in one of a number of different languages, for use with text that can be included on the ballot. The *Identifier* 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. An example of usage for the string “This is content in Spanish.” is as follows:

```
<InternationalizedText>
  <Text Language="es">Este es el contenido en Español.</LanguageString>
</InternationalizedText>
```

Values for *language* include:

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

Table 4.31 – Attributes for <InternationalizedText>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.32 – Elements for <InternationalizedText>

Element	Multiplicity	Type	Element Description
<Text>	1 or more	LanguageString	A string of text, i.e., possibly non-English.

Schema Definition:

```
<xsd:complexType name="InternationalizedText">
  <xsd:sequence>
    <xsd:element name="Text" type="LanguageString" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
```

Table 4.33 – Attributes for <Text>

Attribute	Required	Type	Attribute Description
language	yes	xsd:language	Identification of the language, e.g., “es”.

Schema Definition:

```
<xsd:complexType name="LanguageString">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="language" type="xsd:language" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

4.2.16 The <Office> Element

For defining an office and associated information associated with a contest and/or a district.

<ElectionReport> includes <Office> (and <OfficeGroup>, see section 4.2.17).

<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 ElectionReport (see section 4.2.17) to assign a name to a grouping of office definitions.

<Office> includes an optional <ElectoralDistrictId> 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, <ElectoralDistrictId> would include a reference to the <GpUnit> defined for the district associated with that office.

Table 4.34 – Attributes for <Office>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.35 – Elements for <Office>

Element	Multiplicity	Type	Element Description
<ContactInformation>	0 or 1	ContactInformation	For associating various contact information with the office.
<ElectoralDistrictId>	0 or 1	xsd:IDREF	Unique identifier for a <GpUnit> element. For associating the office with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc.
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID with the office.
<FilingDeadline>	0 or 1	xsd:date	Date and time when a candidate must have filed for the contest for the office.
<IsPartisan>	0 or 1	xsd:boolean	Boolean to indicate whether the office is partisan, e.g., “yes” or “no”. Assumed to be “no” if not present.
<Name>	1	InternationalizedText	Name of the office; can appear on the ballot.

Element	Multiplicity	Type	Element Description
<OfficeHolderPersonId>	0 or more	xsd:IDREF	Unique identifier for a <Candidate> element defined for the office holder.

Table 4.36 – Attributes for <Term>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.37 – Elements for <Term>

Element	Multiplicity	Type	Element Description
<StartDate>	0 or 1	xsd:date	Start date for the current term of the office.
<EndDate>	0 or 1	xsd:date	End date for the current term of the office.
<Type>	0 or 1	OfficeTermType	Enumerated type of term, e.g., <i>full-term</i> , <i>unexpired-term</i> , etc.

Schema Definition:

```

<xsd:complexType name="Office">
  <xsd:sequence>
    <xsd:element name="ContactInformation" type="ContactInformation"
minOccurs="0"/>
    <xsd:element name="ElectoralDistrictId" type="xsd:IDREF" minOccurs="0"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
    <xsd:element name="FilingDeadline" type="xsd:date" minOccurs="0"/>
    <xsd:element name="IsPartisan" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="Name" type="InternationalizedText"/>
    <xsd:element name="OfficeHolderPersonId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Term" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="StartDate" type="xsd:date" minOccurs="0"/>
          <xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
          <xsd:element name="Type" type="OfficeTermType" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="label" type="xsd:string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>

```

```
</xsd:complexType>
```

4.2.17 The <OfficeGroup> Element

<OfficeGroup> is used to assign a name to a grouping of office definitions. It includes references to <Office> elements 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. <ElectionReport> includes <OfficeGroup>.

As an example, <OfficeGroup> can be used to group state offices, etc., as follows:

```
<OfficeGroup>
  <Name>State House Offices</Name>
  <SubOfficeGroup>
    <Name>Upper House Offices</Name>
    <OfficeId>123iqw</OfficeId>
    <OfficeId>122iqw</OfficeId>
    <OfficeId>121iqw</OfficeId>
  </SubOfficeGroup>
  ...
</OfficeGroup>
```

Table 4.38 – Attributes for <OfficeGroup>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.39 – Elements for <OfficeGroup>

Element	Multiplicity	Type	Element Description
<Name>	1	xsd:string	Name of the office grouping.
<OfficeId>	0 or more	xsd:IDREF	Unique identifier for an <Office> element. For associating a name with a grouping of Office elements.
<SubOfficeGroup>	0 or more	OfficeGroup	For defining a nested hierarchy of <Office> element groupings.

Schema Definition:

```
<xsd:complexType name="OfficeGroup">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

```
        <xsd:element name="SubOfficeGroup" type="OfficeGroup" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
```


4.2.18 The <Party> Element and Extension Base

Used to describe a political party that can then be referenced in other elements.

<ElectionReport> includes <Party>. <Candidate>, <PartyContest>, <PartyRegistration>, and <Person> reference <Party>.

<Party> is an abstract element with one xsi:type:

<Party xsi:type="Coalition">, used to define coalitions (see section 4.2.18.1)

The <Color> element specifies a 6-digit RGB code displayable using HTML. For example, the color green is encoded as:

<Party Color="00FF00" ... />

Table 4.40 – Attributes for <Party>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.41 – Elements for <Party>

Element	Multiplicity	Type	Element Description
<Abbreviation>	0 or 1	xsd:string	Short name for the party, e.g., "DEM".
<Color>	0 or 1	HTMLColorString	For associating an HTML RGB color coding with the party.
<ExternalIdentifiers>	0 or 1	ExternalIdentifiers	For associating an ID with the party.
<LogoUri>	0 or 1	xsd:anyUri	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.

Schema Definition:

```
<xsd:complexType name="Party">
  <xsd:sequence>
    <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Color" type="HtmlColorString" minOccurs="0"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

```
    <xsd:element name="LogoUri" type="xsd:anyURI" minOccurs="0"/>
    <xsd:element name="Name" type="InternationalizedText"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
```

4.2.18.1 The xsi:type Coalition

For defining a coalition, i.e., a collection of parties organized for the purpose of endorsing a candidates in a contest. It is an xsi:type of `<Party>` and therefore inherits its attributes and elements (see Table 4.40 and Table 4.41). Its syntax is:

```
<Party xsi:type="Coalition" ... />
```

`<Coalition>` elements are created by defining `<Party>` elements from `<ElectionReport>`, but using the above syntax (thus, coalition elements can be referenced via any references to `<Party>` included in other elements, e.g., `<Candidate>` or `<CandidateSelection>`). `<Coalition>` elements themselves are composed of multiple occurrences of `<Party>` references along with a reference to an associated contest(s), for example:

```
<Party xsi:type="Coalition" objectId="102abc">
  <Name>Orange Coalition</Name>
  <ContestId>2930abq</ContestId>
  <PartyId>493cba</PartyId>
  <PartyId>493cba</PartyId>
  <PartyId>493cba</PartyId>
</Party>
```

If `<ContestId>` is not present, a general default is that the coalition endorses the associated parties.

Table 4.42 – Elements for `<Party xsi:type="Coalition">`

Element	Multiplicity	Type	Element Description
<code><ContestId></code>	0 or more	xsd:IDREF	Unique identifier for a <code><Contest></code> element. For associating contests with the coalition.
<code><PartyId></code>	0 or more	xsd:IDREF	Unique identifier for a <code><Party></code> element. For associating parties with the coalition.

Schema Definition:

```
<xsd:complexType name="Coalition">
  <xsd:complexContent>
    <xsd:extension base="Party">
      <xsd:sequence>
        <xsd:element name="ContestId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4.2.19 The <PartyRegistration> Element

For tracking the number of registered voters per party per geo-political 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>.

Table 4.43 – Elements for <PartyRegistration>

Element	Multiplicity	Type	Element Description
<Count>	1	xsd:integer	A count for tracking the number of registered voters.
<PartyId>	1	xsd:IDREF	Unique identifier for a <Party> element. For associating a political party.

Schema Definition:

```
<xsd:complexType name="PartyRegistration">
  <xsd:sequence>
    <xsd:element name="Count" type="xsd:integer"/>
    <xsd:element name="PartyId" type="xsd:IDREF"/>
  </xsd:sequence>
</xsd:complexType>
```

4.2.20 The <Person> Element

For defining information about a person; the person may be a candidate, election official, authority for a reporting unit, etc. <ElectionReport> includes <Person>. <Candidate> and <ElectionAuthority> and <GpUnit> reference <Person>. <Person> optionally includes <ContactInformation> for associating contact information.

Multiple occurrences of the <MiddleName> element can be used as needed, e.g., for names such as “John Andrew Winston Smith”, as follows:

```
<Person objectId="P12321">
  <FirstName>John</FirstName>
  <MiddleName>Andrew</MiddleName>
  <MiddleName>Winston</MiddleName>
  <LastName>Smith</LastName>
</Person>
```

Table 4.44 – Attributes for <Person>

Attribute	Required	Type	Attribute Description
objectId	yes	xsd:ID	Unique internal identifier used by other element(s) to reference this element.

Table 4.45 – Elements for <Person>

Element	Multiplicity	Type	Element Description
<ContactInformation>	0 or more	ContactInformation	For associating contact information with the person.
<DateOfBirth>	0 or 1	xsd:date	Person’s date of birth.
<FirstName>	0 or 1	xsd:string	Person’s first (given) name.
<FullName>	0 or 1	InternationalizedText	Person’s full name.
<Gender>	0 or 1	xsd:string	Person’s gender.
<LastName>	0 or 1	xsd:string	Person’s last (family) name.
<MiddleName>	0 or more	xsd:string	Person’s middle name.
<Nickname>	0 or 1	xsd:string	Nickname associated with the person.
<PartyId>	0 or 1	xsd:IDREF	Unique identifier for a Party element. For associating a political party with the person.
<Prefix>	0 or 1	xsd:string	A prefix associated with the person, e.g., Mr.
<Profession>	0 or 1	InternationalizedText	Person’s profession.
<Suffix>	0 or 1	xsd:string	A suffix associated with the

Element	Multiplicity	Type	Element Description
			person, e.g., Jr.
<Title>	0 or 1	InternationalizedText	A title associated with the person.

Schema Definition:

```

<xsd:complexType name="Person">
  <xsd:sequence>
    <xsd:element name="ContactInformation" type="ContactInformation"
minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="DateOfBirth" type="xsd:date" minOccurs="0"/>
    <xsd:element name="FirstName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="FullName" type="InternationalizedText" minOccurs="0"/>
    <xsd:element name="Gender" type="xsd:string" minOccurs="0"/>
    <xsd:element name="LastName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="MiddleName" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Nickname" type="xsd:string" minOccurs="0"/>
    <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"/>
    <xsd:element name="Prefix" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Profession" type="InternationalizedText" minOccurs="0"/>
    <xsd:element name="Suffix" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Title" type="InternationalizedText" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>

```

4.2.21 The <Schedule> and <Hours> Elements

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

<ContactInformation> includes <Schedule>.

Table 4.46 – Attributes for <Schedule>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.47 – Elements for <Schedule>

Element	Multiplicity	Type	Element Description
<Hours>	0 or more	Hours	For specifying a range of hours for a schedule.
<IsOnlyByAppointment>	0 or 1	xsd:boolean	If an appointment is only by appointment; assumed to be “no” if not present.
<IsOrByAppointment>	0 or 1	xsd:boolean	If an appointment can be by appointment presumably as desired; assumed to be “no” if not present.
<IsSubjectToChange>	0 or 1	xsd:boolean	If an appointment may be subject to change; assumed to be “no” if not present.
<StartDate>	0 or 1	xsd:date	For the starting date of the schedule.
<EndDate>	0 or 1	xsd:date	For the ending date of the schedule.

Schema Definition:

```

<xsd:complexType name="Schedule">
  <xsd:sequence>
    <xsd:element name="Hours" type="Hours" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="IsOnlyByAppointment" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="IsOrByAppointment" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="IsSubjectToChange" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="StartDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>

```

4.2.21.1 The <Hours> Element

<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, for example, for specific hours on a Wednesday and Thursday:

```
<Schedule startDate="2015-05-15">
  <Hours Day="Friday" StartTime="09:00:00-05:00"
    EndTime="17:00:00-05:00"/>
  <Hours Day="Saturday" StartTime="09:00:00-05:00"
    EndTime="21:00:00-05:00"/>
</Schedule>
```

Table 4.48 – Attributes for <Schedule>

Attribute	Required	Type	Attribute Description
label	no	xsd:string	For use as needed and compatibility with the VIP schema.

Table 4.49 – Elements for <Hours>

Element	Multiplicity	Type	Element Description
<Day>	0 or 1	DayType	Day of week or weekend.
<StartTime>	1	TimeWithZone	Start time of the schedule.
<EndTime>	1	TimeWithZone	End time of the schedule.

Schema Definition:

```
<xsd:complexType name="Hours">
  <xsd:sequence>
    <xsd:element name="Day" type="DayType" minOccurs="0"/>
    <xsd:element name="StartTime" type="TimeWithZone"/>
    <xsd:element name="EndTime" type="TimeWithZone"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
```


4.2.22 The <SpatialDimension> and <SpatialExtent> Elements

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> elements are properly related. <GpUnit> includes <SpatialDimension>.

Table 4.50 – Elements for <SpatialDimension>

Element	Multiplicity	Type	Element Description
<MapUri>	0 or 1	xsd:anyURI	Typically a URL to a map of the <GpUnit>.
<SpatialExtent>	0 or 1	SpatialExtent	For associating a <GpUnit> element's spatial extent information.

Schema Definition:

```

<xsd:complexType name="SpatialDimension">
  <xsd:sequence>
    <xsd:element name="MapUri" type="xsd:anyURI" minOccurs="0"/>
    <xsd:element name="SpatialExtent" type="SpatialExtent" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

4.2.22.1 The SpatialExtent Element

<SpatialDimension> includes <SpatialExtent> for defining a <GpUnit> element's spatial extent data and the format used for the spatial extent.

Table 4.51 – Elements for <SpatialExtent>

Element	Multiplicity	Type	Element Description
<Coordinates>	1	xsd:string	The data coordinates constituting the spatial extent.
<Format>	1	GeoSpatialFormat	Enumerated type for the format used, e.g., <i>gml</i> , <i>kml</i> , <i>wkt</i> , etc.

Schema Definition:

```
<xsd:complexType name="SpatialExtent">
  <xsd:sequence>
    <xsd:element name="Coordinates" type="xsd:string"/>
    <xsd:element name="Format" type="GeoSpatialFormat"/>
  </xsd:sequence>
</xsd:complexType>
```

4.2.23 The <SummaryCounts> (extension base <Counts>) Element

For reporting on contest-wide or geo-political unit-wide summary counts. Includes <Counts> as an extension base and therefore inherits the elements from <Counts> (see Table 4.20), but it is included directly from <Contest> and <GpUnit> (i.e., it is not included as an xsi:type of <Counts>).

<Contest> includes <SummaryCounts> for providing a geographical scope-wide summary of miscellaneous counts associated with a contest, including total number of ballots cast containing the contest, total number of overvotes, undervotes, and write-ins. <SummaryCounts> can optionally reference <GpUnit> elements defined for lower level reporting units, e.g., precincts, so as to associate summary counts with each precinct or other lower-level reporting units within the scope of the contest.

<GpUnit> includes <SummaryCounts> for providing summary counts specific to that <GpUnit>, e.g., a <GpUnit> defined for a district can include <SummaryCounts> for this purpose. <SummaryCounts>, in this case, would not include the <GpUnitId> element in <Counts>.

In <Counts>, if the type of count item is not listed in enumeration <CountItemType>, use *other* and include the type (that is not listed in the enumeration) in <OtherType>.

Table 4.52 – Elements for <SummaryCounts>

Element	Multiplicity	Type	Element Description
<BallotsCast>	0 or 1	xsd:integer	Number of ballots cast.
<BallotsOutstanding>	0 or 1	xsd:integer	Number of ballots not yet counted.
<BallotsRejected>	0 or 1	xsd:integer	Number of ballots rejected.
<Overvotes>	0 or 1	xsd:integer	Number of overvotes.
<Undervotes>	0 or 1	xsd:integer	Number of undervotes.
<WriteIns>	0 or 1	xsd:integer	Number of write-ins.

Schema Definition:

```

<xsd:complexType name="SummaryCounts">
  <xsd:complexContent>
    <xsd:extension base="Counts">
      <xsd:sequence>
        <xsd:element name="BallotsCast" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="BallotsOutstanding" type="xsd:integer"
minOccurs="0"/>
        <xsd:element name="BallotsRejected" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="Overvotes" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="Undervotes" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="WriteIns" type="xsd:integer" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </complexContent>
</complexType>

```

```
</xsd:complexContent>  
</xsd:complexType>
```

4.2.24 The <VoteCounts> (extension base <Counts>) Element

For reporting on vote counts for ballot selections in a contest. <VoteCounts> includes <Counts> as an extension base and therefore inherits the elements from <Counts> (see Table 4.20), but it is included directly by <BallotSelection> (i.e., it is not referenced as an xsi:type of <Counts>).

In <Counts>, if the type of count item is not listed in enumeration <CountItemType>, use *other* and include the type (that is not listed in the enumeration) in <OtherType>.

Table 4.53 – Elements for <VoteCounts>

Element	Multiplicity	Type	Element Description
<Count>	0 or 1	xsd:float	Count of contest votes cast; can include a fractional component in special cases.

Schema Definition:

```

<xsd:complexType name="VoteCounts">
  <xsd:complexContent>
    <xsd:extension base="Counts">
      <xsd:sequence>
        <xsd:element name="Count" type="xsd:float" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

4.3 Enumerations

The following sections deal with enumerations in the schema.

4.3.1 The BallotMeasureType Enumeration

Enumeration for types of ballot measures in the *<BallotMeasureContest>* element.

Table 4.54 – Values for <BallotMeasureType>

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

Schema Definition:

```
<xsd:simpleType name="BallotMeasureType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="ballot-measure"/>  
    <xsd:enumeration value="initiative"/>  
    <xsd:enumeration value="referendum"/>  
    <xsd:enumeration value="other"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

4.3.2 The CandidatePostElectionStatus Enumeration

Enumeration for various post-election statuses applicable to a candidate in the *<Candidate>* element.

Table 4.55 – Values for <CandidatePostElectionStatus>

Value	Value Description
advanced-to-runoff	For candidates who have advanced to a runoff.
projected-winner	For a projected contest winner.
winner	For the official contest winner or one of N contest winners for N-of-M voting.
withdrawn	For candidates who have withdrawn from the contest.

Schema Definition:

```
<xsd:simpleType name="CandidatePostElectionStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="advanced-to-runoff"/>
    <xsd:enumeration value="projected-winner"/>
    <xsd:enumeration value="winner"/>
    <xsd:enumeration value="withdrawn"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.3.3 The CandidatePreElectionStatus Enumeration

Enumeration for various pre-election statuses applicable to a candidate in the *<Candidate>* element.

Table 4.56 – Values for <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.
write-in	For when the candidate is a write-in.

Schema Definition:

```
<xsd:simpleType name="CandidatePreElectionStatus">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="filed"/>  
    <xsd:enumeration value="qualified"/>  
    <xsd:enumeration value="withdrawn"/>  
    <xsd:enumeration value="write-in"/>  
  </xsd:restriction>  
</xsd:simpleType>
```


4.3.4 The CountItemStatus Enumeration

Enumeration for various counting-related statuses for types of ballots or write-ins, used in the *<CountStatus>* element.

Table 4.57 – Values for <CountItemStatus>

Value	Value Description
completed	For counts that are complete.
in-process	For counts that are in process
not-processed	When the counting has not started or is not underway.
Unknown	When the status of the counting is unknown.

Schema Definition:

```
<xsd:simpleType name="CountItemStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="completed"/>
    <xsd:enumeration value="in-process"/>
    <xsd:enumeration value="not-processed"/>
    <xsd:enumeration value="unknown"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.3.5 The CountItemType Enumeration

Enumeration for the items that are counted during the course of an election and for which the status of the counts are of interest. Used in the `<Counts>` and `<CountStatus>` elements.

Table 4.58 – Values for `<CountItemType>`

Value	Value Description
absentee	For any/all types of absentee, generally when absentee is not broken out into specific types.
absentee-fwab	A type of absentee; for Federal Write-in Absentee Ballots.
absentee-in-person	A type of absentee; for absentee ballots cast in-person, e.g., at a county office.
absentee-mail	A type of absentee; for postal mail absentee ballots.
early	For ballots cast during early voting periods.
election-day	For ballots cast on election day.
provisional	For challenged ballots.
total	For the total of all ballot types.
uocava	For ballots from UOCAVA voters.
write-in	For write-ins on ballots.
other	Used when the type of counting item is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="CountItemType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="absentee"/>
    <xsd:enumeration value="absentee-fwab"/>
    <xsd:enumeration value="absentee-in-person"/>
    <xsd:enumeration value="absentee-mail"/>
    <xsd:enumeration value="early"/>
    <xsd:enumeration value="election-day"/>
    <xsd:enumeration value="provisional"/>
    <xsd:enumeration value="total"/>
    <xsd:enumeration value="uocava"/>
    <xsd:enumeration value="write-in"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.3.6 The DayType Enumeration

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

Table 4.59 – Values for <DayType>

Value	Value Description
all	Used for all days of the week.
sunday	
monday	
tuesday	
wednesday	
thursday	
friday	
saturday	
weekday	Used for any day of the week.
weekend	Used for both Saturday and Sunday.

Schema Definition:

```

<xsd:simpleType name="DayType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="all"/>
    <xsd:enumeration value="Sunday"/>
    <xsd:enumeration value="Monday"/>
    <xsd:enumeration value="Tuesday"/>
    <xsd:enumeration value="Wednesday"/>
    <xsd:enumeration value="Thursday"/>
    <xsd:enumeration value="Friday"/>
    <xsd:enumeration value="Saturday"/>
    <xsd:enumeration value="weekday"/>
    <xsd:enumeration value="weekend"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.3.7 The DeviceType Enumeration

Enumeration for the type of device in the *<Device>* element.

Table 4.60 – Values for <DeviceType>

Value	Value Description
electronic	For DRE (Direct Record Electronic) and touchscreen devices such as tablets.
lever	For lever machines.
manual-count	For hand-counted paper ballots.
mixed-systems	For devices, e.g., that print voter choices on an optical scan ballot (hybrid of a DRE and an optical scan system)
opscan-central	For an optical scanner used at a central office with no opportunity for voter correction of mistakes.
opscan-precinct	For an optical scanner used at a precinct or other location where voter correction of mistakes such as overvotes is possible.
punch-card	For punch card devices.
unknown	Used when the type of device is unknown.
other	Used when the device type is not listed in this enumeration.

Schema Definition:

```

<xsd:simpleType name="DeviceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="electronic"/>
    <xsd:enumeration value="lever"/>
    <xsd:enumeration value="manual-count"/>
    <xsd:enumeration value="mixed-systems"/>
    <xsd:enumeration value="opscan-central"/>
    <xsd:enumeration value="opscan-precinct"/>
    <xsd:enumeration value="punch-card"/>
    <xsd:enumeration value="unknown"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.3.8 The ElectionType Enumeration

Enumeration for the type of election in the *<ElectionReport>* element.

Table 4.61 – Values for <ElectionType>

Value	Value Description
general	For the election held typically on the national day for elections (e.g., the Tuesday after the 1 st Monday in November).
partisan-primary-closed	For a primary election that is for a specific party where voter eligibility is based on registration.
partisan-primary-open	For a primary election that is for a specific party where voter declares desired party or chooses in private.
primary	For a primary election, type not specified such as nonpartisan.
runoff	For an election to decide a prior contest that ended with no candidate receiving a majority of the votes.
special	For an election held out of sequence for special circumstances, e.g., to fill a vacated office.
other	Used when the election type is not listed in this enumeration.

Schema Definition:

```
<xsd:simpleType name="ElectionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="general"/>
    <xsd:enumeration value="partisan-primary-closed"/>
    <xsd:enumeration value="partisan-primary-open"/>
    <xsd:enumeration value="primary"/>
    <xsd:enumeration value="runoff"/>
    <xsd:enumeration value="special"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.3.9 The GeoSpatialFormat Enumeration

Enumeration for geospatial vector data formats used in geographic information system (GIS) software, used in the *<SpatialExtent>* element.

Table 4.62 – Values for <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.

Schema Definition:

```
<xsd:simpleType name="GeoSpatialFormat">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="geo-json"/>
    <xsd:enumeration value="gml"/>
    <xsd:enumeration value="kml"/>
    <xsd:enumeration value="shp"/>
    <xsd:enumeration value="wkt"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.3.10 The IdentifierType Enumeration

Enumeration for election data-related codes in the *<ExternalIdentifiers>* element³.

Table 4.63 – Values for <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.

Schema Definition:

```
<xsd:simpleType name="IdentifierType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="fips"/>
    <xsd:enumeration value="local-level"/>
    <xsd:enumeration value="national-level"/>
    <xsd:enumeration value="ocd-id"/>
    <xsd:enumeration value="state-level"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
```

³ A future version of this schema may incorporate a registry of election data-related codes.

4.3.11 The OfficeTermType Enumeration

Enumeration for the office term type in the *<Office>* element.

Table 4.64 – Values for <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 fullterm expired.

Schema Definition:

```
<xsd:simpleType name="OfficeTermType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="full-term"/>  
    <xsd:enumeration value="unexpired-term"/>  
  </xsd:restriction>  
</xsd:simpleType>
```


4.3.12 The ReportDetailLevel Enumeration

Enumeration for the detail level of the election results report in the *<Election>* element.

Table 4.65 – Values for <ReportDetailLevel>

Value	Value Description
precinct-level	For reports that contain counts from precincts in the reporting jurisdiction.
summary-contest	For reports that contain only aggregated counts.

Schema Definition:

```
<xsd:simpleType name="ReportDetailLevel">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="precinct-level"/>  
    <xsd:enumeration value="summary-contest"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

4.3.13 The ReportingUnitType Enumeration

Enumeration for the type of geo-political unit in the `<GpUnit xsi:type="ReportingUnit">` element⁴.

Table 4.66 – Values for <ReportingUnitType>

Value	Value Description
ballot-batch	Used for reporting batches of ballots that may cross precinct boundaries.
city	Used for a city that reports results and/or for a district that encompasses the city.
city-council	Used for city council districts.
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.
county	Used also for “Parish”.
county-council	Used for county council districts.
dropbox	Used for a dropbox for absentee ballots.
judicial	Used for judicial districts.
municipality	Used as applicable for various units such as towns, townships, villages that report votes.
polling-place	Used for a polling place.
precinct	Used also for “Ward” or “District” when these terms are used to mean the same thing as “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 a district that encompasses the state.
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.
township	Used in some mid-western states.
utility	Used for a utility district.
village	Used for villages that report votes.
vote-center	Used for a vote center.
ward	Used for combinations or groupings of precincts or other units – use “Precinct” or “CombinedPrecinct” if this term is being used interchangeably.

⁴ A future version of this schema may specify geographies differently using a structured registry of defined types.

Value	Value Description
water	Used for a water district.
other	Used for other types of reporting units not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="ReportingUnitType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ballot-batch"/>
    <xsd:enumeration value="city"/>
    <xsd:enumeration value="city-council"/>
    <xsd:enumeration value="combined-precinct"/>
    <xsd:enumeration value="congressional"/>
    <xsd:enumeration value="county"/>
    <xsd:enumeration value="county-council"/>
    <xsd:enumeration value="drop-box"/>
    <xsd:enumeration value="judicial"/>
    <xsd:enumeration value="municipality"/>
    <xsd:enumeration value="polling-place"/>
    <xsd:enumeration value="precinct"/>
    <xsd:enumeration value="school"/>
    <xsd:enumeration value="special"/>
    <xsd:enumeration value="split-precinct"/>
    <xsd:enumeration value="state"/>
    <xsd:enumeration value="state-house"/>
    <xsd:enumeration value="state-senate"/>
    <xsd:enumeration value="town"/>
    <xsd:enumeration value="township"/>
    <xsd:enumeration value="utility"/>
    <xsd:enumeration value="village"/>
    <xsd:enumeration value="vote-center"/>
    <xsd:enumeration value="ward"/>
    <xsd:enumeration value="water"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.3.14 The ResultsStatus Enumeration

Enumeration for the status of the election results in the *<ElectionReport>* element.

Table 4.67 – Values for <ResultsStatus>

Value	Value Description
certified	For results that have been certified by the election authority.
correction	For results that are a correction to an earlier report.
pre-election	For a pre-election data.
recount	For results that are a recount of an earlier election.
unofficial-complete	For results that are unofficial and complete, e.g., the complete election night results.
unofficial-partial	For results that are unofficial and partial, e.g., partial election night results.

Schema Definition:

```

<xsd:simpleType name="ResultsStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="certified"/>
    <xsd:enumeration value="correction"/>
    <xsd:enumeration value="pre-election"/>
    <xsd:enumeration value="recount"/>
    <xsd:enumeration value="unofficial-complete"/>
    <xsd:enumeration value="unofficial-partial"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.3.15 The VoteVariation Enumeration

Enumeration for contest algorithm or rules in the <Contest> element.

Table 4.68 – Values for <VoteVariationType>

Value	Value Description
1-of-m	For 1 of M voting.
approval	For approval voting.
borda	For the Borda count voting.
cumulative	For cumulative voting.
majority	For majority voting.
n-of-m	For N of M voting.
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.

Schema Definition:

```

<xsd:simpleType name="VoteVariation">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1-of-m"/>
    <xsd:enumeration value="approval"/>
    <xsd:enumeration value="borda"/>
    <xsd:enumeration value="cumulative"/>
    <xsd:enumeration value="majority"/>
    <xsd:enumeration value="n-of-m"/>
    <xsd:enumeration value="plurality"/>
    <xsd:enumeration value="proportional"/>
    <xsd:enumeration value="range"/>
    <xsd:enumeration value="rcv"/>
    <xsd:enumeration value="super-majority"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

4.4 Primitives

The following sections deal with primitives in the schema.

4.4.1 The HTMLColorString Primitive

For a string containing a 6-digit Red-Green-Blue (RGB) code that can be displayed using HTML. Used in *<Party>* to associate a web-displayable color with the party. The RGB code is specified in hexadecimal, such that the RGB code for the color green is “00FF00” (“#00” + “#FF” + “#00”).

Schema definition:

```
<xsd:simpleType name="HtmlColorString">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9a-f]{6}"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.4.2 The Primitive ShortString Primitive

For defining a 16-character annotation, used with character strings in *<AnnotatedString>*.

Schema definition:

```
<xsd:simpleType name="ShortString">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="16"/>
  </xsd:restriction>
</xsd:simpleType>
```

4.4.3 The Primitive TimeWithZone Primitive

For defining a time pattern, including the time zone, used in *<Hours>* for defining a schedule.

Hh:mm:ss+05

Schema definition:

```
<xsd:simpleType name="TimeWithZone">
  <xsd:restriction base="xsd:time">
    <xsd:pattern value="(( [01] [0-9] | 2 [0-3] ) : [0-5] [0-9] : [0-5] [0-9] | ( 24 : 00 : 00 ) )
(Z | [+ -] ( ( 0 [0-9] | 1 [0-3] ) : [0-5] [0-9] | 14 : 00 ) )"/>
  </xsd:restriction>
</xsd:simpleType>
```

5 ElectionResultsReporting Schema Aspects and Usage

This chapter presents an overview of various aspects and usage of the schema, including

- How UML relationships from Chapter 3 are implemented in the XML schema
- General layout of instance files
- How geo-political geography is implemented
- How contests and ballot selections are implemented
- How vote counts and summary vote counts are implemented
- How ballot counts are linked to geo-political geography

No attempt is made to provide a complete overview of usage, but more to describe sufficient usage of certain elements such that readers can understand in general how all elements are to be used. Readers should consult the references for XML and XML digital signature usage in References.

5.1 How UML Relationships Are Implemented in the XML Schema

The XML schema is generated from the UML model, thus having some understanding of how the XML is generated will assist users of the schema as well as others who may wish to extend the model or schema for other purposes. This section provides a brief overview of the schema generation.

When XML is generated from the UML model, the classes in the model get generated as major XML elements, e.g., the UML Candidate class results in the XML `<Candidate>` element. The attributes in the UML classes generally get generated as XML sub-elements within the XML major element, e.g., the UML attribute Name in the Candidate class gets generated as the sub-element `<Name>` in the `<Candidate>` major element.

In general, the schema treats major elements such as candidates or contests or parties as if they are tables in a database that are linked together depending on the type of relationship between the objects. The relationships between UML classes result in XML elements that function differently depending upon the type of relationship. As a brief guide, using Figure 5 on page 15, the directed composition relationship between Election and Contest results in 1 or more XML sub-elements named `<Contest>` being created in `<Election>`.

“Is a Type of” or “Instance of” relationships are slightly more complex. For Contest and CandidateContest in Figure 5, CandidateContest is a type of Contest and thus is generated with `<Contest>` as an XML extension base and all attributes and elements of `<Contest>` are global to elements that use `<Contest>` as an extension base. The XML syntax for using `<Contest>` for a candidate contest is as follows:

```
<Contest xsi:type="CandidateContest" objectId="OID-AG">
  <Name>Attorney General</Name>
  ...
</Contest>
```

Figure 9 – Is a Type of Relationship in XML

“Is a Type of” relationships are used in cases such as `<Contest>` where there can be several types of contests, but the different types of contests generally require the same set of attributes plus additional attributes that are specific to a type. Thus, `<Contest>` is generated as an abstract XML element with all of its sub-elements and attributes global to the specific types of contests, e.g., `CandidateContest`. But, the sub-elements of `CandidateContest` are private to `CandidateContest`.

5.1.1 Directed Association Relationships and the *objectId* Attribute

Directed association relationships require more explanation. Directed association relationships are used throughout the UML model and in the XML schema to reference or link one element to another, and they give XML instance files the capability to be structured and used as tables in a database, with pointers in one table referencing other tables as needed. An example of this is the `<PartyId>` sub-element in the `<Candidate>` element, which serves to link a candidate with the candidate’s party.

First, when the XML schema is generated, every class in the UML model that is at the endpoint of a directed association relationship is generated with an *objectId* attribute of type `xsd:ID`. For example, `<Party>` is referenced from a number of classes, thus the `<Party>` XML element is generated with the *objectId* attribute. Type `xsd:ID` requires that *objectId* be a unique legal XML value that does not contain a colon, and digits, the hyphen, and the period may not be used as the starting character.

Secondly, for the class at the starting point of the directed association, an element is generated that serves to reference the element at the endpoint. Furthermore, the name of the element is the name of the class at the endpoint with “Id” concatenated to it.

For Candidate and Party from Figure 5, the directed association relationship results in an XML sub-element named `<PartyId>` of type `xsd:IDREF` being created for `<Candidate>`. The `<PartyId>` sub-element links to a specific `<Party>` element that will be associated with that candidate, thus linking the candidate to the candidate’s party. The usage is shown below:

```
<Party objectId="OID-IND">
  <Name>Independent</Name>
</Party>
...
...
<Candidate objectId="OID-C1">
  <Name>John Brown</Name>
  <PartyId>OID-IND</PartyId>
</Candidate>
```

Figure 10 – Directed Association in XML

In this example, “OID-IND” is the value of the *objectId* attribute for the `<Party>` element defined for the Independent Party. To link candidate John Brown to this `<Party>` element, the corresponding `<Candidate>` element uses `<PartyId>` and the same “OID-IND” value.

5.2 Wrapper Elements

Collections of elements in *<ElectionReport>* and *<Election>* that are likely to be quite large are wrapped by container elements so as to make instance files easier to read in XML viewers. For example, in *<Election>*, *<Contest>* is wrapped by *<ContestCollection>*, which then contains 1 or more *<Contest>* sub-elements, as follows:

```
</ContestCollection>
  <Contest xsi:type="CandidateContest" objectId="OID-GOV">
    ...
  </Contest>
  <Contest xsi:type="CandidateContest" objectId="OID-AG">
    ...
  </Contest>
  ...
</ContestCollection>
```

Figure 11 - Use of Wrapper Elements

This applies as well to *<BallotStyle>*, *<Candidate>*, *<GpUnit>*, *<Office>* and *<OfficeGroup>*, *<Party>*, and *<Person>*. Additionally, *<BallotSelection>* wraps the *<VoteCounts>* element using *<VoteCountsCollection>*.

5.3 General Layout of XML Instance Files

In general, instance files are built by creating collections of elements such for office descriptions, geo-political units, candidates, contests, etc., and then linking the elements as necessary using the *objectId* attributes. Most elements are optional, thus instance files will vary in content but generally follow the same structure below:

- *<ElectionReport>* element (pre-election elements)
 - *<GpUnit>* elements
 - *<Office>* and *<OfficeGroup>* elements
 - *<Party>* elements
 - *<Person>* elements
- *<Election>* element(s) (elements generally associated with an election):
 - *<BallotStyle>* elements
 - *<Candidate>* elements
 - Repeated sequences of *<Contest>* elements:
 - *<Contest>* element for Contest 1
 - *<BallotSelection>* elements for the ballot selections in Contest 1
 - *<VoteCounts>* elements associated with the ballot selections
 - *<Contest>* element for Contest 2, etc.

Many of the elements make use of enumerated values, for example, the *<CountStatus>* element makes use of the *<CountItemType>* enumeration for filtering vote counts by ballot type. Most enumerations include a value of *other* for use when none of the enumerated values is applicable,

and the elements using these enumerations include an *<OtherType>* element that can be used to hold another value.

The elements for *<ElectionReport>* and *<Election>*, *<Candidate>*, *<Office>*, *<Party>*, and *<Person>* are relatively straightforward in structure and usage, with the *objectId* attributes used to link elements as needed. The usage for the *<GpUnit>* element and the elements used with *<Contest>* is more complicated and explained in the following sections.

5.4 Defining Geo-Political Geography Using the *<GpUnit>* Element

A primary feature of the schema is the geo-political geography element *<GpUnit>*, which can be used in a variety of ways to exactly mirror a jurisdiction's geo-political geography. *<GpUnit>* elements are used in an election report to list the geo-political units of geography that lay within the geographic scope of the election results such as counties, cities, precincts, etc., so that ballot and vote counts can be associated with them. *<GpUnit>* elements are required in two places:

1. The *<Election>* element contains a required link to a *<GpUnit>* defined for the geographical scope of the election, using the *<ElectionScopeId>* sub-element.
2. The *<Contest>* element contains a required link to a *<GpUnit>* defined for the geographical scope of the contest, using the *<ElectoralDistrictId>* sub-element. The *<Office>* element contains the same sub-element, but it is optional.

<GpUnit> elements, as described in section 4.2.14, can be used for the following major types of objects:

- Geographies that report election results or that can have vote counts associated with them, including for states, counties, cities, towns, precincts, split precincts, etc.
- Geographies that serve as electoral districts, or districts that do not have contests associated
- Specific voting devices
- Polling places, drop boxes, vote centers, and other miscellaneous objects

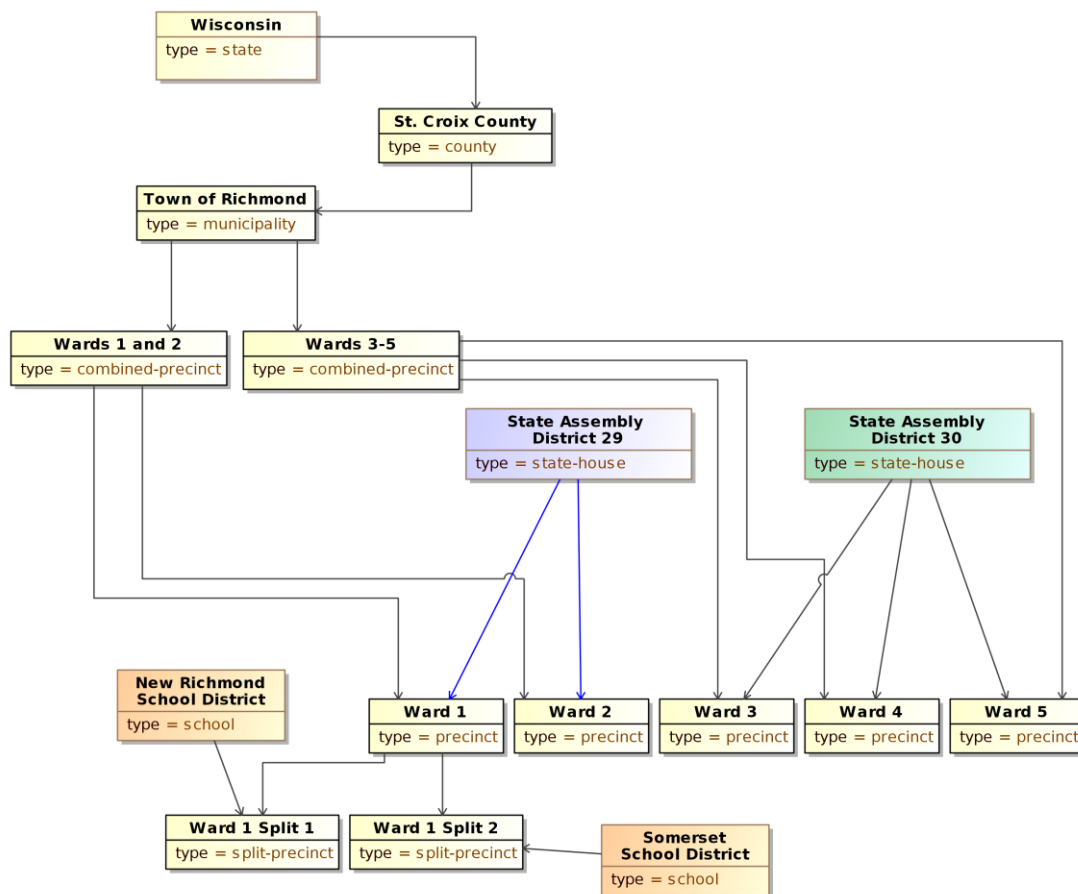


Figure 12 - Geo-political Geography Example from Wisconsin

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.

5.4.1 Defining and Associating *<GpUnit>* Elements

Figure 12 shows an example of geo-political geography in the town of Richmond, WI. Figure 13 shows the *<GpUnit>* definitions for the wards in Figure 12 (in Wisconsin, a ward is the same as a precinct):

```
<GpUnit xsi:type="ReportingUnit" objectId="OID-State">
  <ComposingGpUnitId>OID-ST-CROIX</ComposingGpUnitId>
  <Name>State of Wisconsin</Name><Type>state</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" objectId="OID-ST-CROIX">
  <ComposingGpUnitId>OID-Richmond</ComposingGpUnitId>
  <Name>St. Croix County</Name><Type>county</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" objectId="OID-Richmond">
```

```

    <ComposingGpUnitId>OID-WARDS1-2</ComposingGpUnitId>
    <ComposingGpUnitId>OID-WARDS3-5</ComposingGpUnitId>
    <Name>Town of Richmond</Name><Type>municipality</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARDS1-2">
    <ComposingGpUnitId>OID-WARD1</ComposingGpUnitId>
    <ComposingGpUnitId>OID-WARD2</ComposingGpUnitId>
    <Name>Combined Wards 1 and 2</Name><Type>combined-precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARDS3-5">
    <ComposingGpUnitId>OID-WARD3</ComposingGpUnitId>
    <ComposingGpUnitId>OID-WARD4</ComposingGpUnitId>
    <ComposingGpUnitId>OID-WARD5</ComposingGpUnitId>
    <Name>Combined Wards 3-5</Name><Type>combined-precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD1">
    <ComposingGpUnitId>OID-WARD1-1</ComposingGpUnitId>
    <ComposingGpUnitId>OID-WARD1-2</ComposingGpUnitId>
    <Name>Ward 1</Name><Type>precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD1-1">
    <Name>Ward 1-1</Name><Type>split-precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD1-2">
    <Name>Ward 1-2</Name><Type>split-precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD2">
    <Name>Ward 2</Name><Type>precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD3">
    <Name>Ward 3</Name><Type>precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD4">
    <Name>Ward 4</Name><Type>precinct</Type>
  </GpUnit>
  <GpUnit xsi:type="ReportingUnit" objectId="OID-WARD5">
    <Name>Ward 5</Name><Type>precinct</Type>
  </GpUnit>

```

Figure 13 - <GpUnit> Definitions

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 element. Thus, the definition for the combined precincts and Ward 1 use <ComposingGpUnitId> to reference the precinct <GpUnit> elements that are contained within the combined precincts and the split precinct <GpUnit> elements that are contained within Ward 1.

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

*<ComposingGpUnitId> **must only** reference a geo-political unit that is contained within the parent <GpUnit> element.*

Accordingly, split precincts are contained within precincts and thus precinct elements may reference split-precincts – but not vice versa. Applications need to know that, always,

relationships between *<GpUnit>* elements are structured to be containment relationships, e.g., state elements can reference only objects such as counties within the states, or precinct elements can only reference split-precinct elements, and so forth. If this rule is violated, an application would not be able to find information correctly within the instance file.

A second example of linking *<GpUnit>* elements is from Figure 12 on page 102 for associating the wards with the districts they compose, shown below:

```

<!-- Definition for House District 29 and links to its composing wards -->
<GpUnit xsi:type="ReportingUnit" objectId="OID-HD29">
  <ComposingGpUnitId>OID-WARD1</ComposingGpUnitId>
  <ComposingGpUnitId>OID-WARD2</ComposingGpUnitId>
  <IsDistricted>yes</IsDistricted>
  <Name>House District 29</Name><Type>state-house</Type>
</GpUnit>

<!-- Definition for House District 30 and links to its composing wards -->
<GpUnit xsi:type="ReportingUnit" objectId="OID-HD30">
  <ComposingGpUnitId>OID-WARD3</ComposingGpUnitId>
  <ComposingGpUnitId>OID-WARD4</ComposingGpUnitId>
  <ComposingGpUnitId>OID-WARD5</ComposingGpUnitId>
  <IsDistricted>yes</IsDistricted>
  <Name>House District 30</Name><Type>state-house</Type>
</GpUnit>

```

Figure 14 - Linking *<GpUnit>* Elements to Form Relationships

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 *<ElectoralDistrictId>* sub-element, which links to the electoral district associated with the contest (see section 4.2.6). 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.

5.4.2 Using One *<GpUnit>* Element for Duplicate Electoral Districts

There are, quite often, multiple jurisdictional-wide contests that use, for their respective electoral districts, the same physical district. In a given county, for example, contests for county executive and at large county council positions will all use the same physical district. In states, the contest for governor and the contests for any other state-wide positions will share the same physical district.

Each *<Contest>* element requires an *<ElectoralDistrictId>* sub-element that links a *<GpUnit>* element to the contest; the *<GpUnit>* element will have been defined for the physical geography of the associated district. Because *<GpUnit>* elements defined for states or for large counties may have very long lists of *<ComposingGpUnitId>* sub-elements, one for each precinct

in the county or state, instance files could grow quite large if different *<GpUnit>* elements are defined for each of the contests that share the same physical district.

An option for reducing the file size is to define a *<GpUnit>* element for the, say, county or state, and then re-use the element for contests that are county-wide or state-wide. The generating application could, effectively, keep track of the *objectId* attribute of the county or state *<GpUnit>* element and then use the associated value for each contest's *<ElectoralDistrictId>* sub-element.

Thus, if all contest definitions for jurisdictional-wide contests use the same value for the *<ElectoralDistrictId>* sub-element, duplication can be reduced and file sizes will be smaller.

5.5 Defining Contests Using the *<Contest>* Element

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
 - d. *RetentionContest*, for a contest involving a judicial or other retention
2. A link to a *GpUnit* element defined for the contest's electoral district, using the *<ElectoralDistrictId>* sub-element
3. The *<BallotSelection>* 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
 - c. *PartySelection*, if the selection is for a straight party selection
4. *<VoteCounts>*, to report the number of votes for the ballot selection, for the contest as a whole or broken out by, e.g., precincts and/or types of ballots
5. *<SummaryCounts>*, to report summary counts for the contest, e.g., total ballots cast, overvotes, undervotes, etc.

It is possible to define contests without associated *<BallotSelection>* elements as well as without any of the elements for election results; but these elements are required if associating vote counts with ballot selections.

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 for the number of seats associated with the office and the number of selections a voter can make in the contest (both are generally 1).

PartyContest is included as a contest type because a number of states keep track of the number of straight party selections made by voters and straight party selection does not “fit” within the definition of a candidate contest, i.e., there are no candidates. However, some jurisdictions treat straight party selection as a candidate contest, thus the schema does not prevent usage of the *CandidateContest* type being used for straight party selections.

Following is a short example of a contest involving *CandidateContest* that illustrates using ballot selections and counts, with the counts being aggregated, that is, associated with the electoral district of the contest as a whole:

```
<Candidate objectId="OID-C1">
  <Name>John Brown</Name>
</Candidate>
<Candidate objectId="OID-C2">
  <Name>Thomas Paine</Name>
</Candidate>
...
...
<!-- Governor contest -->
<Contest xsi:type="CandidateContest" objectId="OID-GOV">

  <!-- Vote count for John Brown -->
  <BallotSelection xsi:type="CandidateSelection" objectId="OID-C1">
    <CandidateId>OID-C1</CandidateId>
    <VoteCountsCollection>
      <VoteCounts>
        <Count>2716</Count>
      </VoteCounts>
    </VoteCountsCollection>
  </BallotSelection>

  <!-- Vote count for Thomas Paine -->
  <BallotSelection xsi:type="CandidateSelection" objectId="OID-C2">
    <CandidateId>OID-C2</CandidateId>
    <VoteCountsCollection>
      <VoteCounts>
        <Count>2392</Count>
      </VoteCounts>
    </VoteCountsCollection>
  </BallotSelection>

  <ElectoralDistrictId>OID-State</ElectoralDistrictId>
  <Name>Governor</Name>
  <VotesAllowed>1</VotesAllowed>

</Contest>
```

Figure 15 - Basic Contest Definition and Linkage to Vote Counts

Contests for *BallotMeasureContest* and *PartyContest* are implemented as per this example.

5.6 Associating Vote Counts with Contests

The ElectionResultsReporting schema includes the capability to associate vote counts with contests in a variety of ways, including the following:

1. As aggregated vote counts (applying to the entire electoral district)
 - The vote counts can be broken out (or filtered by) ballot type, device type, and by specific device and manufacturer.
2. As subsets of vote counts, each associated with specific geo-political units within the electoral district such as precincts or split precincts
 - The vote counts can be broken out by ballot type, device type, and by specific device and manufacturer.
3. As summary vote counts that can be aggregated as well as associated with specific geo-political units.
 - The summaries can be broken out by ballot type, device type, and by specific device and manufacturer.
 - The summaries can include:
 - Total number of ballots cast containing the contest.
 - Total number of overvotes and undervotes in the contest.
 - Total number of write-ins in the contest.

The following sections go into more detail on how to report counts accordingly.

5.6.1 Associating Aggregated Vote Counts

The previous section showed an example of basic contest definition involving aggregated vote counts applying to the entire electoral district. To go into greater level of detail and report the aggregated counts by ballot type or device type, one includes additional values from the `<CountItemType>` or `<DeviceType>` enumerations. These enumerations specify different ballot types or device items that can be used as, essentially, filters on the aggregated results. An example of this is as follows, using the contest for governor and values from the `<CountItemType>` enumeration:

```
<!-- Governor contest -->
<Contest xsi:type="CandidateContest" objectId="OID-GOV">

  <!-- Vote counts for John Brown broken down by ballot type -->
  <BallotSelection xsi:type="CandidateSelection" objectId="OID-C1">

    <CandidateId>OID-C1</CandidateId>

    <VoteCountsCollection>
      <VoteCounts>
        <BallotType>election-day</BallotType>
        <Count>2716</Count>
      </VoteCounts>
      <VoteCounts>
        <BallotType>early</BallotType>
        <Count>700</Count>
      </VoteCounts>
    </VoteCountsCollection>
  </BallotSelection>
</Contest>
```



```

        </VoteCounts>
        <VoteCounts>
            <BallotType>absentee</BallotType>
            <Count>16</Count>
        </VoteCounts>
    </VoteCountsCollection>

</BallotSelection>

</Contest>

```

Figure 16 - Linking Aggregated Vote Counts to Contests

This example answers the questions, how many votes did John Brown get for the contest as a whole, filtered by ballots cast on election day, by ballots cast during the early voting period, or by absentee ballots.

5.6.2 Associating Vote Counts Broken Down by Precincts plus Ballot and Device Type

For associating vote counts broken down by geo-political units such as precincts and split precincts, one uses the *<GpUnitId>* sub-element to link the counts to *<GpUnit>* elements defined for, the precincts and split precincts. For example, to associate the vote count for John Brown by ward in Figure 12, the XML is:

```

<!-- Governor contest -->
<Contest xsi:type="CandidateContest" objectId="OID-GOV">

    <!-- Vote count for John Brown broken down by ward -->
    <BallotSelection xsi:type="CandidateSelection" objectId="OID-C1">

        <CandidateId>OID-C1</CandidateId>

        <VoteCountsCollection>
            <VoteCounts>
                <Count>627</Count>
                <GpUnitId>OID-WARD1-1</GpUnitId>
            </VoteCounts>
            <VoteCounts>
                <Count>325</Count>
                <GpUnitId>OID-WARD1-2</GpUnitId>
            </VoteCounts>

            <VoteCounts>
                <Count>432</Count>
                <GpUnitId>OID-WARD2</GpUnitId>
            </VoteCounts>

        </VoteCountsCollection>

    </BallotSelection>

```

Figure 17 - Linking Vote Counts to Precincts

In this example, note that the vote counts for John Brown are broken down, in Ward 1, for Ward 1's splits (as opposed to breaking it down by Ward 1 as a whole). Since all of Ward 1 is in the electoral district for the Governor's contest, it is not necessary to break down the vote count by the split precincts of Ward 1, but doing so would answer questions such as, "How did the voters in the New Richmond or Somerset school districts vote in the Governor's contest?"

Note also that WI generally reports by combined precinct, thus the example in Figure 17 is not technically correct. Reporting by combined precinct, however, is done exactly as with precincts and split precincts.

To break the vote counts down further by ballot type or device type, one again uses values from the `<CountItemType>` or `<DeviceType>` enumerations. Using Ward 2 from the previous example, the XML is:

```
<!-- Governor contest -->
<Contest xsi:type="CandidateContest" objectId="OID-GOV">

  <!-- Vote count for John Brown broken down by ward -->
  <BallotSelection xsi:type="CandidateSelection" objectId="OID-C1">

    <CandidateId>OID-C1</CandidateId>

    <VoteCountsCollection>
      <VoteCounts>
        <BallotType>election-day</BallotType>
        <Count>300</Count>
        <Device><Type>opscan-precinct</Type></Device>
        <GpUnitId>OID-WARD2</GpUnitId>
      </VoteCounts>

      <VoteCounts>
        <BallotType>early</BallotType>
        <Count>132</Count>
        <Device><Type>opscan-precinct</Type></Device>
        <GpUnitId>OID-WARD2</GpUnitId>
      </VoteCounts>
    </VoteCountsCollection>

  </BallotSelection>
```

Figure 18 - Filtering Vote Counts by Device and Ballot Type

Thus, using the two examples, John Brown received 300 votes in Ward 1 on election day using precinct-count optical scanner. He received 132 votes during the early voting period, again using a precinct-count optical scanner.

5.6.3 Associating Summary Vote and Ballot Counts

It is possible to associate summaries of vote and ballot counts for the contest for various purposes including for reconciliations and checks on the contest. These summaries include:

- Total number of ballots cast containing the contest.
- Total number of overvotes and undervotes in the contest.
- Total number of write-ins in the contest.

As with previous examples, it is possible to further filter these counts by ballot type and device type, using values from the `<CountItemType>` or `<DeviceType>` enumerations. It is also possible to link the counts to specific geo-political units such as precincts and split precincts by including the `<GpUnitId>` sub-element to link the counts to `<GpUnit>` elements defined for, the precincts and split precincts.

Following is an example of summary counts for a contest using `SummaryCounts` to report summary vote counts by ballot type and further broken down by ward:

```
<Contest xsi:type="CandidateContest" objectId="OID-GOV">
  <BallotSelection ...
  ...
  <ElectoralDistrictId>OID-State</ElectoralDistrictId>
  <Name>Governor</Name>
  ...
  ...
  <SummaryCounts>
    <BallotsCast>888033</BallotsCast>
    <Overvotes>6013</Overvotes>
    <Undervotes>1000</Undervotes>
    <GpUnit>OID-WARD3</GpUnit>
  </SummaryCounts>
  <SummaryCounts>
    <BallotsCast>8506</BallotsCast>
    <Overvotes>133</Overvotes>
    <Undervotes>10</Undervotes>
    <GpUnit>OID-WARD3</GpUnit>
  </SummaryCounts>
  ...
</Contest>
```

Figure 19 - Associating Summary Counts with Contests

Some jurisdictions report items such as overvotes and undervotes as candidates in a contest as opposed to reporting them separately as a summary of counts. This schema does not prevent one from treating overvotes, undervotes, or other types of counts as candidates in a contest, thus one would report their specific vote counts as shown in section 5.6.1.

5.7 Associating Summary Ballot Counts with Geo-political Units

Lastly, it is possible to associate the ballots cast in each geo-political unit and filter them by ballot types and device types by, again, including the `<SummaryCounts>` element. The summary counts include:

- Total number of ballots cast at that geo-political unit
- Total number of overvotes and undervotes on all ballots cast at that geo-political unit

- Total number of write-ins on all ballots cast at that geo-political unit
- Summary ballot counts for the geo-political unit broken out by
 - the type of ballot
 - the type of device
 - the specific device serial number and manufacturer

Thus, one can associate summary counts for each precinct, etc., and, as possible, go down to the level of specific devices. To do this, the *<SummaryCounts>* element needs to be added to each *<GpUnit>* definition, as follows:

```
<GpUnit xsi:type="ReportingUnit" objectId="OID-WARD5">
  <Name>Ward 5</Name>

  <SummaryCounts>
    <BallotsCast>8033</BallotsCast>
    <Overvotes>113</Overvotes>
    <Undervotes>1010</Undervotes>
  </SummaryCounts>
  <SummaryCounts>
    <Device><Type>opscan-precinct</Type></Device>
    <BallotsCast>2013</BallotsCast>
    <Overvotes>100</Overvotes>
    <Undervotes>1000</Undervotes>
  </SummaryCounts>
  <SummaryCounts>
    <Device>
      <Type>electronic</Type>
    </Device>
    <BallotsCast>6020</BallotsCast>
    <Overvotes>13</Overvotes>
    <Undervotes>10</Undervotes>
  </SummaryCounts>

  <Type>precinct</Type>
</GpUnit>
```

Figure 20 - Associating Summary Ballot Counts with <GpUnit> Elements

In this example, Ward 5 had a total of 8033 ballots cast, with 113 overvotes and 1010 undervotes on all ballots. The count is then broken down by device type, with 2013 ballots cast on precinct optical scan and 6020 ballots cast on an all-electronic device such as a DRE.

Appendix A—Acronyms

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

CDF	Common Data Format
EAC	Election Assistance Commission
EMS	Election Management System
FIPS	Federal Information Processing Standard
FWAB	Federal Write-in Absentee Ballot
OCD-ID	Open Civic Data Identifiers
UML	Unified Modeling Language
UOCAVA	Uniform and Overseas Citizens Assistance in Voting Act
VIP	Voting Information Project
VVSG	Voluntary Voting Systems Guidelines
XML	Extensible Markup Language

Appendix B—Glossary

Selected terms used throughout this document are defined below. In some of the definitions, there is ancillary information that is not part of the definition but helpful in understanding the definition; this ancillary information is preceded with “*Note:*”.

Absentee ballot:	Ballot resulting from absentee voting.
Absentee voting:	Voting that can occur unsupervised at a location chosen by the voter and usually done ahead of election day. <i>Note:</i> In some jurisdictions, absentee voting is also called early voting and vote by mail.
Affiliation:	Association with a political party. <i>See also:</i> endorsement. <i>Note:</i> Affiliation with a political party does not imply endorsement by that political party; endorsement does not imply affiliation.
Ballot measure:	Contest in which the choices are typically Yes and No. <i>Syn:</i> referendum.
Ballot rotation:	The process of varying the order of the contest choices within a given contest.
Ballot style:	The list of contests and candidates associated with a particular ballot and its associated precinct or split precinct (and party, for some primaries), including language used and the ordering of contests and candidates. <i>Note:</i> In closed primaries, the same collection of ballot style layouts are used within the same ballot style geography, with the specific ballot style layout depending on the voter's affiliation.
Closed primary:	Primary election in which only voters registered with a political party may vote in those party-specific contests associated with that party.
Combined precinct:	Two or more precincts assigned the same polling place. <i>Syn:</i> consolidated precinct.
Contest:	A single decision being put before the voters (e.g., the selection of candidates to fill a particular public office or the approval or disapproval of a constitutional amendment).
Cross-party endorsement:	Endorsement of a given contest choice by two or more political parties.
Direct record electronic (DRE):	An electronic vote-capture device that provides choices that are visible to the voter on a front panel of the machine in which voters directly enter choices into electronic storage with the use of a touch-screen, push-buttons, or similar device. <i>Note:</i> An alphabetic keyboard is often provided with the entry device to allow for the

possibility of write-in votes. The voter's choices are stored in these machines and added to the choices of all other voters.

Early voting:

Voting that occurs prior to election day at a polling location under the supervision of poll workers or election administrative staff. *See also:* in-person voting. *Note:* some jurisdictions, early voting is referred to as in-person absentee voting.

Election certification:

The process of certifying the results of an election, including absentee or early votes, votes cast on election day, and valid provisional ballots.

Election day:

The date on which the election is considered held. Absentee votes and early votes may be cast in advance of election day. *Note:* Some jurisdictions have an election period of several days or weeks for in-person or remote voting, and there is no one day that is election day.

Election night:

The period of time starting when the polls close in a jurisdiction in which results may first begin to be reported and ending when all precincts have reported.

Election management system (EMS):

Computer systems used to perform such tasks as preparing ballots, setting up tally systems, maintaining voter registration information, generating reports, and to consolidate, report, and display election results. *Note:* This device receives results data from the vote-capture devices or by manual input, accumulates the results, and reports the accumulated results.

Election official:

Any county clerk and recorder, election judge, member of a canvassing board, member of a board of county commissioners, member or secretary of a board of directors authorized to conduct public elections, representative of a governing body, or other person contracting for or engaged in the performance of election duties as required by the election code.

Electoral District:

As used in elections, an administrative division in which voters are entitled to vote in contests that are specific to that division, such as those for state senators and delegates.

Endorsement:

Approval by a political party (e.g., as the candidate that the party elects to field in a particular contest and/or as the candidate that should receive straight party votes). *See also:* affiliation. *Note:* In some states, more than one party may endorse a contest selection.

General election:

Election in which the candidates for contests and offices have generally been chosen during a primary election. *Note:* In presidential systems, the term refers to a regularly scheduled election where both the president, and either "a class" of or all members of

the national legislature are elected at the same time. A general election day may also include elections for local officials.

In-person voting:

Voting that occurs at a polling place under the supervision of poll workers on election day. *See also:* early voting. *Syn:* polling place voting.

Jurisdiction:

Term as used in election contexts to signify a geographical area to which a practical authority has been granted to administer elections for political offices. Areas of jurisdiction apply to local, state, and federal levels. *Note:* States, counties, cities, towns and townships are all examples of jurisdictions.

Municipality:

Term as used in election contexts to signify a jurisdiction such as city or town that has some form of local government for which elections are generally conducted.

N-of-M:

Voting variation in which the voter is entitled to allocate a fixed number of votes (N) over a list of M contest choices or write-ins, with the constraint that at most 1 vote may be allocated to a given contest choice. *See also:* approval voting, cumulative voting. *Note:* This usually occurs when multiple seats are concurrently being filled in a governing body such as a city council or school board where candidates run at-large. The voter is not obliged to allocate all N votes. 1-of-M is N-of-M voting where $N = 1$.

Nonpartisan primary:

Election held to narrow the field of candidates in non-party-specific contests prior to the general election. *Note:* A primary election may be comprised of a nonpartisan primary for some contests or measures and a partisan and/or open primary for others.

Open primary:

Primary held in a state where voters do not register as a party member. *Note:* There are two variations. In one, the voter declares a choice of party ballot to the pollworker and is issued a ballot containing only contests for that party (and nonpartisan contests). In the other, the ballot issued contains all eligible contest from all parties and the voter selects the party of choice, privately, by only selecting candidates in contests of the desired party. Selections in more than one party void the partisan section of the ballot.

Overvote:

Occurs when a voter selects more than one candidate in a 1-of-M contest or more than N candidates in an N-of-M contest. The vote for that contest is considered an overvote and not counted towards any candidate in that contest (unless approval voting applies for that contest). *Note:* Usually the rest of a properly marked ballot is counted. Large numbers of overvotes can be indicative of confusing ballot layout or confusing instructions.

Partisan primary:	Election held to determine which candidate(s) will represent a political party for particular offices in the general election.
Polling place:	Location at which voters cast their ballots in-person on vote-capture devices (e.g., DRE) under the supervision of poll workers usually on election day. <i>Syn:</i> polling station or poll. <i>Note:</i> A polling place is typically in 1-to-1 correspondence with a precinct except for combined precincts and vote centers.
Precinct:	An election administration division corresponding to a contiguous geographic area that is the basis for determining the contests and measures on which the voters legally residing in that area are eligible to vote. <i>Syn:</i> a beat, box, polling district, ward. <i>See also:</i> combined precinct, election district, split precinct.
Primary election:	Election generally held to determine which candidate(s) will represent a political party for particular offices in the general election and/or to narrow the field of candidates in non-party-specific contests prior to the general election. In some cases, such as for local and state central committee members for a particular party, certain contests serve as the sole election as opposed to a primary followed by a general election. <i>Note:</i> From the functional viewpoint of the voting system, the defining features of a primary election are the presence of party-specific contests and a requirement to report separate totals for the different political parties.
Provisional ballot:	Ballot cast by a voter whose eligibility to vote is disputed by an election official. <i>Syn:</i> a challenged ballot.
Reporting unit:	An administrative division that reports votes or to which votes are associated, e.g., state, county, city, precinct, etc.
Schema:	A file containing definitions of data elements and attributes with rules for usage, e.g., for XML.
Split precinct:	Precinct serving voters from two or more administrative divisions, such as election districts, that may require different ballot styles, other than for primaries. <i>Syn:</i> split.
Straight party voting:	Voting variation in a general election in which a vote in a designated, special contest (in which the choices are political parties) implies votes in accordance with the endorsements of the selected party in all other contests on the ballot in which straight party voting is allowed. <i>Note:</i> There are different tabulation rules for processing paper ballots containing a straight party selection and direct selections of one or more candidates in individual contests that are state specific.

Tabulator:	A programmed device that counts votes.
Town:	An urban area that has a name, defined boundaries, and local government, and that is generally larger than a village and smaller than a city.
Township:	A subdivision of a county in most northeast and Midwest U.S. states, having the status of a unit of local government with varying governmental powers. <i>Syn:</i> civil township.
UOCAVA voter:	From the Uniform and Overseas Citizens Assistance in Voting Act (UOCAVA); A U.S. citizen who is an active member of the Uniformed Services and the Merchant Marine, or the commissioned corps of the Public Health Service or the National Oceanic and Atmospheric Administration, their eligible family members, and U.S. citizens residing outside the United States.
Undervote:	Occurs when the voter does not select a candidate in a 1-of-M contest or selects fewer than N candidates in a N-of-M contest. <i>Note:</i> can indicate a conscious choice of the voter to not vote in the contest. As with overvotes, large numbers of undervotes can be indicative of confusing ballot layout or confusing instructions.
Vote-capture device:	An automated device that is used directly by a voter to cast a ballot. <i>See also:</i> direct record electronic (DRE).
Vote center:	A polling place where voters from multiple precincts may cast their ballots. <i>Syn:</i> a super precinct. <i>See also:</i> combined precinct.
Voter turnout:	The number of voters who have cast ballots as a percentage of the total number of voters who can cast ballots. <i>Note:</i> Various states calculate this differently, sometimes using the total number of potentially eligible voters whether registered or not.
Voting variation:	A contest option or feature, e.g., approval voting, cumulative voting.
Write-in:	Vote for a candidate who is explicitly named by the voter in lieu of choosing a candidate who is already listed on the ballot. <i>Note:</i> In some states, this does not preclude writing in the name of a candidate who is already listed on the ballot. Some states require write-in candidates to be registered prior to the elections to be considered a valid write-in selection.

Appendix C—References

- [1] Voluntary Voting System Guidelines, Version 1.0,
http://www.eac.gov/testing_and_certification/2005_vvsg.aspx
- [2] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C
Recommendation November 26, 2008, <http://www.w3.org/TR/xml/>
- [3] UML Specification version 1.1 (OMG document ad/97-08-11) 2011-09-22,
<http://omg.org/>
- [4] Election Assistance Commission, Election Administration and Voting
Survey 2014,
http://www.eac.gov/research/election_administration_and_voting_survey.aspx
- [5] Florida Election Results Summary, XML Schema Reference Version 3.0-S,
July 2015, <http://dos.myflorida.com/media/695297/ds-de-137-summary.pdf>
- [6] Voting Information Project (VIP), PEW Center on the States,
[http://www.pewcenteronthestates.org/uploadedFiles/Voting_Information_Pr
oject_brief.pdf](http://www.pewcenteronthestates.org/uploadedFiles/Voting_Information_Project_brief.pdf)
- [7] OASIS EML Version 7 Technical Specification, [https://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=election](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=election)
- [8] Cambridge Election Commission, 51 Inman Street, First Floor, Cambridge,
MA 02139, <https://www.cambridgema.gov/GIS/mapgallery>
- [9] Hilkert, Scott, Managing Principal and COO, Catalyst Consulting Group,
Inc., “Application of Lessons Learned in the Illinois EDC Project Toward
Emerging Election Data Standards and System Guidelines”
[http://grouper.ieee.org/groups/1622/WorkingDocuments/workshop-2009-10-
NIST/hilkert-Lessons-Learned-in-Illinois-EDC-1-1.pdf](http://grouper.ieee.org/groups/1622/WorkingDocuments/workshop-2009-10-NIST/hilkert-Lessons-Learned-in-Illinois-EDC-1-1.pdf)
- [10] XML Signature Syntax and Processing (Second Edition), W3C
Recommendation June 10, 2008, <http://www.w3.org/TR/xmlsig-core/>
- [11] American National Standards Institute (ANSI) Codes,
<https://www.census.gov/geo/reference/ansi.html>
- [12] Open Civic Data, <http://opencivicdata.readthedocs.org/en/latest/ocdids.html>

Appendix D—File Download Locations

The files associated with this specification are available for download from a NIST repository. These files are:

- This specification
- ElectionResultsReporting XML schema
- Example XML files
- Validation tools
- ElectionResultsReporting UML model

Other files or updates to the files may be added. The repository URL is:

<http://vote.nist.gov/ElectionResultsReporting>

Appendix E—ElectionResultsReporting XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Version 1.0-49, September 4, 2015, NIST ElectionResultsReporting, National Institute of
Standards and Technology -->
<xsd:schema xmlns="NIST_V1_election_results_reporting.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="NIST_V1_election_results_reporting.xsd" elementFormDefault="qualified"
version="1.0">
  <!-- ===== Imports ===== -->
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/2000/09/xmldsig#"/>
  <!-- ===== Roots ===== -->
  <xsd:element name="ElectionReport" type="ElectionReport"/>
  <!-- ===== Primitives ===== -->
  <xsd:simpleType name="HtmlColorString">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9a-f]{6}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ShortString">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="16"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="TimeWithZone">
    <xsd:restriction base="xsd:time">
      <xsd:pattern value="((([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]|(24:00:00))(Z|[-+]?((0[0-
9]|1[0-3]):[0-5][0-9]|14:00)))"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- ===== Enumerations ===== -->
  <xsd:simpleType name="BallotMeasureType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ballot-measure"/>
      <xsd:enumeration value="initiative"/>
      <xsd:enumeration value="referendum"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="CandidatePostElectionStatus">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="advanced-to-runoff"/>
      <xsd:enumeration value="projected-winner"/>
      <xsd:enumeration value="winner"/>
      <xsd:enumeration value="withdrawn"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="CandidatePreElectionStatus">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="filed"/>
      <xsd:enumeration value="qualified"/>
      <xsd:enumeration value="withdrawn"/>
      <xsd:enumeration value="write-in"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="CountItemStatus">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="completed"/>
      <xsd:enumeration value="in-process"/>
      <xsd:enumeration value="not-processed"/>
      <xsd:enumeration value="unknown"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

</xsd:simpleType>
<xsd:simpleType name="CountItemType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="absentee"/>
    <xsd:enumeration value="absentee-fwab"/>
    <xsd:enumeration value="absentee-in-person"/>
    <xsd:enumeration value="absentee-mail"/>
    <xsd:enumeration value="early"/>
    <xsd:enumeration value="election-day"/>
    <xsd:enumeration value="provisional"/>
    <xsd:enumeration value="total"/>
    <xsd:enumeration value="uocava"/>
    <xsd:enumeration value="write-in"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DayType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="all"/>
    <xsd:enumeration value="sunday"/>
    <xsd:enumeration value="monday"/>
    <xsd:enumeration value="tuesday"/>
    <xsd:enumeration value="wednesday"/>
    <xsd:enumeration value="thursday"/>
    <xsd:enumeration value="friday"/>
    <xsd:enumeration value="saturday"/>
    <xsd:enumeration value="weekday"/>
    <xsd:enumeration value="weekend"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DeviceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="electronic"/>
    <xsd:enumeration value="lever"/>
    <xsd:enumeration value="manual-count"/>
    <xsd:enumeration value="mixed-systems"/>
    <xsd:enumeration value="opscan-central"/>
    <xsd:enumeration value="opscan-precinct"/>
    <xsd:enumeration value="punch-card"/>
    <xsd:enumeration value="unknown"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ElectionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="general"/>
    <xsd:enumeration value="partisan-primary-closed"/>
    <xsd:enumeration value="partisan-primary-open"/>
    <xsd:enumeration value="primary"/>
    <xsd:enumeration value="runoff"/>
    <xsd:enumeration value="special"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="GeoSpatialFormat">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="geo-json"/>
    <xsd:enumeration value="gml"/>
    <xsd:enumeration value="kml"/>
    <xsd:enumeration value="shp"/>
    <xsd:enumeration value="wkt"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="IdentifierType">

```

```

    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="fips"/>
      <xsd:enumeration value="local-level"/>
      <xsd:enumeration value="national-level"/>
      <xsd:enumeration value="ocd-id"/>
      <xsd:enumeration value="state-level"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="OfficeTermType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="full-term"/>
      <xsd:enumeration value="unexpired-term"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ReportDetailLevel">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="precinct-level"/>
      <xsd:enumeration value="summary-contest"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ReportingUnitType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ballot-batch"/>
      <xsd:enumeration value="city"/>
      <xsd:enumeration value="city-council"/>
      <xsd:enumeration value="combined-precinct"/>
      <xsd:enumeration value="congressional"/>
      <xsd:enumeration value="county"/>
      <xsd:enumeration value="county-council"/>
      <xsd:enumeration value="drop-box"/>
      <xsd:enumeration value="judicial"/>
      <xsd:enumeration value="municipality"/>
      <xsd:enumeration value="polling-place"/>
      <xsd:enumeration value="precinct"/>
      <xsd:enumeration value="school"/>
      <xsd:enumeration value="special"/>
      <xsd:enumeration value="split-precinct"/>
      <xsd:enumeration value="state"/>
      <xsd:enumeration value="state-house"/>
      <xsd:enumeration value="state-senate"/>
      <xsd:enumeration value="town"/>
      <xsd:enumeration value="township"/>
      <xsd:enumeration value="utility"/>
      <xsd:enumeration value="village"/>
      <xsd:enumeration value="vote-center"/>
      <xsd:enumeration value="ward"/>
      <xsd:enumeration value="water"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ResultsStatus">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="certified"/>
      <xsd:enumeration value="correction"/>
      <xsd:enumeration value="pre-election"/>
      <xsd:enumeration value="recount"/>
      <xsd:enumeration value="unofficial-complete"/>
      <xsd:enumeration value="unofficial-partial"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="VoteVariation">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="1-of-m"/>

```

```

        <xsd:enumeration value="approval"/>
        <xsd:enumeration value="borda"/>
        <xsd:enumeration value="cumulative"/>
        <xsd:enumeration value="majority"/>
        <xsd:enumeration value="n-of-m"/>
        <xsd:enumeration value="plurality"/>
        <xsd:enumeration value="proportional"/>
        <xsd:enumeration value="range"/>
        <xsd:enumeration value="rcv"/>
        <xsd:enumeration value="super-majority"/>
        <xsd:enumeration value="other"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- ===== Interfaces Defined ===== -->
<!-- ===== Interfaces Extended ===== -->
<!-- ===== Classes ===== -->
<xsd:complexType name="AnnotatedString">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="annotation" type="ShortString"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="BallotMeasureContest">
    <xsd:complexContent>
        <xsd:extension base="Contest">
            <xsd:sequence>
                <xsd:element name="ConStatement" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="EffectOfAbstain" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="FullText" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="InfoUri" type="xsd:anyURI" minOccurs="0"/>
                <xsd:element name="PassageThreshold" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="ProStatement" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="SummaryText" type="InternationalizedText" minOccurs="0"/>
                <xsd:element name="Type" type="BallotMeasureType" minOccurs="0"/>
                <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BallotMeasureSelection">
    <xsd:complexContent>
        <xsd:extension base="BallotSelection">
            <xsd:sequence>
                <xsd:element name="Selection" type="InternationalizedText"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BallotSelection" abstract="true">
    <xsd:sequence>
        <xsd:element name="SequenceOrder" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="VoteCountsCollection" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="VoteCounts" type="VoteCounts" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="BallotStyle">
    <xsd:sequence>

```



```

        <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
        <xsd:element name="GpUnitId" type="xsd:IDREF" maxOccurs="unbounded"/>
        <xsd:element name="ImageUri" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="OrderedContest" type="OrderedContest" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="Candidate">
    <xsd:sequence>
        <xsd:element name="BallotName" type="InternationalizedText"/>
        <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
        <xsd:element name="FileDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="IsIncumbent" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="IsTopTicket" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"/>
        <xsd:element name="PersonId" type="xsd:IDREF" minOccurs="0"/>
        <xsd:element name="PostElectionStatus" type="CandidatePostElectionStatus"
minOccurs="0"/>
        <xsd:element name="PreElectionStatus" type="CandidatePreElectionStatus" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="CandidateContest">
    <xsd:complexContent>
        <xsd:extension base="Contest">
            <xsd:sequence>
                <xsd:element name="NumberElected" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="PrimaryPartyId" type="xsd:IDREF" minOccurs="0"/>
                <xsd:element name="VotesAllowed" type="xsd:integer"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CandidateSelection">
    <xsd:complexContent>
        <xsd:extension base="BallotSelection">
            <xsd:sequence>
                <xsd:element name="CandidateId" type="xsd:IDREF" maxOccurs="unbounded"/>
                <xsd:element name="EndorsementPartyId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="IsWriteIn" type="xsd:boolean" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Coalition">
    <xsd:complexContent>
        <xsd:extension base="Party">
            <xsd:sequence>
                <xsd:element name="ContestId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ContactInformation">
    <xsd:sequence>
        <xsd:element name="AddressLine" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Email" type="AnnotatedString" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>

```

```

        <xsd:element name="Fax" type="AnnotatedString" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Phone" type="AnnotatedString" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Schedule" type="Schedule" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Uri" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="Contest" abstract="true">
    <xsd:sequence>
        <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
        <xsd:element name="BallotSelection" type="BallotSelection" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="BallotSubTitle" type="InternationalizedText" minOccurs="0"/>
        <xsd:element name="BallotTitle" type="InternationalizedText" minOccurs="0"/>
        <xsd:element name="CountStatus" type="CountStatus" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="ElectoralDistrictId" type="xsd:IDREF"/>
        <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
        <xsd:element name="HasRotation" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="Name" type="xsd:string"/>
        <xsd:element name="SequenceOrder" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="SubUnitsReported" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="SummaryCounts" type="SummaryCounts" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="TotalSubUnits" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="VoteVariation" type="VoteVariation" minOccurs="0"/>
        <xsd:element name="OtherVoteVariation" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="CountStatus">
    <xsd:sequence>
        <xsd:element name="Status" type="CountItemStatus"/>
        <xsd:element name="Type" type="CountItemType"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Counts" abstract="true">
    <xsd:sequence>
        <xsd:element name="Device" type="Device" minOccurs="0"/>
        <xsd:element name="GpUnitId" type="xsd:IDREF" minOccurs="0"/>
        <xsd:element name="IsSuppressedForPrivacy" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="Type" type="CountItemType" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Device">
    <xsd:sequence>
        <xsd:element name="Manufacturer" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Model" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="DeviceType" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Election">
    <xsd:sequence>
        <xsd:element name="BallotStyleCollection" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="BallotStyle" type="BallotStyle" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="CandidateCollection" minOccurs="0">

```

```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Candidate" type="Candidate" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="ContactInformation" type="ContactInformation" minOccurs="0"/>
      <xsd:element name="ContestCollection" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Contest" type="Contest" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="CountStatus" type="CountStatus" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ElectionScopeId" type="xsd:IDREF"/>
      <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
      <xsd:element name="Name" type="InternationalizedText"/>
      <xsd:element name="StartDate" type="xsd:date"/>
      <xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
      <xsd:element name="Type" type="ElectionType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ElectionAdministration">
    <xsd:sequence>
      <xsd:element name="ContactInformation" type="ContactInformation" minOccurs="0"/>
      <xsd:element name="ElectionOfficialPersonId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ElectionReport">
    <xsd:sequence>
      <xsd:element name="Election" type="Election" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
      <xsd:element name="Format" type="ReportDetailLevel"/>
      <xsd:element name="GeneratedDate" type="xsd:dateTime"/>
      <xsd:element name="GpUnitCollection" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="GpUnit" type="GpUnit" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="Issuer" type="xsd:string"/>
      <xsd:element name="IssuerAbbreviation" type="xsd:string"/>
      <xsd:element name="IsTest" type="xsd:boolean" minOccurs="0"/>
      <xsd:element name="Notes" type="xsd:string" minOccurs="0"/>
      <xsd:element name="OfficeCollection" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Office" type="Office" maxOccurs="unbounded"/>
            <xsd:element name="OfficeGroup" type="OfficeGroup" minOccurs="0"
maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="PartyCollection" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Party" type="Party" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

    <xsd:element name="PersonCollection" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Person" type="Person" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="SequenceStart" type="xsd:integer"/>
    <xsd:element name="SequenceEnd" type="xsd:integer"/>
    <xsd:element name="Status" type="ResultsStatus"/>
    <xsd:element name="TestType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="VendorApplicationId" type="xsd:string"/>
    <xsd:element ref="ds:Signature" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ExternalIdentifiers">
  <xsd:sequence>
    <xsd:element name="ExternalIdentifier" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Type" type="IdentifierType"/>
          <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
          <xsd:element name="Value" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="label" type="xsd:string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="GpUnit" abstract="true">
  <xsd:sequence>
    <xsd:element name="ComposingGpUnitId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="SummaryCounts" type="SummaryCounts" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="Hours">
  <xsd:sequence>
    <xsd:element name="Day" type="DayType" minOccurs="0"/>
    <xsd:element name="StartTime" type="TimeWithZone"/>
    <xsd:element name="EndTime" type="TimeWithZone"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="InternationalizedText">
  <xsd:sequence>
    <xsd:element name="Text" type="LanguageString" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="LanguageString">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="language" type="xsd:language" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Office">
  <xsd:sequence>

```

```

<xsd:element name="ContactInformation" type="ContactInformation" minOccurs="0"/>
<xsd:element name="ElectoralDistrictId" type="xsd:IDREF" minOccurs="0"/>
<xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
<xsd:element name="FilingDeadline" type="xsd:date" minOccurs="0"/>
<xsd:element name="IsPartisan" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="Name" type="InternationalizedText"/>
<xsd:element name="OfficeHolderPersonId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="Term" minOccurs="0">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="StartDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="Type" type="OfficeTermType" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="label" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
<xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="OfficeGroup">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SubOfficeGroup" type="OfficeGroup" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="OrderedContest">
  <xsd:sequence>
    <xsd:element name="ContestId" type="xsd:IDREF"/>
    <xsd:element name="OrderedBallotSelectionId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Party">
  <xsd:sequence>
    <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Color" type="HtmlColorString" minOccurs="0"/>
    <xsd:element name="ExternalIdentifiers" type="ExternalIdentifiers" minOccurs="0"/>
    <xsd:element name="LogoUri" type="xsd:anyURI" minOccurs="0"/>
    <xsd:element name="Name" type="InternationalizedText"/>
  </xsd:sequence>
  <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="PartyContest">
  <xsd:complexContent>
    <xsd:extension base="Contest"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PartyRegistration">
  <xsd:sequence>
    <xsd:element name="Count" type="xsd:integer"/>
    <xsd:element name="PartyId" type="xsd:IDREF"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PartySelection">
  <xsd:complexContent>
    <xsd:extension base="BallotSelection">
      <xsd:sequence>
        <xsd:element name="PartyId" type="xsd:IDREF" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Person">
    <xsd:sequence>
        <xsd:element name="ContactInformation" type="ContactInformation" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="DateOfBirth" type="xsd:date" minOccurs="0"/>
        <xsd:element name="FirstName" type="xsd:string" minOccurs="0"/>
        <xsd:element name="FullName" type="InternationalizedText" minOccurs="0"/>
        <xsd:element name="Gender" type="xsd:string" minOccurs="0"/>
        <xsd:element name="LastName" type="xsd:string" minOccurs="0"/>
        <xsd:element name="MiddleName" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Nickname" type="xsd:string" minOccurs="0"/>
        <xsd:element name="PartyId" type="xsd:IDREF" minOccurs="0"/>
        <xsd:element name="Prefix" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Profession" type="InternationalizedText" minOccurs="0"/>
        <xsd:element name="Suffix" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Title" type="InternationalizedText" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="objectId" type="xsd:ID" use="required"/>
</xsd:complexType>
<xsd:complexType name="ReportingDevice">
    <xsd:complexContent>
        <xsd:extension base="GpUnit">
            <xsd:sequence>
                <xsd:element name="Device" type="Device" minOccurs="0"/>
                <xsd:element name="SerialNumber" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ReportingUnit">
    <xsd:complexContent>
        <xsd:extension base="GpUnit">
            <xsd:sequence>
                <xsd:element name="AuthorityId" type="xsd:IDREF" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="ContactInformation" type="ContactInformation" minOccurs="0"/>
                <xsd:element name="CountStatus" type="CountStatus" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RetentionContest">
    <xsd:complexContent>
        <xsd:extension base="BallotMeasureContest">
            <xsd:sequence>
                <xsd:element name="CandidateId" type="xsd:IDREF"/>
                <xsd:element name="OfficeId" type="xsd:IDREF" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Schedule">
    <xsd:sequence>
        <xsd:element name="Hours" type="Hours" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="IsOnlyByAppointment" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="IsOrByAppointment" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="IsSubjectToChange" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="StartDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="EndDate" type="xsd:date" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="SpatialDimension">
    <xsd:sequence>
        <xsd:element name="MapUri" type="xsd:anyURI" minOccurs="0"/>
        <xsd:element name="SpatialExtent" type="SpatialExtent" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SpatialExtent">
    <xsd:sequence>
        <xsd:element name="Coordinates" type="xsd:string"/>
        <xsd:element name="Format" type="GeoSpatialFormat"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SummaryCounts">
    <xsd:complexContent>
        <xsd:extension base="Counts">
            <xsd:sequence>
                <xsd:element name="BallotsCast" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="BallotsOutstanding" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="BallotsRejected" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="Overvotes" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="Undervotes" type="xsd:integer" minOccurs="0"/>
                <xsd:element name="WriteIns" type="xsd:integer" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="VoteCounts">
    <xsd:complexContent>
        <xsd:extension base="Counts">
            <xsd:sequence>
                <xsd:element name="Count" type="xsd:float"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```