

NIST Special Publication 1500-102

Voter Records Interchange Common Data Format Specification

Version 1.0

John P. Wack, editor
Sam Dana
John Dziurlaj
E. John Sebes
Sarah Whitt

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.SP.1500-102>



NIST Special Publication 1500-102

Voter Records Interchange Common Data Format Specification

Version 1.0

John P. Wack, NIST, editor

Sam Dana, Prometheus Computing

John Dziurlaj, Democracy Fund

E. John Sebes, OSET Foundation

Sarah Whitt, Wisconsin Elections Commission

This publication is available free of charge from:

<http://dx.doi.org/10.6028/NIST.SP.1500-102>

Nov 2017



U. S. Department of Commerce
Wilbur Ross, Secretary

National Institute of Standards and Technology
Walter Copan, Director

National Institute of Standards and Technology (NIST) Special Publication 1500-102
122 pages (November 2017)

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

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

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

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

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

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 document reports on ITL's research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

This document is a specification for a common data format for voter records data interchange related to voter registration, i.e., National Voter Registration Act (NVRA) or Federal Post Card Application (FPCA) style registration requests from online voter registration portals made to voter registration (VR) systems, and responses to the requests returning from the VR system. The specification includes XML and JSON schemas.

Keywords

Common data format (CDF); FPCA; JSON; NVRA; voter registration; VVSG; XML.

Acknowledgements

The editor wishes to thank his colleagues of the National Institute of Standards and Technology Voter Records Interchange Public Working Group, who contributed to its technical content. The editor gratefully acknowledges and appreciates the following contributors for their keen and insightful assistance with developing this specification:

Kim Brace <i>Election Data Services, Inc.</i>	Thomas Connolly <i>NY State Board of Elections</i>	Monica Crane-Childers <i>Democracy Works</i>
Ericka Haas <i>ERIC for the States</i>	Jared Marcotte <i>The Turnout</i>	Lauren Massa-Lochridge
John McCarthy <i>Verified Voting</i>	Janet Modrow <i>Florida Division of Elections</i>	Katy Owens-Hubler <i>Democracy Fund</i>
Tammy Patrick <i>Democracy Fund</i>	Jen Tolentino <i>Rock the Vote</i>	David Wallick <i>Everyone Counts</i>
Stafford Ward <i>U.S. Vote Foundation</i>		

In addition to the above acknowledgments, the editor also gratefully acknowledges and appreciate the significant contributions from individuals and organizations involved in the NIST Voting Interoperability Public Working Group as well as in the public and private sectors, whose thoughtful and constructive comments improved the overall quality, thoroughness, and usefulness of this publication.

Executive Summary

This document presents a specification for voter records data interchange related to voter registration, i.e., registration requests from online voter registration (OVR) made to voter registration (VR) systems, and responses to the requests returning from the VR system. The specification includes XML (eXtensible Markup Language) [1] and JSON (JavaScript Object Notation) [2] schemas.

The purpose of this specification is to provide data interchange formats in XML and JSON for voter registration requests and responses when using the NVRA (National Voter Registration Act) [3] or FPCA (Federal Post Card Application) [4] forms or other U.S. state-specific forms. Advantages of using this specification include:

- A ready data interchange format for online voter registration systems to remove the need for individual OVR system development projects to define data models and formats.
- Applications for OVR systems currently being developed with early adoption and feedback for the CDF VRI to refine development of the CDF for additional use cases.
- A data model in UML (Unified Modeling Language) [5] that itemizes and defines the data involved in voter records interchange related to registration and that is easily extensible to additional use cases. The XML and JSON schemas associated with this specification are generated from this model.

The specification implements use cases for digital VR applications forms transmitted within state OVR systems (e.g., the DMV or MVA) or to state OVR systems by third party OVR systems, following the formats of the NVRA and FPCA voter registration application forms, including state-specific additions to these forms. It includes voter records update (change of name, change of address), change of voter status, and absentee ballot request. Lastly, it includes transfer of voter registration data from 3rd-party OVR registrar to state OVR systems.

Rather than implement new and complicated functionality in this specification for representing addresses, it was decided to use the existing U.S. Thoroughfare, Landmark, and Postal Address Data Standard [6] which contains four major classes of addresses that are broken out into 11 different types of addresses.

This specification is geared towards the following audiences:

- Election officials;
- Voting equipment manufacturers;
- Voter registration database providers;
- Voter registration assistance organizations;
- Election-affiliated organizations and the public.

Table of Contents

Executive Summary	4
1 Introduction	8
1.1 Purpose.....	8
1.2 Audience	8
1.3 Motivation and Methodology	8
2 Overview of Digital OV Transactions as Implemented by this Specification ...	11
2.1 Voter Records Request.....	12
2.2 Voter Records Response.....	14
2.3 State-specific Request and Responses	16
2.4 The U.S. Thoroughfare, Landmark, and Postal Address Data Standard.....	17
2.4.1 Thoroughfare Classes	19
2.4.2 Landmark Classes.....	19
2.4.3 Postal Delivery Classes.....	20
2.4.4 General Class.....	20
3 XML Schema Documentation.....	21
3.1 Schema Stylistic Conventions.....	21
3.2 Roots.....	21
3.3 Imports	22
3.4 Interfaces	22
3.5 Enumerations.....	23
3.5.1 The AssertionValue Enumeration.....	23
3.5.2 The BallotReceiptMethod Enumeration.....	24
3.5.3 The ContactMethodType Enumeration	25
3.5.4 The IdentifierType Enumeration	26
3.5.5 The PhoneCapability Enumeration.....	27
3.5.6 The RegistrationError Enumeration.....	28
3.5.7 The RegistrationForm Enumeration	29
3.5.8 The RegistrationHelperType Enumeration	30
3.5.9 The RegistrationMethod Enumeration.....	31
3.5.10 The RegistrationProxy Enumeration	33
3.5.11 The RegistrationRequestType Enumeration	34

3.5.12 The ReportingUnitType Enumeration	35
3.5.13 The SignatureSource Enumeration	37
3.5.14 The SignatureType Enumeration	38
3.5.15 The SuccessAction Enumeration	39
3.5.16 The VoterClassificationType Enumeration	40
3.5.17 The VoterIdType Enumeration	42
3.6 Interfaces	43
3.7 Classes (Elements).....	44
3.7.1 The <AdditionalInfo> Element	44
3.7.2 The <ContactMethod> Element/Extension Base	45
3.7.3 The <ElectionAdministration> Element	48
3.7.4 The <ExternalIdentifier> Element.....	49
3.7.5 The <File> Element/Extension Base	50
3.7.6 The <LatLng> Element.....	52
3.7.7 The <Location> Element	53
3.7.8 The <Name> (<PreviousName>) Element	54
3.7.9 The <Party> Element	55
3.7.10 The <RegistrationHelper> Element	56
3.7.11 The <RegistrationProxy> Element	57
3.7.12 The ReportingUnit Element	59
3.7.13 The <Signature> (<PreviousSignature>) Element	60
3.7.14 The <VoterClassification> Element	61
3.7.15 The <VoterId> Element	63
3.7.16 The <VoterRecordsRequest> Element	65
3.7.17 The <VoterRecordsResponse> Element/Extension Base.....	67
3.7.18 The <VoterRegistration> Element	71
4 XML and JSON Usage Examples	75
4.1 Example 1: NVRA-style Voter Registration Request in XML	75
4.2 Example 2: NVRA Voter Registration Request in JSON	79
4.3 Example 3: NVRA Voter Records Response in XML.....	83

List of Appendices

Appendix A— Acronyms	85
Appendix B— Glossary	86
Appendix C— References	88
Appendix D— File Download Locations.....	89
Appendix E— XML Schema.....	90
Appendix F— JAVA Schema.....	100

List of Figures

Figure 1 - National Voter Registration Act (NVRA) form	11
Figure 2 - Federal Post Card Application (FPCA) form	12
Figure 3 - Voter Records Request UML class diagram.....	13
Figure 4 - Voter Records Response UML class diagram	15
Figure 5 - FGDC address types mapped to a single Address type	18
Figure 6 - Example NVRA form for an address update request.....	75
Figure 7 - Example NVRA form for an address update request.....	79
Figure 8 - Populated NVRA voter registration response form	83

1 Introduction

This document is a specification for a common data format (CDF) for voter records data interchange related to voter registration, i.e., registration requests from online voter registration (OVR) made to voter registration (VR) systems, and responses to the requests returning from the VR system. The specification includes XML (eXtensible Markup Language) [1] and JSON (JavaScript Object Notation) [2] schemas.

Primary features of this specification include:

- Capability to specify voter registration requests and responses when using the NVRA (National Voter Registration Act) [3] or FPCA (Federal Post Card Application) [4] forms or other U.S. state-specific forms.
- A data model in UML (Unified Modeling Language) [5] that itemizes and defines the data involved in voter records interchange related to registration and that is used to derive the XML and JSON schemas.
- Detailed instructions for implementation and use of the XML and JSON schemas.
- A flexible specification to cover additional use cases for other types of registration transactions, such as for voter record maintenance.

1.1 Purpose

The purpose of this specification is to provide data interchange formats in XML and JSON for voter registration requests and responses so as to assist election officials and developers in implementing and supporting the development of OVR systems within States. Advantages of using this specification include:

- A ready data interchange format for online voter registration systems to remove the need for individual OVR system development projects to define data models and formats.
- Applications for OVR systems currently being developed with early adoption and feedback for the CDF VRI to refine development of the CDF for additional use cases.
- A UML model that is easily extensible to additional use cases.

1.2 Audience

The intended audience of this specification includes election officials, VR system designers and developers, as well as others in the election community including the general public. Some background in election administration and registration is useful in understanding the material in this specification.

1.3 Motivation and Methodology

This document was motivated primarily to assist election officials and developers in implementing and supporting OVR systems by reducing or eliminating non-standard exchange

formats for voter registration data. The current varying systems 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 automate voter registration and related parts of voter record management.

NIST and a community of U.S. election officials, analysts, and election system technologists analyzed varying VR scenarios and use cases and their associated data interchanges, to analyze existing practices and to create a standard data interchange format for emerging OVR systems. This specification implements the following use cases:

1. OVR Submission: Digital VR applications forms transmitted within state OVR systems or to state OVR systems by third party OVR systems, following the formats of the NVRA and FPCA voter registration application forms, including state-specific additions to these forms.
2. VR Update Submission: Similar application forms including: voter registration update (change of name, change of address), change of voter status, and absentee ballot request.
3. OVR Transfer: Subsets of such applications used for 3rd-party OVR registrars to transfer users and user data to state OVR systems.

A UML data model was subsequently generated to represent the data associated with OVR requests and responses and to show how the data elements are related and organized. Finally, XML and JSON schemas were generated from the UML data model.

The advantages of using a UML data model as an intermediate step to generating the XML and JSON schemas include that the model is independent of the concrete data formats (or other potential formats that could be derived), and relationships between data elements are easier to correctly define and visualize when they are independent of any specific data format. If changes are needed to the specific XML and JSON formats, one can make changes to the UML model and then generate or derive new versions of the formats.

Much of the data involved in voter registration requests consists of voter addresses: current registration addresses, postal addresses, previous registration addresses, etc. Rather than implement new and complicated functionality in this specification for representing addresses, it was decided to use the existing U.S. Thoroughfare, Landmark, and Postal Address Data Standard [6] which contains four major classes of addresses that are broken out into 11 different types of addresses. These can be used to represent addresses where voters live and where they receive postal mail (if different), including overseas addresses.

While this specification is focused on digital OVR submission, subsequent versions of this specification may implement additional use cases, including:

- DMV Match: Subsets of such digital applications exchanged between state VR systems and DMV or similar systems, to perform driver's license data matching (if required) as part of OVR processing.

- DMV Notification: Data exchanged by DMV (Department of Motor Vehicles) or similar systems and VR systems, as part of NVRA compliance to digitally notify VR systems of DMV records of DMV customers that requested voter registration. May also include: similar data push from DMV of existing DMV records recently updated with change-of-address, as part of semi-automated steps toward permanent voter registration, or other forms of data exchange to VR systems that might facilitate elements of automatic and/or permanent voter registration.
- Cross-State Records Match: Data interchange between state VR systems for and systems for records matching, e.g., the ERIC (Electronic Registration Information Center) [7] system, or as part of inter-state cross-check activities.
- EAVS Submission: Subsets of voter records externalized from voter records systems for purposes of data aggregation and reporting, including but not limited to EAVS (Election Administration Voting Survey) [8] reporting.

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

The use case for the digital OVR request includes:

- A client IT “OVR submitter”,
- A service IT system “VR authority”, and
- The submission a digital NVRA, FPCA, or state-specific “form” via transmission from OVR submitter to VR authority.

Currently, VR authorities are typically the back-end components of actual state OVR systems, while submitters include clients of state OVR systems such as the DMV/MVA or other clients that are operated by 3rd party VR organizations and that integrate with OVR systems by sending all or part of an NVRA/FPCA dataset collected by the client from a human registrant.

An OVR transaction, as used in this specification, generally consists of a voter records request followed by a voter records response, although this may vary across U.S. states and territories. The request can consist of multiple requests if multiple actions are being requested for the same voter. As well, the response could consist of multiple responses.

The UML model in Figure 3 shows three types of requests that would be sent from an OVR submitter to a VR system:

1. Request a registration for a voter using the digital NVRA or FPCA form.
2. In the case of a registration request using the FPCA form, also request a ballot for an election (FPCA registrations can include a ballot request).
3. Request a state-specific action for a voter using possibly a state-specific digital form.

An OVR submitter essentially specifies the type of registration form (“nvra” or “f pca”) and the registration action(s) being requested for the transaction. In the case of “f pca” when a ballot is also requested, there could be two actions, “registration” and “ballot-request”. The VoterRegistration class contains the remaining data elements, all taken from information contained on the digital form.

Both the registration request and response models contain a class ExternalIdentifier, which is used to associate an identifier with an item. In the case of the request model, it is used optionally to associate an identifier to the political party, and in the response model, to optionally associate identifiers with political geography such as precincts and districts.

2.2 Voter Records Response

The voter records response simply returns a response to the voter records request. The UML model is simpler than the request model in that a response generally contains little data other than the results of the request, which are:

1. The registration request was acknowledged, but no further status is available.
2. The registration request was rejected, including a reason(s) for the rejection.
3. The registration request succeeded, including the action(s) taken.

The UML model in Figure 4 shows the `VoterRecordsResponse` class, which has three corresponding types of responses. The first, `RegistrationAcknowledgement`, represents the typical current practice in online voter registration, where a registration authority operates an online service that receives VR requests and saves them for later processing by local elections offices' staff. In this typical practice, the acceptance of an online VR request is simply an acknowledgement that the request was received without error.

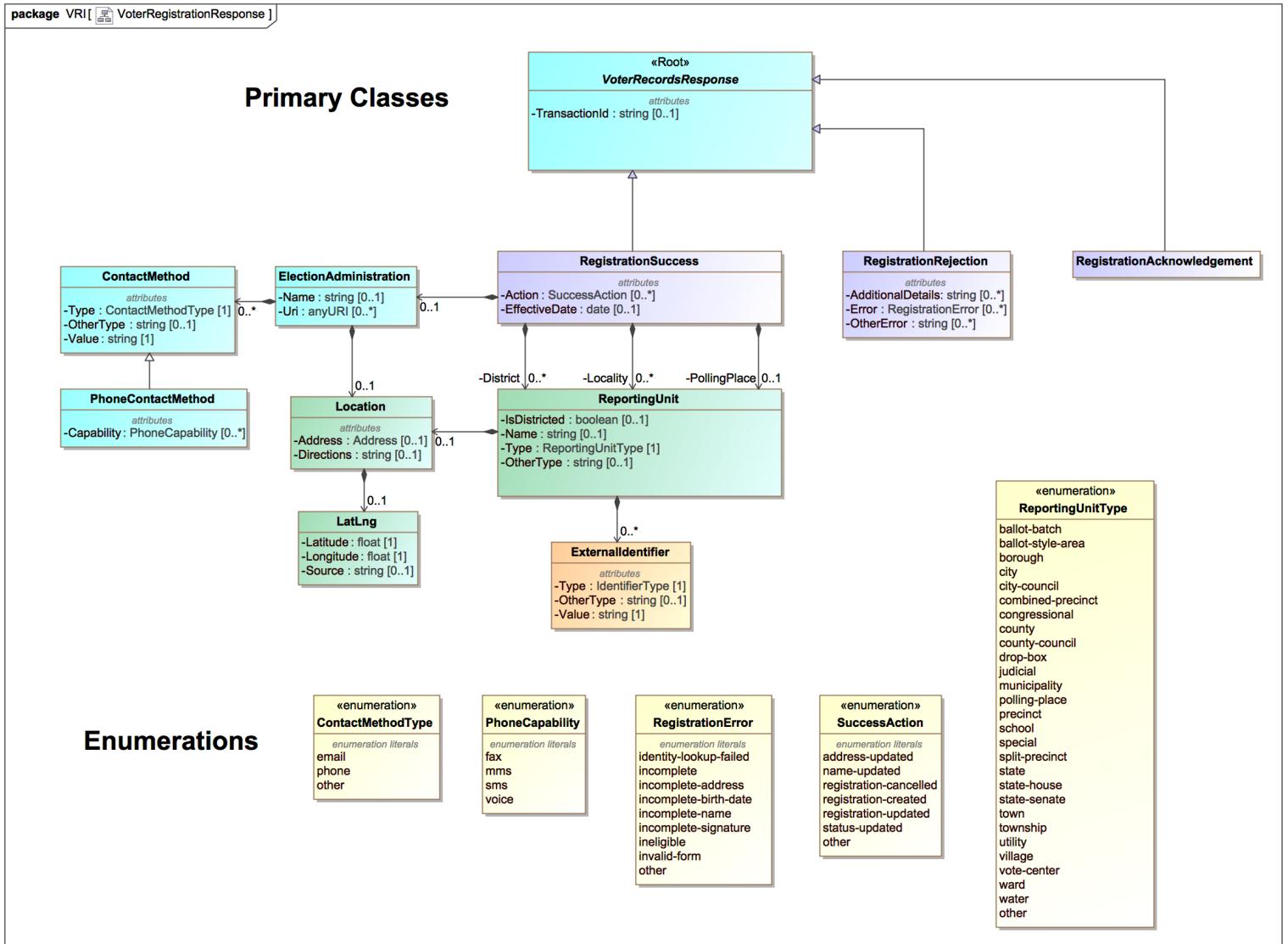


Figure 4 - Voter Records Response UML class diagram

The second type of response, `RegistrationRejection`, contains the reason(s) for the rejection as well as any additional details.

The third type of response, RegistrationSuccess, is used for cases where the receiving service is able to process a request to success and notify the sender synchronously. The contents of a RegistrationSuccess are modeled on the information that becomes available to a newly registered voter as a result of successful registration. In the UML model, the RegistrationSuccess class optionally includes these items: voter's assigned polling place and precinct, the location of the local election authority, and a list of districts that the voter resides in, and if available a list of ballot items in an upcoming election that the voter is eligible to vote in. The successful registration also returns the registration action(s) that occurred, which may differ from what was requested. For example, a request for a new voter registration may succeed, but if the voter was already registered, the response may indicate a registration update as opposed to a registration create.

2.3 State-specific Request and Responses

The UML model and associated XML and JSON schemas contain several features to enable state-specific requests and responses. Using these features, it is possible to use this specification when local modifications are needed.

The AdditionalInfo class is used for information not addressed in this schema by other attributes, e.g., state-specific data that does not "fit" in any other attribute. The type of data will thus be highly specific to the generating application, and consuming applications must "know" the meaning of the data to make use of it. Each AdditionalInfo class would contain the name of the data, and then its value, depending on whether the data represents the value directly as a string or represents a file name.

For example, if a state requires its registration form to include the voter's language, it would need to use the AdditionalInfo class to contain a value representing the voter's language. An XML example is as follows, using English as the voter's language:

```
<AdditionalInfo>
  <Name>Language</Name>
  <StringValue>en-US</StringValue>
</AdditionalInfo>
```

The XML and JSON usage examples in Section 4 contain a number of examples showing usage of the AdditionalInfo class.

Additionally, each enumeration generally contains an "other" value that can be used when none of the enumeration values are sufficient. If "other" is used as the enumeration value, there is an attribute named OtherType that can be used to hold the other data. For example, a state may wish to implement a specific "address-update" transaction, and would thus need to use "other" for the Type attribute in the VoterRecordsRequest class. The OtherType attribute would then contain the type of registration request, i.e., "address-update", e.g. in XML,

```
<VoterRecordsRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema" ... >
  <GeneratedDate>2017-02-10</GeneratedDate>
  <Issuer>Ohio Secretary of State</Issuer>
  <TransactionId>2a642eb5-169e-4a3b-8899-adc7ea6d00d0</TransactionId>
```

```

<Type>other</Type>
<OtherType>address-update</OtherType>
...

```

2.4 The U.S. Thoroughfare, Landmark, and Postal Address Data Standard

Perhaps the most complex part of a voter registration request or other related data exchange of voter record data is the voter's address. There are multiple types of addresses for VR purposes, e.g., current registration address, previous registration address, postal mailing address, overseas address, and multiple types of addresses for location and mailing purposes, e.g., structured street address, unstructured street address, rural addresses, PO box addresses, military and diplomatic addresses, and mailing addresses outside the U.S. Rather than revisit the complexities of address structure, this specification makes use of an existing XML-based standard for structuring addresses: the U.S. Thoroughfare, Landmark, and Postal Address Data Standard [6] issued by the Federal Geographic Data Committee (FGDC) [9] and covering the complexity of addresses managed by or encountered by organizations and agencies such as the United States Census and USPS (U.S. Postal Service). Use of the FGDC standard greatly simplifies this specification and leaves maintenance of the standard to the more appropriate management body.

Briefly, the FGDC standard classifies all US addresses into a simple, complete taxonomy of address classes organized into four groups consisting of 11 address types, with the fourth class being useful for unstructured and non-U.S. addresses:

1. Thoroughfare Classes - 5 address types
2. Landmark Classes - 2 address types
3. Postal Deliver Classes - 3 address types
4. General Class - 3 address types

The VoterRegistration class in the UML request model has four distinct addresses: RegistrationAddress, PreviousRegistrationAddress, MailingAddress, and MailForwardingAddress, all of type Address. Address is itself mapped to 11 address types taken from the FGDC standard, as shown in Figure 5.

Accordingly, the XML and JSON schemas generated from the UML model include the FGDC XML schema and map the use of the `<Address>` type to one of the 11 different address types in the FGDC schema, as shown below using XML:

```

<!-- === Interface Address === -->
<xsd:group name="Address">
  <xsd:choice>
    <xsd:element name="CommunityAddress_type" type="addr:CommunityAddress_type"/>
    <xsd:element name="FourNumberAddressRange_type"
      type="addr:FourNumberAddressRange_type"/>
    <xsd:element name="GeneralAddressClass_type"
      type="addr:GeneralAddressClass_type"/>
    <xsd:element name="IntersectionAddress_type"
      type="addr:IntersectionAddress_type"/>
    <xsd:element name="LandmarkAddress_type" type="addr:LandmarkAddress_type"/>
    <xsd:element name="NumberedThoroughfareAddress_type"
      type="addr:NumberedThoroughfareAddress_type"/>
  </xsd:choice>
</xsd:group>

```

```

    type="addr:NumberedThoroughfareAddress_type"/>
<xsd:element name="TwoNumberAddressRange_type"
    type="addr:TwoNumberAddressRange_type"/>
<xsd:element name="USPSGeneralDeliveryOffice_type"
    type="addr:USPSGeneralDeliveryOffice_type"/>
<xsd:element name="USPSPostalDeliveryBox_type"
    type="addr:USPSPostalDeliveryBox_type"/>
<xsd:element name="USPSPostalDeliveryRoute_type"
    type="addr:USPSPostalDeliveryRoute_type"/>
<xsd:element name="UnnumberedThoroughfareAddress_type"
    type="addr:UnnumberedThoroughfareAddress_type"/>
</xsd:choice>
</xsd:group>
```

The following sections contain brief overviews of each of the address classes and their types.

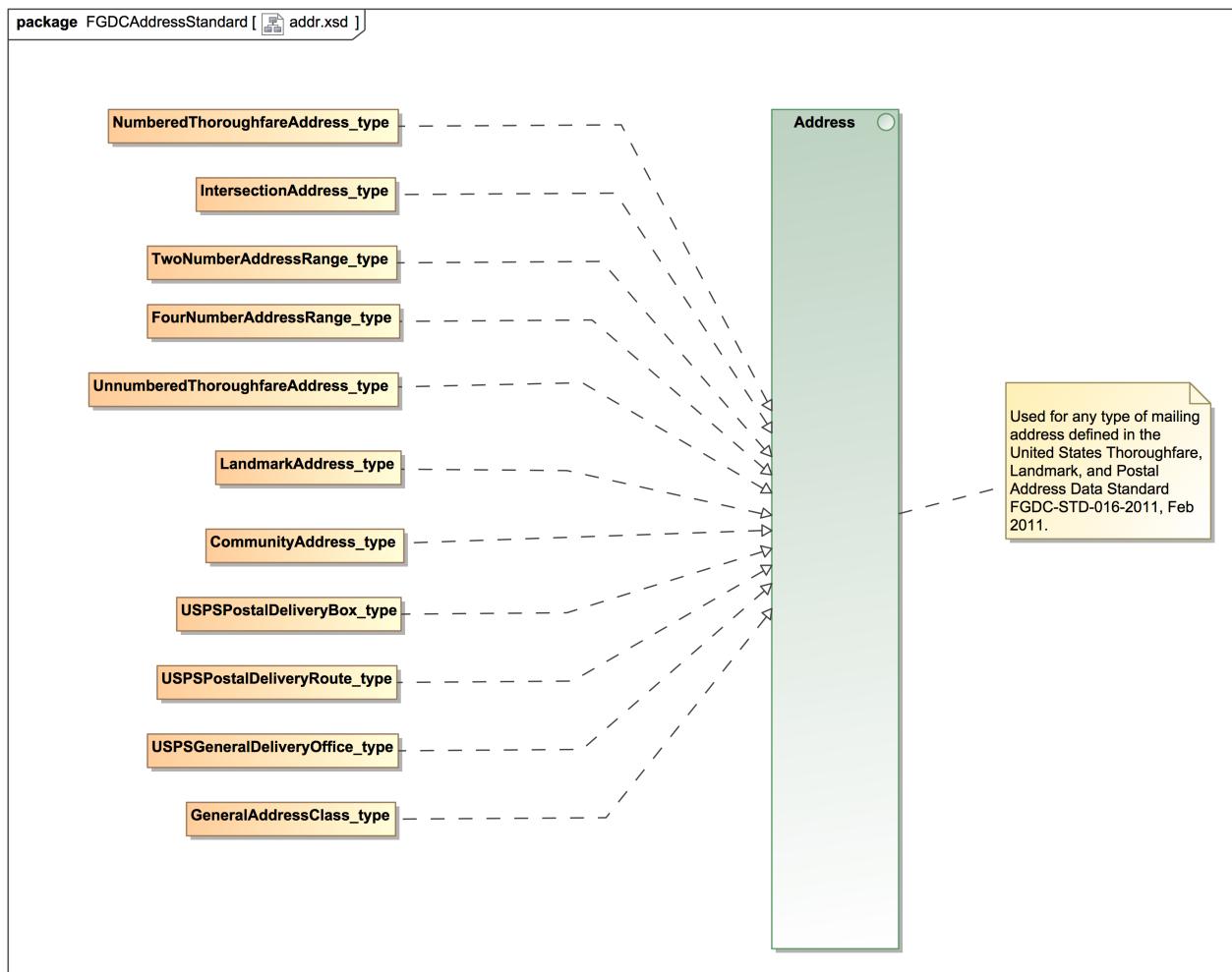


Figure 5 - FGDC address types mapped to a single Address type

2.4.1 Thoroughfare Classes

Most business and residential addresses are Thoroughfare Addresses, therefore this class will be used most often. It specifies a location by reference to a thoroughfare, i.e., a road or part of a road or other access route along which a delivery point can be accessed. A thoroughfare is typically but not always a road - it may be, for example, a walkway, a railroad, or a river. The thoroughfare address types are:

- Numbered Thoroughfare Address, e.g., *123 Main Street*.
- Intersection Address, e.g., *Fifth Avenue and Main Street*.
- Two Number Address Range, e.g., *405-411 West Green Street*.
- Four Number Address Range, e.g., *900-962, 901-963 Milton Street*.
- Unnumbered Thoroughfare Address, e.g., *Forest Service Road 698*.

Of the 5 types list, *Thoroughfare* Addresses are generally used most often. Unnumbered Thoroughfare Addresses are also used for those areas where no address numbers have been assigned and the addresses often include only the thoroughfare name.

The following XML example shows a use of the `<NumberedThoroughfareAddress_type>` for the `<MailingAddress>` element:

```
<MailingAddress>
  <NumberedThoroughfareAddress_type>
    <addr:CompleteAddressNumber>
      <addr_type:AddressNumber>500</AddressNumber>
    </addr:CompleteAddressNumber>
    <addr:CompleteStreetName>
      <addr_type:StreetNamePreDirectional>W</StreetNamePreDirectional>
      <addr_type:StreetName>TUSCARAWAS</StreetName>
      <addr_type:StreetNamePostType>AVE</StreetNamePostType>
    </addr:CompleteStreetName>
    <addr_type:CompletePlaceName>
      <PlaceName PlaceNameType="MunicipalJurisdiction">BARBERTON</PlaceName>
    </CompletePlaceName>
    <addr_type:StateName>OH</StateName>
    <addr_type:ZipCode>44203</ZipCode>
  </NumberedThoroughfareAddress_type>
</MailingAddress>
```

2.4.2 Landmark Classes

Landmark addresses specify a location by reference to a named landmark. A landmark is a relatively permanent feature of the manmade landscape that has recognizable identity within a particular cultural context, e.g., a large statue or structure such as an apartment complex. The landmark address types are:

- Landmark Address, e.g., *Statue of Liberty*.
- Community Address, e.g., *123 Urbanizacion Los Olmos*.

2.4.3 Postal Delivery Classes

Postal delivery addresses specify points of postal delivery that have no definite relation to the location of the recipient, such as a post office box, rural route box, overseas military address, or general delivery office. The USPS specifies each class in detail in USPS Publication 28[10]. The postal delivery types are:

- USPS Postal Delivery Box, e.g., *PO Box 16953*.
- USPS Postal Delivery Route, e.g., *RR 1, Box 100*.
- USPS General Delivery Office, e.g., *General Delivery*.

2.4.4 General Class

The general class provides a "catch-all" way to handle addresses that do not conform to any of the thoroughfare, landmark, or postal classes, including non-U.S. addresses. There are three types:

1. The complete address as a single unparsed string of text, e.g., *PO Box 1511, Ames, IA 50010*.
2. The complete address with place, state and zip code parsed out to a single field, e.g., *PO Box 1511; Place State ZIP = Ames, IA 50010*.
3. The complete address with place, state and zip code parsed out to separate fields, e.g., *PO Box 1511; Complete Place Name = Ames; State Name = IA; Zip Code = 50010*.

A complete overview of the FGDC standard and how to use it is beyond the scope of this specification, thus readers and developers are referred to the standard documentation, which contains complete descriptions and examples [6].

3 XML Schema Documentation

This section contains documentation and discussion of the features included in the VRI XML schema.

In the sections below, an XML element or enumeration name is denoted using italics and angle brackets, e.g., <*ElectionReport*> or <*ReportingUnitType*>. Attributes, enumeration values, or other XML syntax 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., <*VoterRegistration*> is a sub-element of <*VoterRecordsRequest*>. “Includes” is used to denote that an element contains another element as a sub-element, e.g., <*VoterRecordsRequest*> includes <*VoterRegistration*>.

3.1 Schema Stylistic Conventions

The XML schema was written observing the following stylistic conventions:

- Element, attribute, enumeration, and primitive names observe variations of “CamelCase” conventions², that is:
 - Element and enumeration names begin with a capital letter and the names that consist of multiple words are concatenated and each word begins with a capital letter, thus “CamelCase”. For example, <*VoterRegistration*>.
 - Attribute names begin with a non-capital (lower-case) letter and the 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 “camelCase”. For example, *mimeType*.
- Enumeration value names are in non-capital letters, and names that consist of multiple words are separated by hyphens. For example, *1-of-n*.
- Element, attribute, and enumeration value ordering is alphabetical, with the following exception:
 - If there is an enumeration value of *other*, it comes last in the list of values.

3.2 Roots

The schema contains two root elements:

1. <*VoterRecordsRequest*>, used as a root for registration request messages.
2. <*VoterRecordsResponse*>, used as a root for registration response messages.

² See <https://en.wikipedia.org/wiki/CamelCase>.

Schema Definition:

```
<!-- === Roots === -->
<xsd:element name="VoterRecordsRequest" type="VoterRecordsRequest"/>
<xsd:element name="VoterRecordsResponse" type="VoterRecordsResponse"/>
```

3.3 Imports

The schema (and instance files) imports the Federal Geographic Data Committee (FGDC) address schema [6], which contains 11 types of addresses that are used to specify postal and registration addresses for voters, used in the <*VoterRegistration*> element.

Schema Definition:

```
<!-- === Imports === -->
<xsd:import namespace=http://www.fgdc.gov/schemas/address/addr schemaLocation=
"addr.xsd"/>
```

3.4 Interfaces

The schema includes an interface that maps the **Address** type to a choice of one of the address types from the FGDC schema.

```
<!-- === Interface Address === -->
<xsd:group name="Address">
  <xsd:choice>
    <xsd:element name="CommunityAddress_type" type="addr:CommunityAddress_type"/>
    <xsd:element name="FourNumberAddressRange_type"
      type="addr:FourNumberAddressRange_type"/>
    <xsd:element name="GeneralAddressClass_type"
      type="addr:GeneralAddressClass_type"/>
    <xsd:element name="IntersectionAddress_type"
      type="addr:IntersectionAddress_type"/>
    <xsd:element name="LandmarkAddress_type" type="addr:LandmarkAddress_type"/>
    <xsd:element name="NumberedThoroughfareAddress_type"
      type="addr:NumberedThoroughfareAddress_type"/>
    <xsd:element name="TwoNumberAddressRange_type"
      type="addr:TwoNumberAddressRange_type"/>
    <xsd:element name="USPSGeneralDeliveryOffice_type"
      type="addr:USPSGeneralDeliveryOffice_type"/>
    <xsd:element name="USPSPostalDeliveryBox_type"
      type="addr:USPSPostalDeliveryBox_type"/>
    <xsd:element name="USPSPostalDeliveryRoute_type"
      type="addr:USPSPostalDeliveryRoute_type"/>
    <xsd:element name="UnnumberedThoroughfareAddress_type"
      type="addr:UnnumberedThoroughfareAddress_type"/>
  </xsd:choice>
</xsd:group>
```

3.5 Enumerations

The following sections deal with the enumerations (i.e., simple types) in the schema, which are generated from the enumerations in the UML models.

3.5.1 The AssertionValue Enumeration

Used in *requests*.

Enumeration for assertions from a voter or a third party as a department of motor vehicles (DMV) in response to questions on a registration form, used in the `<Assertion>` sub-element of `<VoterClassification>`.

Table 3.1 – Values for <AssertionValue>

Value	Value Description
no	For a voter's assertion of "no" or "false".
yes	For a voter's assertion of "yes" or "true".
unknown	For a voter's assertion of "unknown".

Schema Definition:

```

<xsd:simpleType name="AssertionValue">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="no"/>
    <xsd:enumeration value="yes"/>
    <xsd:enumeration value="unknown"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.2 The BallotReceiptMethod Enumeration

Used in *requests*.

Enumeration for methods for delivering a ballot to the voter, used in the *<BallotReceiptPreference>* sub-element of *<VoterRegistration>*. The sub-element may be repeated multiple times with different values as applicable, e.g., to specify both mail and online.

Table 3.2 – Values for <BallotReceiptMethod>

Value	Value Description
email	For electronic mail.
email-or-online	For electronic mail or downloadable from a website (this value, from the NVRA form, is ambiguous, thus the separate values for email and online).
fax	For use of a fax.
mail	For postal mail.
online	For downloadable from a website, e.g., the voter is sent a hypertext link to a ballot.

Schema Definition:

```

<xsd:simpleType name="BallotReceiptMethod">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="email"/>
    <xsd:enumeration value="email-or-online"/>
    <xsd:enumeration value="fax"/>
    <xsd:enumeration value="mail"/>
    <xsd:enumeration value="online"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.3 The ContactMethodType Enumeration

Used in *request* AND *response* messages.

Enumeration for methods for contacting a voter, used in the *<Type>* sub-element of *<ContactMethod>*.

Table 3.3 – Values for <ContactMethodType>

Value	Value Description
email	For electronic mail.
phone	For use of a phone.
other	Used when the type of contact method is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="ContactMethodType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="email"/>
    <xsd:enumeration value="phone"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.4 The IdentifierType Enumeration

Used in ***request*** messages.

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

Table 3.4 – 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 <i>ocd-id</i> .
ocd-id	For Open Civic Data identifiers [11].
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.

3.5.5 The PhoneCapability Enumeration

Used in *request* AND *response* messages.

Enumeration for telephone capabilities, used in the <Capability> sub-element of <PhoneContactMethod>.

Table 3.5 – Values for <PhoneCapability>

Value	Value Description
fax	For telephones that include facsimile capabilities.
mms	For telephones that contain Multimedia Messaging Service (MMS) capabilities.
sms	For telephones that contain Short Messaging Service (SMS) capabilities.
voice	For telephones that contain voice capabilities.

Schema Definition:

```
<xsd:simpleType name="PhoneCapability">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="fax"/>
    <xsd:enumeration value="mms"/>
    <xsd:enumeration value="sms"/>
    <xsd:enumeration value="voice"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.5.6 The RegistrationError Enumeration

Used in *response* messages.

Enumeration for registration-related errors, used in the *<Error>* sub-element of *<RegistrationRejection>*.

Table 3.6 – Values for <RegistrationError>

Value	Value Description
identity-lookup-failed	A lookup on the voter's identity failed.
incomplete	The registration request is incomplete.
incomplete-address	An address is incomplete.
incomplete-birth-date	The registration request does not contain a birthdate.
incomplete-name	The voter's name is incomplete.
incomplete-signature	The registration request does not contain a signature.
ineligible	The voter is ineligible to be registered.
invalid-form	The registration form specified is invalid.
other	Used when the type of error is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="RegistrationError">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="identity-lookup-failed"/>
    <xsd:enumeration value="incomplete"/>
    <xsd:enumeration value="incomplete-address"/>
    <xsd:enumeration value="incomplete-birth-date"/>
    <xsd:enumeration value="incomplete-name"/>
    <xsd:enumeration value="incomplete-signature"/>
    <xsd:enumeration value="ineligible"/>
    <xsd:enumeration value="invalid-form"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.7 The RegistrationForm Enumeration

Used in ***request*** messages.

Enumeration for types of registration forms, used in the `<RegistrationForm>` sub-element of `<VoterRecordsRequest>`.

Table 3.7 – Values for `<RegistrationForm>`

Value	Value Description
f pca	For the Federal Post Card Application form.
n v r a	For the National Voter Registration Act form.
o t h e r	Used when the type of form is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="RegistrationForm">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="f pca"/>
    <xsd:enumeration value="n v r a"/>
    <xsd:enumeration value="o t h e r"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.8 The RegistrationHelperType Enumeration

Used in ***request*** messages.

Enumeration for types of registration helpers, used in the **<Type>** sub-element of **<RegistrationHelper>**.

Table 3.8 – Values for <RegistrationForm>

Value	Value Description
assistant	For a registration assistant .
witness	For a registration witness .

Schema Definition:

```
<xsd:simpleType name="RegistrationHelperType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="assistant"/>
    <xsd:enumeration value="witness"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.5.9 The RegistrationMethod Enumeration

Used in ***request*** messages.

Enumeration for the method used by the voter to register, used in the **<RegistrationMethod>** sub-element of **<VoterRegistration>**.

Table 3.9 – Values for <RegistrationMethod>

Value	Value Description
armed-forces-recruitment-office	The voter assisted by an armed forces recruitment office.
motor-vehicle-office	The voter via an MVA/DMV.
other-agency-designated-by-state	The voter assisted by an unspecified state-designated agency.
public-assistance-office	The voter assisted by a public assistance office.
registration-drive-from-advocacy-group-or-political-party	The voter via a registration drive.
state-funded-agency-serving-persons-with-disabilities	The voter assisted by a state-designated agency serving persons with disabilities.
voter-via-election-registrars-office	The voter via an election or registrar's office.
voter-via-email	The voter via email.
voter-via-fax	The voter via fax.
voter-via-internet	The voter via the Internet, e.g., a website.
voter-via-mail	The voter via postal mail.
unknown	
other	Used when the type of method is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="RequestSourceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="armed-forces-recruitment-office"/>
    <xsd:enumeration value="motor-vehicle-office"/>
    <xsd:enumeration value="other-agency-designated-by-state"/>
    <xsd:enumeration value="public-assistance-office"/>
    <xsd:enumeration value="registration-drive-from-advocacy-group-or-political-party"/>
    <xsd:enumeration value="state-funded-agency-serving-persons-with-disabilities"/>
    <xsd:enumeration value="voter-via-election-registrars-office"/>
    <xsd:enumeration value="voter-via-email"/>
    <xsd:enumeration value="voter-via-fax"/>
    <xsd:enumeration value="voter-via-internet"/>
  
```

```
<xsd:enumeration value="voter-via-mail"/>
<xsd:enumeration value="unknown"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
```

3.5.10 The RegistrationProxy Enumeration

Used in ***request*** messages.

Enumeration for the [registration proxy](#) used for the voter's registration request, used in the **<Type>** sub-element of **<RegistrationProxy>**.

Table 3.10 – Values for <RegistrationProxy>

Value	Value Description
armed-forces-recruitment-office	The voter assisted by an armed forces recruitment office.
motor-vehicle-office	The voter via an MVA/DMV.
other-agency-designated-by-state	The voter assisted by an unspecified state-designated agency.
public-assistance-office	The voter assisted by a public assistance office.
registration-drive-from-advocacy-group-or-political-party	The voter via a registration drive.
state-funded-agency-serving-persons-with-disabilities	The voter assisted by a state-designated agency serving persons with disabilities.
other	Used when the type of source is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="RegistrationProxyType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="armed-forces-recruitment-office"/>
    <xsd:enumeration value="motor-vehicle-office"/>
    <xsd:enumeration value="other-agency-designated-by-state"/>
    <xsd:enumeration value="public-assistance-office"/>
    <xsd:enumeration value="registration-drive-from-advocacy-group-or-political-
party"/>
    <xsd:enumeration value="state-funded-agency-serving-persons-with-
disabilities"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>

```

3.5.11 The RegistrationRequestType Enumeration

Used in *request* messages.

Enumeration for the type of voter records request, used in the *<Type>* sub-element of *<VoterRecordsRequest>*.

Table 3.11 – Values for <RegistrationRequestType>

Value	Value Description
ballot-request	For requesting a ballot, possibly in conjunction with an FPCA registration request.
registration	For a voter registration request.
other	Used when the type of request is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="RegistrationRequestType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ballot-request"/>
    <xsd:enumeration value="registration"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.12 The ReportingUnitType Enumeration

Used in *request* AND *response* messages.

Enumeration for the type of geopolitical unit, used in the *<Type>* sub-element in the *<ReportingUnit>* element.

Table 3.12 – Values for <ReportingUnitType>

Value	Value Description
ballot-batch	Used for reporting batches of ballots that may cross precinct boundaries.
ballot-style-area	Used for ballot style areas generally composed of precincts
borough	Used in CT, NJ, PA, other states, and New York City for boroughs. For AK and LA, see county.
city	Used for a city that reports results and/or for the district that encompasses it.
city-council	Used for city council districts.
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 for a county and/or for the district that encompasses it. In AK, used for counties that are called boroughs. In LA, used for parishes.
county-council	Used for county council districts.
drop-box	Used for a dropbox for absentee ballots.
judicial	Used for judicial districts.
municipality	Used as applicable for various units such as towns, townships, villages that report votes and/or for the district that encompasses it.
polling-place	Used for a polling place.
precinct	Used also for “Ward” or “District” when these terms are used interchangeably with “Precinct”.
school	Used for a school district.
special	Used for a special district.
split-precinct	Used for splits of precincts.
state	Used for a state and/or for the district that encompasses it.
state-house	Used for a state house or assembly district.
state-senate	Used for a state senate district.
town	Used in some New England states as a type of municipality that reports votes and/or for the district that encompasses it.
township	Used in some mid-western states as a type of municipality that reports votes and/or for the district that encompasses it.

Value	Value Description
utility	Used for a utility district.
village	Used as a type of municipality that reports votes and/or for the district that encompasses it.
vote-center	Used for a vote center.
ward	Used for combinations or groupings of precincts or other units.
water	Used for a water district.
other	Used for other types of reporting units not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="ReportingUnitType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ballot-batch"/>
    <xsd:enumeration value="ballot-style-area"/>
    <xsd:enumeration value="borough"/>
    <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>

```

3.5.13 The SignatureSource Enumeration

Used in *request* messages.

Enumeration for source of the voter's signature, used in the <Source> sub-element of <Signature>.

Table 3.13 – Values for <SignatureSource>

Value	Value Description
dmv	For the department of motor vehicles, motor vehicle authority.
local	For an unspecified local source.
state	For an unspecified state source.
voter	The voter has included a signature on the form.
other	Used when the source of the signature is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="Source">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="dmv"/>
    <xsd:enumeration value="local"/>
    <xsd:enumeration value="state"/>
    <xsd:enumeration value="voter"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.14 The SignatureType Enumeration

Used in ***request*** messages.

Enumeration for the type of voter signature, used in the **<Type>** sub-element of **<Signature>**.

Table 3.14 – Values for <SignatureType>

Value	Value Description
dynamic	For use with biometrics or other artifacts captured as part of the act of the voter signing the registration form.
electronic	For a facsimile of the signature applied to a marking surface, e.g., paper.
other	Used when the type of signature is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="SignatureType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="digital"/>
    <xsd:enumeration value="dynamic"/>
    <xsd:enumeration value="electronic"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.15 The SuccessAction Enumeration

Used in *response* messages.

Enumeration for a response to a voter records request, indicating that the response to the request is successful and the action that occurred, used in the *<Action>* sub-element of *<RegistrationSuccess>*. The success action may not necessarily match the requested action, as noted in section 3.7.17.3).

Table 3.15 – Values for <SuccessAction>

Value	Value Description
address-updated	For indicating that an address was updated.
name-updated	For indicating that a name was updated.
registration-cancelled	For indicating that a registration was cancelled.
registration-created	For indicating that a registration was created.
registration-updated	For indicating that a registration was updated.
status-updated	For indicating that a registration status was updated.
other	Used for other types of success actions not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="SuccessAction">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="address-updated"/>
    <xsd:enumeration value="name-updated"/>
    <xsd:enumeration value="registration-cancelled"/>
    <xsd:enumeration value="registration-created"/>
    <xsd:enumeration value="registration-updated"/>
    <xsd:enumeration value="status-updated"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.5.16 The VoterClassificationType Enumeration

Used in ***request*** messages.

Enumeration for voter status classifications, used in the **<Type>** sub-element of **<VoterClassification>**. Whether the voter status, e.g., *eighteen-on-election-day*, is true, false, or unknown depends on the value of the **<Assertion>** sub-element.

Table 3.16 – Values for <VoterClassificationType>

Value	Value Description
activated-national-guard	The voter is an activated National Guard member on State orders (FPCA).
active-duty	The voter is a member of the Uniformed Services or Merchant Marine on active duty (FPCA).
active-duty-spouse-or-dependent	The voter is an eligible spouse or dependent (FPCA).
citizen-abroad-intent-to-return	The voter is a US citizen residing outside the US and has intention to return (FPCA).
citizen-abroad-never-resided	The voter is a US citizen and has never resided in the US (FPCA).
citizen-abroad-return-uncertain	The voter is a US citizen residing outside the US and their return is uncertain (FPCA).
deceased	The voter is deceased (NVRA).
declared-incompetent	The voter has been declared incompetent (NVRA).
eighteen-on-election-day	The voter will be 18 on election day (NVRA).
felon	The voter is a felon (NVRA).
permanently-denied	The voter has not been permanently denied (NVRA).
protected-voter	The voter status is protected (NVRA).
restored-felon	The voter is a restored felon (NVRA).
united-states-citizen	The voter is a United States citizen (NVRA).
other	Used when the type of voter classification is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="VoterClassificationType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="activated-national-guard"/>
    <xsd:enumeration value="active-duty"/>
  
```

```
<xsd:enumeration value="active-duty-spouse-or-dependent"/>
<xsd:enumeration value="citizen-abroad-intent-to-return"/>
<xsd:enumeration value="citizen-abroad-return-uncertain"/>
<xsd:enumeration value="citizen-abroad-never-resided"/>
<xsd:enumeration value="deceased"/>
<xsd:enumeration value="declared-incompetent"/>
<xsd:enumeration value="eighteen-on-election-day"/>
<xsd:enumeration value="felon"/>
<xsd:enumeration value="permanently-denied"/>
<xsd:enumeration value="protected-voter"/>
<xsd:enumeration value="restored-felon"/>
<xsd:enumeration value="united-states-citizen"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
```

3.5.17 The VoterIdType Enumeration

Used in *request* messages.

Enumeration for the type of voter ID, used in the *<Type>* sub-element of *<VoterId>*.

Table 3.17 – Values for <VoterIdType>

Value	Value Description
drivers-license	Used for a driver's license.
local-voter-registration-id	Used for a local voter registration ID.
ssn	Used for a complete Social Security number.
ssn4	Used for the last four digits of a Social Security number.
state-id	Used for a state ID that is not a state voter registration ID.
state-voter-registration-id	Used for a state's voter registration ID.
unspecified-document	Used for an unspecified document, not known whether the document contains name, address, or photo ID.
unspecified-document-with-name-and-address	Used for a document that contains the voter's name and address, such as a utility bill.
unspecified-document-with-photo-identification	Used for a document that contains a photograph of the voter.
unknown	
other	Used when the type of ID is not included in this enumeration.

Schema Definition:

```

<xsd:simpleType name="VoterIdType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="drivers-license"/>
    <xsd:enumeration value="local-voter-registration-id"/>
    <xsd:enumeration value="unspecified-document-with-name-and-address"/>
    <xsd:enumeration value="unspecified-document"/>
    <xsd:enumeration value="unspecified-document-with-photo-identification"/>
    <xsd:enumeration value="ssn"/>
    <xsd:enumeration value="ssn4"/>
    <xsd:enumeration value="state-id"/>
    <xsd:enumeration value="state-voter-registration-id"/>
    <xsd:enumeration value="unknown"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>

```

3.6 Interfaces

The UML model includes an interface to the FGDC address schema, which permits any one of the 11 address subtypes to be used in any of the address elements that are of type <Address>.

Schema Definition:

```
<xsd:group name="Address">
  <xsd:choice>
    <xsd:element name="CommunityAddress_type"
      type="addr:CommunityAddress_type"/>
    <xsd:element name="FourNumberAddressRange_type"
      type="addr:FourNumberAddressRange_type"/>
    <xsd:element name="GeneralAddressClass_type"
      type="addr:GeneralAddressClass_type"/>
    <xsd:element name="IntersectionAddress_type"
      type="addr:IntersectionAddress_type"/>
    <xsd:element name="LandmarkAddress_type"
      type="addr:LandmarkAddress_type"/>
    <xsd:element name="NumberedThoroughfareAddress_type"
      type="addr:NumberedThoroughfareAddress_type"/>
    <xsd:element name="TwoNumberAddressRange_type"
      type="addr:TwoNumberAddressRange_type"/>
    <xsd:element name="USPSGeneralDeliveryOffice_type"
      type="addr:USPSGeneralDeliveryOffice_type"/>
    <xsd:element name="USPSPostalDeliveryBox_type"
      type="addr:USPSPostalDeliveryBox_type"/>
    <xsd:element name="USPSPostalDeliveryRoute_type"
      type="addr:USPSPostalDeliveryRoute_type"/>
    <xsd:element name="UnnumberedThoroughfareAddress_type"
      type="addr:UnnumberedThoroughfareAddress_type"/>
  </xsd:choice>
</xsd:group>
```

3.7 Classes (Elements)

The following sections deal with the elements (i.e., complex types) in the schema, which are generated from the UML model classes.

3.7.1 The <AdditionalInfo> Element

Used in *request* messages.

<*VoterRegistration*> optionally includes this element for specifying information not addressed in this schema by other elements and attributes, e.g., state-specific information that does not “fit” in any other element. The information will thus be highly specific to the generating application, and consuming applications must “know” the meaning of the information to make use of it. For this reason, use of this element is discouraged as much as is possible.

The <*StringValue*> and <*FileValue*> sub-elements are both optional, however at least one of them must be included.

Table 3.18 – Elements for <AdditionalInfo>

Element	Multiplicity	Type	Element Description
< <i>FileValue</i> >	0 or 1	file	Used if the value is in a file; contains the filename and MIME type
< <i>Name</i> >	1	xsd:string	Name of the value.
< <i>StringValue</i> >	0 or 1	xsd:string	Used if the value is a string; contains the string.

XML example:

```
<AdditionalInfo>
  <Name>Voter Preferred Language</Name>
  <StringValue>en_US</StringValue>
</AdditionalInfo>
```

Schema definition:

```
<xsd:complexType name="AdditionalInfo">
  <xsd:sequence>
    <xsd:element name="FileValue" type="File" minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="StringValue" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.2 The <ContactMethod> Element/Extension Base

Used in *request* and *response* messages.

<*ElectionAdministration*> optionally includes this element to specify how to contact the election administration.

<*VoterRegistration*> optionally includes this element to specify the method for contacting a voter regarding the voter's registration request. If the voter can be contacted in multiple ways, the application creating the XML instance file should order the occurrences of <*ContactMethod*> by priority.

The <*PhoneContactMethod*> element uses <*ContactMethod*> as an extension base, thus <*ContactMethod*> can be used with *xsi:type="PhoneContactMethod"* when the contact method is for a telephone and it is necessary to describe the capabilities of the telephone.

The <*Capability*> sub-element is provided by the <*PhoneContactMethod*> element.

Table 3.19 – Elements for <ContactMethod>

Element	Multiplicity	Type	Element Description
< <i>OtherType</i> >	0 or 1	<i>xsd:string</i>	Used when < <i>ContactMethodType</i> > value is <i>other</i> .
< <i>Type</i> >	1	<i>ContactMethodType</i>	The contact method type, e.g., <i>email</i> or <i>phone</i> .
< <i>Value</i> >	1	<i>xsd:string</i>	Contains an email address or phone number, etc.

XML example:

```
<!-- This shows that a telephone number can receive voice and text messages
-->
<ContactMethod xsi:type="PhoneContactMethod">
    <Capability>sms</Capability>
    <Capability>voice</Capability>
    <Type>phone</Type>
    <Value>304-123-4567</Value>
</ContactMethod>
```

Schema definition:

```
<xsd:complexType name="ContactMethod">
  <xsd:sequence>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Type" type="ContactMethodType"/>
    <xsd:element name="Value" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.2.1 The PhoneContactMethod xsi:type

Used in *request* and *response* messages.

<RegistrationAssistant>, and *<RegistrationProxy>* use this element to specify a telephone number as well as the capabilities of the telephone, e.g., *sms*, *fax*, etc.

<PhoneContactMethod> is an *xsi:type* of *<ContactMethod>*, i.e., it uses *<ContactMethod>* as an extension base. Thus, the elements that include *<ContactMethod>* could use *xsi:type="PhoneContactMethod"* as applicable.

Table 3.20 – Elements for <PhoneContactMethod>

Element	Multiplicity	Type	Element Description
<i><Capability></i>	0 or more	PhoneCapability	Specifies the phone's capabilities, e.g., <i>fax</i> , <i>sms</i> .

XML example:

```
<ContactMethod xsi:type="PhoneContactMethod">
    <Capability>sms</Capability>
    <Capability>voice</Capability>
    <Type>phone</Type>
    <Value>304-123-4567</Value>
</ContactMethod>
```

Schema definition:

```
<xsd:complexType name="PhoneContactMethod">
    <xsd:complexContent>
        <xsd:extension base="ContactMethod">
            <xsd:sequence>
                <xsd:element name="Capability" type="PhoneCapability" minOccurs="0"
                    maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

3.7.3 The <ElectionAdministration> Element

Used in *response* messages.

<*ElectionAdministration*> optionally includes <*ContactInformation*> to specify contact information for the election authority.

Table 3.21 – Elements for <ElectionAdministration>

Element	Multiplicity	Type	Element Description
< <i>ContactMethod</i> >	0 or more	<i>ContactMethod</i>	For including various contact information.
< <i>Location</i> >	0 or 1	<i>Location</i>	Location of the election authority.
< <i>Name</i> >	0 or 1	xsd:string	Name of the election authority.
< <i>Uri</i> >	0 or more	xsd:anyURI	A URL for the election authority.

XML example:

```
<ElectionAdministration>
  <Name>Ohio County Board of Elections</Name>
  <Uri>http://sos.wv.gov/elections/ohio-county/index.html
</ElectionAdministration>
```

Schema definition:

```
<xsd:complexType name="ElectionAdministration">
  <xsd:sequence>
    <xsd:element name="ContactMethod" type="ContactMethod" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element name="Location" type="Location" minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Uri" type="xsd:anyURI" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.4 The <ExternalIdentifier> Element

Used in *request* AND *response* messages.

<*Party*> and <*ReportingUnit*> optionally include this element for associating a jurisdiction's codes, i.e., identifiers, with political parties or geopolitical units such as counties, towns, precincts, etc. Multiple occurrences of <*ExternalIdentifier*> can be used to associate multiple codes, e.g., if there is a desire to associate multiple codes with an object such as state-specific codes as well as OCD-IDs (Open Civic Data Identifiers [11]).

Table 3.22 – Elements for <ExternalIdentifier>

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

Usage examples:

```

<!-- This shows a state-wide ID and an OCD-ID -->
<ExternalIdentifier>
    <Type>state-level</Type>
    <Value>54</Value>
</ExternalIdentifier>
<ExternalIdentifier>
    <Type>ocd-id</Type>
    <Value>ocd-division/country:us/state:wv</Value>
</ExternalIdentifier>

<!-- If the type of identifier is not listed in enumeration <IdentifierType>,
     use other and include the type in OtherType
-->
<ExternalIdentifier>
    <Type>other</Type>
    <Value>101-A</Value>
    <OtherType>Ohio County Precinct Numbers</OtherType>
</ExternalIdentifier>

```

Schema definition:

```

<xsd:complexType name="ExternalIdentifier">
    <xsd:sequence>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="IdentifierType"/>
        <xsd:element name="Value" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:complexType>

```

3.7.5 The <File> Element/Extension Base

Used in ***request*** messages.

<*VoterId*> optionally uses this type for <*FileValue*> to specify a filename for voter identification purposes such as for a utility bill. <*AdditionalInfo*> also optionally includes <*FileValue*>.

<*File*> extends the *xsd:base64Binary* simple type to add the attributes for filename and (Multi-Purpose Internet Mail Extensions) MIME type, e.g., *application/pdf* for a file of type PDF.

The <*Image*> element uses this element as an extension base, thus <*File*> can be used with *xsi:type="Image"* when the type of file is for an image, e.g., *image/png*.

Table 3.23 – Attributes for <File>

Attribute	Required	Type	Attribute Description
fileName	no	<i>xsd:string</i>	The filename.
mimeType	no	<i>xsd:string</i>	The MIME type associated with the file.

XML example:

```
<!-- This shows including a document of type jpeg --->
<AdditionalInfo>
    <Name>certificate</Name>
    <FileValue FileValue="cer859-X" mimeType=image/jpeg />
</AdditionalInfo>
```

Schema definition:

```
<xsd:complexType name="File">
    <xsd:simpleContent>
        <xsd:extension base="xsd:base64Binary">
            <xsd:attribute name="fileName" type="xsd:string"/>
            <xsd:attribute name="mimeType" type="xsd:string"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

3.7.5.1 The <Image> xsi:type

Used in **request** messages.

<*Signature*> optionally includes this element to indicate that a file contains an image of a voter's signature. <Image> uses <File> as an extension base, thus attributes of <File> can be included in <Image>.

XML example:

```
<File xsi:type="Image" FileValue="0174859-S" mimeType="image/jpeg"/>
```

Schema definition:

```
<xsd:complexType name="Image">
  <xsd:complexContent>
    <xsd:extension base="File"/>
  </xsd:complexContent>
</xsd:complexType>
```

3.7.6 The <LatLng> Element

Used in ***response*** messages.

<*Location*> optionally includes this element to specify the latitude and longitude of a voter's voting location.

Table 3.24 – Elements for <LatLng>

Element	Multiplicity	Type	Element Description
<Latitude>	1	xsd:float	Latitude of the contact location.
<Longitude>	1	xsd:float	Longitude of the contact location.
<Source>	0 or 1	xsd:string	System used to perform the lookup from location name to lat/lng, e.g., the name of a geocoding service.

XML example:

```
<LatLng>
  <Latitude>40.0640</Latitude>
  <Longitude>80.7209</Longitude>
</LatLng>
```

Schema definition:

```
<xsd:complexType name="LatLng">
  <xsd:sequence>
    <xsd:element name="Latitude" type="xsd:float"/>
    <xsd:element name="Longitude" type="xsd:float"/>
    <xsd:element name="Source" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.7 The <Location> Element

Used in ***response*** messages.

<*ReportingUnit*> and <*ElectionAdministration*> optionally include this element to specify the address and directions to a voter's voting location. The <*LatLng*> element can be included to specify the latitude and longitude of the voting location.

Table 3.25 – Elements for <Location>

Element	Multiplicity	Type	Element Description
<Address>	0 or 1	Address	Address of the voting location.
<Directions>	0 or 1	xsd:string	Directions to the voting location.
<LatLng>	0 or 1	xsd:string	Latitude/longitude of the voting location.

Schema definition:

```

<xsd:complexType name="Location">
  <xsd:sequence>
    <xsd:element name="Address" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Directions" type="xsd:string" minOccurs="0"/>
    <xsd:element name="LatLng" type="LatLng" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.8 The <Name> (<PreviousName>) Element

Used in ***request*** messages.

<*VoterRegistration*> includes this element for specifying the name of a voter and, optionally, for specifying a previous name of the voter, using <*PreviousName*> instead of <*Name*>. <*RegistrationHelper*> also includes this element for specifying the name of a registration helper.

Multiple occurrences of the <*MiddleName*> sub-element can be used as needed, e.g., for names with additional middle names or nicknames such as “John Andrew Winston (Jack) Smith”.

Table 3.26 – Elements for <Name>

Element	Multiplicity	Type	Element Description
< <i>FirstName</i> >	0 or 1	xsd:string	Person’s first (given) name.
< <i>FullName</i> >	0 or 1	Internationalized Text	Person’s full name.
< <i>LastName</i> >	0 or 1	xsd:string	Person’s last (family) name.
< <i>MiddleName</i> >	0 or more	xsd:string	Person’s middle name.
< <i>Prefix</i> >	0 or 1	xsd:string	A prefix associated with the person, e.g., Mr.
< <i>Suffix</i> >	0 or 1	xsd:string	A suffix associated with the person, e.g., Jr.

XML example:

```

<Name>
  <FirstName>John</FirstName>
  <MiddleName>Andrew</MiddleName>
  <MiddleName>Winston</MiddleName>
  <MiddleName>(Jack)</MiddleName>
  <LastName>Smith</LastName>
</Name>

```

Schema definition:

```

<xsd:complexType name="Name">
  <xsd:sequence>
    <xsd:element name="FirstName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="FullName" 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="Prefix" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Suffix" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.9 The <Party> Element

Used in ***request*** messages.

<*VoterRegistration*> includes this element to specify a voter's political party.

Table 3.27 – Elements for <Party>

Element	Multiplicity	Type	Element Description
<Abbreviation>	0 or 1	xsd:string	Short name for the party, e.g., "DEM".
<ExternalIdentifier>	0 or 1	External Identifier	For associating an ID with the party.
<Name>	1	xsd:string	Official full name of the party, e.g., "Republican".

Schema Definition:

```

<xsd:complexType name="Party">
  <xsd:sequence>
    <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ExternalIdentifier" type="ExternalIdentifier"
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Name" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.10 The <RegistrationHelper> Element

Used in ***request*** messages.

<*VoterRegistration*> optionally includes this element to specify information about a registration helper, i.e., a [registration assistant](#) or [registration witness](#) involved in a voter's registration request.

<*RegistrationAssistant*> includes the <*Name*> element to specify the registration helper's name and optionally includes the <*Signature*> element if a registration helper's signature is required.

Table 3.28 – Elements for <RegistrationHelper>

Element	Multiplicity	Type	Element Description
< <i>Address</i> >	0 or 1	Address	Address of the registration helper.
< <i>Name</i> >	1	Name	To specify the name of the helper.
< <i>Phone</i> >	0 or 1	PhoneContactMethod	Registration helper's phone number.
< <i>Signature</i> >	0 or 1	Signature	To specify the signature of the helper.
< <i>Type</i> >	1	RegistrationHelper Type	To specify the type of helper, e.g., <i>assistant</i> .

Schema definition:

```

<xsd:complexType name="RegistrationHelper">
  <xsd:sequence>
    <xsd:element name="Address" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Name" type="Name" minOccurs="0"/>
    <xsd:element name="Phone" type="PhoneContactMethod" minOccurs="0"/>
    <xsd:element name="Signature" type="Signature" minOccurs="0"/>
    <xsd:element name="Type" type="RegistrationHelperType"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.11 The <RegistrationProxy> Element

Used in ***request*** messages.

<*VoterRegistration*> optionally includes this element to specify information about a [registration proxy](#) involved in a voter records request.

<*OriginTransactionId*> can be used to include an optional identifier of the originating external transaction from the proxy, e.g., used for the transaction ID generated by a DMV application enacting a voter registration request to a registration portal application (on behalf of a citizen obtaining a driver's license). This sub-element is not to be confused with <*TransactionId*> in <*VoterRecordsRequest*>, which is used to include a transaction ID of the voter records request, e.g., the transaction ID of the registration portal's voter records request.

Table 3.29 – Elements for <RegistrationProxy>

Element	Multiplicity	Type	Element Description
<Address>	0 or 1	Address	An address associated with the proxy.
<Name>	0 or 1	xsd:string	A name associated with the proxy.
<Origin TransactionId>	0 or 1	xsd:string	An identifier associated with the transaction between the proxy and, e.g., the registration portal.
<OtherType>	0 or 1	xsd:string	Used when < <i>RegistrationProxyType</i> > value is <i>other</i> .
<Phone>	0 or 1	PhoneContact Method	A phone number associated with the proxy.
<TimeStamp>	0 or 1	xsd:date	The date of the request from the proxy.
<Type>	1	Registration ProxyType	The type of the requesting proxy, e.g., <i>motor-vehicle-office</i> , <i>voter-via-email</i> .

Schema definition:

```
<xsd:complexType name="RegistrationProxy">
  <xsd:sequence>
    <xsd:element name="Address" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:group ref="Address" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OriginTransactionId" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Phone" type="PhoneContactMethod" minOccurs="0"/>
    <xsd:element name="TimeStamp" type="xsd:date" minOccurs="0"/>
    <xsd:element name="Type" type="RegistrationProxyType"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.12 The ReportingUnit Element

Used in ***response*** messages.

<*VoterRecordsResponse*> includes this element when a registration request is successful so as to provide a list of geopolitical geography associated with the voter's registration, e.g., the voter's precinct, polling place, districts, etc. The <*Type*> sub-element uses the <*ReportingUnitType*> enumeration to specify the type of geopolitical geography being defined. 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 is not strictly necessary, as it is possible to identify districts by their <*Type*> sub-element. However, if the type of district is not listed in the <*ReportingUnitType*> enumeration and therefore <*OtherType*> is used, then <*IsDistricted*> is necessary. The <*IsDistricted*> boolean can also be used to signify that a <*ReportingUnit*> defined as a jurisdiction, e.g., a county, is also used as a district for, e.g., county-wide contests.

Table 3.30 – Elements for <*ReportingUnit*>

Element	Multiplicity	Type	Element Description
< <i>IsDistricted</i> >	0 or 1	xsd:boolean	Boolean to indicate that the reporting unit is a district.
< <i>Name</i> >	0 or 1	xsd:string	Name of the reporting unit.
< <i>OtherType</i> >	0 or 1	xsd:string	Used when < <i>ReportingUnitType</i> > value is <i>other</i> .
< <i>Type</i> >	1	ReportingUnit Type	Enumerated type of reporting unit, e.g., <i>district</i> , <i>precinct</i> .

Schema definition:

```

<xsd:complexType name="ReportingUnit">
  <xsd:sequence>
    <xsd:element name="ExternalIdentifier" type="ExternalIdentifier"
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="IsDistricted" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="Location" type="Location" minOccurs="0"/>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Type" type="ReportingUnitType"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.13 The <Signature> (<PreviousSignature>) Element

Used in ***request*** messages.

<*VoterRegistration*> includes this element for specifying information about a voter's signature on a registration request. If there is a need to include previous signature that uses a different name, e.g., a maiden name, <*VoterRegistration*> uses <*PreviousSignature*> instead of <*Signature*>.

<*Source*> is used to specify the source of the voter's signature, for example, on file at a department of motor vehicles. <*FileValue*> is used to include an image of the voter's signature.

Table 3.31 – Elements for <Signature>

Element	Multiplicity	Type	Element Description
< <i>Date</i> >	0 or 1	xsd:date	The date of the signature, i.e., when created.
< <i>FileValue</i> >	0 or 1	Image	The signature image in base 64 binary.
< <i>OtherSource</i> >	0 or 1	xsd:string	Used when < <i>Source</i> > value is <i>other</i> .
< <i>OtherType</i> >	0 or 1	xsd:string	Used when < <i>SignatureType</i> > value is <i>other</i> .
< <i>Source</i> >	0 or 1	SignatureSource	A source for the signature, e.g., <i>dmv</i> .
< <i>Type</i> >	0 or 1	SignatureType	A signature type, e.g., <i>dynamic</i> .

Schema definition:

```

<xsd:complexType name="Signature">
  <xsd:sequence>
    <xsd:element name="Date" type="xsd:date" minOccurs="0"/>
    <xsd:element name="FileValue" type="Image" minOccurs="0"/>
    <xsd:element name="OtherSource" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Source" type="SignatureSource" minOccurs="0"/>
    <xsd:element name="Type" type="SignatureType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.14 The <VoterClassification> Element

Used in ***request*** messages.

<*VoterRegistration*> includes this element to describe a voter's classification per criteria on the voter's registration form, e.g., *united-states-citizen* or *eighteen-on-election-day*.

<*VoterClassification*> includes assertions of the voter in response to the voter registration form criteria. For example, an assertion of *true* may be used with a criterion of *united-states-citizen*. Assertions can be negative, such as providing an assertion of *false* for a criterion of *felon*, or an assertion of *unknown* to indicate that the voter does not know whether they meet or do not meet the specific criteria on the form.

Table 3.32 – Elements for <VoterClassification>

Element	Multiplicity	Type	Element Description
<Assertion>	1	AssertionValue	A positive or negative or unknown assertion.
<OtherType>	0 or 1	xsd:string	Used when < <i>VoterClassificationType</i> > value is <i>other</i> .
<Type>	1	VoterClassification Type	A classification type, e.g., <i>disabled</i> .

XML example:

```

<!-- This shows the voter attesting they are 18 on election day,
    a U.S. citizen, and a resident of West Virginia. The state
    attestation is not part of the <VoterClassificationType>
    enumeration, so OtherType is used.
-->
<VoterClassification>
    <Assertion>yes</Assertion>
    <Type>eighteen-on-election-day</Type>
</VoterClassification>
<VoterClassification>
    <Assertion>yes</Assertion>
    <Type>united-states-citizen</Type>
</VoterClassification>
<VoterClassification>
    <Assertion>yes</Assertion>
    <OtherType>west-virginia-resident</OtherType>
    <Type>other</Type>
</VoterClassification>

```

Schema definition:

```

<xsd:complexType name="VoterClassification">
    <xsd:sequence>

```

```
<xsd:element name="Assertion" type="AssertionValue"/>
<xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
<xsd:element name="Type" type="VoterClassificationType"/>
</xsd:sequence>
</xsd:complexType>
```

3.7.15 The <VoterId> Element

Used in ***request*** messages.

Used to include information about a voter's identification that may be required in a registration request. **<VoterRegistration>** includes **<VoterId>**.

<AttestNoSuchId> is used to attest that the voter has no ID of a specified type, thus it must be included with a value of *true* if attesting that the voter has no ID for that specified type. It can be included with a value of *false* to attest that the voter does have an ID of the specified type, in which case either **<StringValue>** or **<FileValue>** must be included; however, it is assumed to be false if not included. The **<StringValue>** and **<FileValue>** sub-elements are both optional, however at least one of them must be included.

Table 3.33 – Elements for <VoterId>

Element	Multiplicity	Type	Element Description
<AttestNoSuchId>	0 or 1	xsd:boolean	Used to attest that the voter has no ID. Assumed to be <i>false</i> if not present.
<DateOfIssuance>	0 or 1	xsd:date	Date the ID was issued.
<FileValue>	0 or 1	File	Used to include a file name for the ID.
<OtherType>	0 or 1	xsd:string	Used when <VoterIdType> value is <i>other</i> .
<StringValue>	0 or 1	xsd:string	Used to include the ID as a string.
<Type>	1	VoterIdType	The type of voter ID

XML example:

```

<VoterId>
    <!-- Attesting that the voter DOES possess a driver's license;
        <AttestNoSuchId> is optional; if included, it must have a
        value of false.
    ---->
    <AttestNoSuchId>false</AttestNoSuchId>
    <StringValue>AB879456</StringValue>
    <Type>drivers-license</Type>
</VoterId>
<VoterId>
    <!-- Attesting that the voter DOES NOT possess a social security
        number; in this case, <AttestNoSuchId> with a value of true
        must be included.
    ---->
    <AttestNoSuchId>true</AttestNoSuchId>
    <Type>ssn4</Type>
</VoterId>

```

Schema definition:

```
<xsd:complexType name="VoterId">
  <xsd:sequence>
    <xsd:element name="AttestNoSuchId" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="DateOfIssuance" type="xsd:date" minOccurs="0"/>
    <xsd:element name="FileValue" type="File" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="StringValue" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Type" type="VoterIdType"/>
  </xsd:sequence>
</xsd:complexType>
```

3.7.16 The <VoterRecordsRequest> Element

The root element for *request* messages.

For defining items pertaining to the status and type of the voter records request and when it was generated. <*VoterRecordsRequest*> includes the <*VoterRegistration*> element to specify various information about the voter in question. It includes the <*AbsenteeBallotRequest*> element to handle a request for an absentee ballot; this request may be part of an FPCA form registration or may be submitted independently.

Table 3.34 – Elements for <VoterRecordsRequest>

Element	Multiplicity	Type	Element Description
< <i>AbsenteeBallotRequest</i> >	0 or 1	<i>AbsenteeBallotRequest</i>	To specify the type of absentee ballot request.
< <i>GeneratedDate</i> >	1	xsd:date	The date that the voter records request was generated.
< <i>Issuer</i> >	0 or 1	xsd:string	The name of the issuer of the voter records request transaction, e.g., <i>State of West Virginia Voter Registration Portal</i> .
< <i>OtherType</i> >	0 or 1	xsd:string	Used when < <i>RequestType</i> > value is <i>other</i> .
< <i>TransactionId</i> >	0 or 1	xsd:string	An identifier of the voter records request transaction.
< <i>Type</i> >	1 or more	Registration RequestType	The type of request, e.g., <i>registration</i> .
< <i>Vendor ApplicationId</i> >	0 or 1	xsd:string	An identifier of the vendor application generating the voter registration request, e.g., X-VRDB Version 3.1.a.
< <i>Voter Registration</i> >	1	Voter Registration	Specifies information about the voter who is the subject of the request.

Schema definition:

```

<xsd:complexType name="VoterRecordsRequest">
  <xsd:sequence>
    <xsd:element name="VoterRegistration" type="VoterRegistration"/>
    <xsd:element name="AbsenteeBallotRequest" type="AbsenteeBallotRequest"
      minOccurs="0"/>
    <xsd:element name="GeneratedDate" type="xsd:date"/>
    <xsd:element name="Issuer" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="TransactionId" type="xsd:string" minOccurs="0"/>
  
```

```
<xsd:element name="Type" type="RegistrationRequestType"
  maxOccurs="unbounded"/>
<xsd:element name="VendorApplicationId" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
```

3.7.17 The <VoterRecordsResponse> Element/Extension Base

The root element for *response* messages.

For defining items pertaining to the status of a response to a voter records request.

<*VoterRecordsResponse*> is an abstract element with three *xsi:type*s that get used according to the type of response:

- <*VoterRecordsResponse xsi:type="RegistrationAcknowledgement"*>, used to indicate an acknowledgement only.
- <*VoterRecordsResponse xsi:type="RegistrationRejection"*>, used to indicate a failure and the type of failure.
- <*VoterRecordsResponse xsi:type="RegistrationSuccess"*>, used to indication that a successful registration action occurred and the type of registration action, which may differ from the type of registration action requested.

<*VoterRecordsResponse*> optionally includes the <*TransactionId*> sub-element associated with the voter records request.

Table 3.35 – Elements for <VoterRecordsResponse>

Element	Multiplicity	Type	Element Description
< <i>TransactionId</i> >	0 or 1	xsd:string	Transaction ID associated with the voter records request.

Schema definition:

```

<xsd:complexType name="VoterRecordsResponse" abstract="true">
  <xsd:sequence>
    <xsd:element name="TransactionId" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

3.7.17.1 The RegistrationAcknowledgement xsi:type

Used in *response* messages.

For indicating that the request was received but action on the request is pending.

Schema Definition:

```
<xsd:complexType name="RegistrationAcknowledgement">
  <xsd:complexContent>
    <xsd:extension base="VoterRecordsResponse"/>
  </xsd:complexContent>
</xsd:complexType>
```

3.7.17.2 The RegistrationRejection xsi:type

Used in *response* messages.

For indicating that the request failed. The *<Error>* sub-element is used to indicate the type of error that occurred. The *<AdditionalDetails>* sub-element can be used to provide more information as to the rejection.

Table 3.36 – Elements for <VoterRecordsResponse xsi:type="RegistrationRejection">

Element	Multiplicity	Type	Element Description
<i><AdditionalDetails></i>	0 or more	xsd:string	Used to provide additional details as applicable.
<i><Error></i>	0 or more	RegistrationError	Used to indicate the type of error.
<i><OtherError></i>	0 or more	xsd:string	Used when <i><RegistrationError></i> value is <i>other</i> .

Schema definition:

```

<xsd:complexType name="RegistrationRejection">
  <xsd:complexContent>
    <xsd:extension base="VoterRecordsResponse">
      <xsd:sequence>
        <xsd:element name="AdditionalDetails" type="xsd:string" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="Error" type="RegistrationError" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="OtherError" type="xsd:string" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

3.7.17.3 The RegistrationSuccess xsi:type

Used in ***response*** messages.

For indicating a successful response to a request. The `<Action>` sub-element is used to indicate the action that occurred, which may differ from what was requested. For example, a request for a new voter registration may succeed, but if the voter was already registered, the response may indicate a registration update as opposed to a registration create.

The response also includes, optionally, information useful to the voter, including a description of the voter's precinct and polling place, as well as the districts (i.e., contests) associated with the precinct.

Table 3.37 – Elements for <VoterRecordsResponse xsi:type="RegistrationSuccess">

Element	Multiplicity	Type	Element Description
<code><Action></code>	0 or 1	<code>SuccessAction</code>	Used to indicate the action that occurred.
<code><OtherAction></code>	0 or 1	<code>xsd:string</code>	Used when <code><SuccessAction></code> value is <i>other</i> .
<code><Districts></code>	0 or more	<code>ReportingUnit</code>	The districts associated with the voter's precinct.
<code><EffectiveDate></code>	0 or 1	<code>xsd:date</code>	The effective date of the action.
<code><PollingPlace></code>	0 or 1	<code>ReportingUnit</code>	The voter's polling place.
<code><Precinct></code>	0 or 1	<code>ReportingUnit</code>	The voter's precinct.

Schema definition:

```

<xsd:complexType name="RegistrationSuccess">
  <xsd:complexContent>
    <xsd:extension base="VoterRecordsResponse">
      <xsd:sequence>
        <xsd:element name="Action" type="SuccessAction" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="OtherAction" type="xsd:string" minOccurs="0"
          maxOccurs="1"/>
        <xsd:element name="Districts" type="ReportingUnit" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="EffectiveDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="PollingPlace" type="ReportingUnit" minOccurs="0"/>
        <xsd:element name="Precinct" type="ReportingUnit" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

3.7.18 The <VoterRegistration> Element

Used in ***request*** messages.

<*VoterRecordsRequest*> includes this element to specify information about the voter.

All sub-elements are optional excepting <*Name*> and <*RegistrationAddress*> and <*RegistrationMethod*>. If the <*RegistrationAddressIsMailingAddress*> boolean is *true*, <*MailingAddress*> need not be included.

Table 3.38 – Elements for <VoterRegistration>

Element	Multiplicity	Type	Element Description
< <i>AdditionalInfo</i> >	0 or more	Additional Info	For including other information not specified by this schema.
< <i>BallotReceiptPreference</i> >	0 or more	Ballot Receipt Method	The voter's preference on how to receive their ballot in order from their most preferred method to least, used if an absentee ballot request.
< <i>ContactMethod</i> >	0 or more	Contact Method	How to contact the voter, listed in order of preference.
< <i>DateOfBirth</i> >	0 or 1	xsd:date	
< <i>Ethnicity</i> >	0 or 1	xsd:string	
< <i>Gender</i> >	0 or 1	xsd:string	Older systems may not understand values other than 'Male' or 'Female' (the only choices available on FPCA).
< <i>LastDateOfUSResidency</i> >	0 or 1	xsd:date	
< <i>MailingAddress</i> >	0 or 1	Address	Where the voter receives postal mail.
< <i>Name</i> >	1	Name	Voter's name.
< <i>OverseasEmployer</i> >			
< <i>Party</i> >	0 or 1	Party	Voter's political party.
< <i>PreviousName</i> >	0 or 1	Name	A voter's previous name.
< <i>PreviousRegistrationAddress</i> >	0 or 1	Address	Where the voter was previously registered.
< <i>PreviousSignature</i> >	0 or 1	Signature	Information about a previous voter signature on the registration form.
< <i>RegistrationAddress</i> >	1	Address	Where the voter is registered or requests to be registered.
< <i>RegistrationAddressIsMailingAddress</i> >	0 or 1	xsd:boolean	If set to <i>true</i> , < <i>MailingAddress</i> > need not be included.

Element	Multiplicity	Type	Element Description
<Registration Form>	0 or 1	Registration Form	If the request is for a voter registration, the registration form used by the voter.
<Registration Helper>	0 or 1	Registration Helper	Included if the registration involves a registration assistant organization.
<Other RegistrationForm >	0 or 1	xsd:string	Used when <RegistrationForm> value is <i>other</i> .
<Registration Method>	1	Registration Method	The method used by the voter to register.
<Other RegistrationForm >	0 or 1	xsd:string	Used when <RegistrationMethod> value is <i>other</i> .
<Registration Proxy>	0 or 1	Registration Method	Included if the registration request is via a proxy, e.g., the DMV.
<Signature>	0 or 1	Signature	Information about the voter signature on the registration form.
<Voter Classification>	0 or more	Voter Classification	How the voter is classified per assertions the voter has made on a registration form.
<VoterId>	0 or more	VoterId	Information to provide voter identity.

Schema definition:

```

<xsd:complexType name="VoterRegistration">
  <xsd:sequence>
    <xsd:element name="AdditionalInfo" type="AdditionalInfo" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element name="BallotReceiptPreference" type="BallotReceiptMethod"
      minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          The voter's preference on how to receive their ballot in order from their
          most preferred method to least. This property is only used if the
          VoterRecordsRequest Type specifies: ballot-request.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="ContactMethod" type="ContactMethod" minOccurs="0"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">
          Contact methods, listed in order of contact preference.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="DateOfBirth" type="xsd:date" minOccurs="0"/>
    <xsd:element name="Ethnicity" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Gender" type="xsd:string" minOccurs="0">

```

```

<xsd:annotation>
<xsd:documentation xml:lang="en">
  Older systems may not understand values other than 'Male' or 'Female'
  (the only choices available on FPCA)
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="LastDateOfUSResidency" type="xsd:date" minOccurs="0"/>
<xsd:element name="MailForwardingAddress" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="MailingAddress" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Name" type="Name"/>
<xsd:element name="OtherRegistrationForm" type="xsd:string" minOccurs="0"/>
<xsd:element name="OtherRegistrationMethod" type="xsd:string" minOccurs="0"/>
<xsd:element name="OverseasEmployer" type="xsd:string" minOccurs="0"/>
<xsd:element name="Party" type="Party" minOccurs="0"/>
<xsd:element name="PreviousName" type="Name" minOccurs="0"/>
<xsd:element name="PreviousRegistrationAddress" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="PreviousSignature" type="Signature" minOccurs="0"/>
<xsd:element name="RegistrationAddress">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group minOccurs="1" maxOccurs="1" ref="Address"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="RegistrationAddressIsMailingAddress" type="xsd:boolean"
  minOccurs="0"/>
<xsd:element name="RegistrationForm" type="RegistrationForm" minOccurs="0"/>
<xsd:element name="RegistrationHelper" type="RegistrationHelper"
  minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="RegistrationMethod" type="RegistrationMethod"/>
<xsd:element name="RegistrationProxy" type="RegistrationProxy"
  minOccurs="0"/>
<xsd:element name="SelectedLanguage" type="xsd:language" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The language specified by the voter, if any.
    </xsd:documentation>
  </xsd:annotation>

```

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="VoterClassification" type="VoterClassification"
  minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="VoterId" type="VoterId" minOccurs="0"
  maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
```

4 XML and JSON Usage Examples

This section contains several examples showing voter records request and responses in XML and JSON, all using the NVRA form. The examples are:

- Voter Registration Request - XML
- Voter Registration Request - JSON
- Voter Registration Response – XML

In the voter records request examples, note that a significant majority of the statements deal with specifying addresses using the FGDC standard [6].

4.1 Example 1: NVRA-style Voter Registration Request in XML

This example shows a fictitious digital NVRA-style registration request for “Jackie Nichole Davidson” in the State of Ohio using XML. This request is for an address update, and an example of the filled-out NVRA-style form is shown in Figure 6.

I am: <input checked="" type="checkbox"/> Registering as an Ohio voter <input checked="" type="checkbox"/> Updating my address <input type="checkbox"/> Updating my name			
1. Are you a U.S. citizen? <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No 2. Will you be at least 18 years of age on or before the next general election? <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No If you answered NO to either of the questions, do not complete this form.			
3. Last Name DAVIDSON	First Name JACKI	Middle Name or Initial NICHOLE	Jr., II, etc.
4. House Number and Street (Enter new address if changed) 6850 W FRANK AVE NW STE 1	Apt. or Lot #	5. City or Post Office NORTH CANTON	6. ZIP Code 44720
7. Additional Mailing Address (If necessary) 2264 W 4TH ST S SIDE A MANSFIELD OH 44906		8. County (where you live) STARK	FOR BOARD USE ONLY SEC4010 (rev. 4/15)
9. Birthdate (MM/DD/YYYY) (required) 11/09/1971	10. Ohio Driver's License number OR Last Four Digits of Social Security number (one form of ID required to be listed or provided) AB879456	11. Phone Number (voluntary) +1 (330) 123-9875	City, Village, Twp. Ward Precinct School Dist. Cong. Dist. Senate Dist. House Dist.
12. PREVIOUS ADDRESS IF UPDATING CURRENT REGISTRATION - Previous House Number and Street 740 POLARIS PKWY STE 1			
Previous City or Post Office LEWIS CENTER	Previous County MONTGOMERY	Previous State Ohio	
13. CHANGE OF NAME ONLY Former Legal Name		Former Signature	
14. I declare under penalty of election falsification I am a citizen of the United States, will have lived in this state for 30 days immediately preceding the next election, and will be at least 18 years of age at the time of the general election.		Your Signature	Date (MM/DD/YYYY)
			

Figure 6 - Example NVRA form for an address update request

The XML for the voter registration request that contains the information exported from the form is shown below. Note that in lines 15 through 18, the `<AdditionalInfo>` element is being used to indicate that the voter's language preference is English; this is needed because the XML (and JSON) schema does not include a voter-preferred language element but the application at the portal required it. Thus, `<AdditionalInfo>` can be used to add other elements that are required by the application but not present in the schema.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- DISCLAIMER: All persons and places listed below are fictitious -->
3  <VoterRecordsRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xmlns="NIST_V0_voter_records_interchange.xsd"
6      xsi:schemaLocation="NIST_V0_voter_records_interchange.xsd
7          NIST_V0_voter_records_interchange.xsd"
8      xmlns:addr="http://www.fgdc.gov/schemas/address/addr"
9      xmlns:addr_type="http://www.fgdc.gov/schemas/address/addr_type">
10     <GeneratedDate>2017-02-10</GeneratedDate>
11     <Issuer>Ohio Secretary of State</Issuer>
12     <TransactionId>2a642eb5-169e-4a3b-8899-adc7ea6d00d0</TransactionId>
13     <Type>registration</Type>
14     <VendorApplicationId>OLVR 2.0</VendorApplicationId>
15     <VoterRegistration>
16         <AdditionalInfo>
17             <Name>Language</Name>
18             <StringValue>en_US</StringValue>
19         </AdditionalInfo>
20         <ContactMethod>
21             <Type>phone</Type>
22             <Value>3301239875</Value>
23         </ContactMethod>
24         <ContactMethod>
25             <Type>email</Type>
26             <Value>FAKEEMAIL@AOL.COM</Value>
27         </ContactMethod>
28         <DateOfBirth>1971-11-09</DateOfBirth>
29         <MailingAddress>
30             <NumberedThoroughfareAddress_type>
31                 <addr:CompleteAddressNumber>
32                     <addr_type:AddressNumber>2264</addr_type:AddressNumber>
33                 </addr:CompleteAddressNumber>
34                 <addr:CompleteStreetName>
35                     <addr_type:StreetNamePreDirectional>W</addr_type:StreetNamePreDirectional>
36                     <addr_type:StreetName>4TH</addr_type:StreetName>
37                     <addr_type:StreetNamePostType>ST</addr_type:StreetNamePostType>
38                     <addr_type:StreetNamePostDirectional>S</addr_type:StreetNamePostDirectional>
39                 </addr:CompleteStreetName>
40                 <addr:CompleteSubaddress>
41                     <addr_type:SubaddressElement>
42                         <addr_type:SubaddressType>SIDE</addr_type:SubaddressType>
43                         <addr_type:SubaddressIdentifier>A</addr_type:SubaddressIdentifier>
44                     </addr_type:SubaddressElement>
45                 </addr:CompleteSubaddress>
46                 <addr_type:CompletePlaceName>
47                     <addr_type:PlaceName
48                         PlaceNameType="MunicipalJurisdiction">MANSFIELD</addr_type:PlaceName>
49                         <addr_type:PlaceName PlaceNameType="County"/>
50                     </addr_type:CompletePlaceName>
51                     <addr_type:StateName>OH</addr_type:StateName>
52                     <addr_type:ZipCode>44906</addr_type:ZipCode>
53                 </NumberedThoroughfareAddress_type>

```

```

54      </MailingAddress>
55      <Name>
56          <FirstName>JACKI</FirstName>
57          <LastName>DAVIDSON</LastName>
58          <MiddleName>NICHOLE</MiddleName>
59      </Name>
60      <PreviousRegistrationAddress>
61          <NumberedThoroughfareAddress_type>
62              <addr:CompleteAddressNumber>
63                  <addr_type:AddressNumber>740</addr_type:AddressNumber>
64              </addr:CompleteAddressNumber>
65              <addr:CompleteStreetName>
66                  <addr_type:StreetNamePreDirectional/>
67                  <addr_type:StreetName>POLARIS</addr_type:StreetName>
68                  <addr_type:StreetNamePostType>PKWY</addr_type:StreetNamePostType>
69                  <addr_type:StreetNamePostDirectional/>
70              </addr:CompleteStreetName>
71              <addr:CompleteSubaddress>
72                  <addr_type:SubaddressElement>
73                      <addr_type:SubaddressType>STE</addr_type:SubaddressType>
74                      <addr_type:SubaddressIdentifier>1</addr_type:SubaddressIdentifier>
75                  </addr_type:SubaddressElement>
76              </addr:CompleteSubaddress>
77              <addr_type:CompletePlaceName>
78                  <addr_type:PlaceName PlaceNameType="MunicipalJurisdiction">LEWIS
79                      CENTER</addr_type:PlaceName>
80                  <addr_type:PlaceName PlaceNameType="County">MONTGOMERY</addr_type:PlaceName>
81              </addr_type:CompletePlaceName>
82              <addr_type:StateName>OH</addr_type:StateName>
83              <addr_type:ZipCode>43035</addr_type:ZipCode>
84          </NumberedThoroughfareAddress_type>
85      </PreviousRegistrationAddress>
86      <RegistrationAddress>
87          <NumberedThoroughfareAddress_type>
88              <addr:CompleteAddressNumber>
89                  <addr_type:AddressNumber>6850</addr_type:AddressNumber>
90              </addr:CompleteAddressNumber>
91              <addr:CompleteStreetName>
92                  <addr_type:StreetNamePreDirectional>W</addr_type:StreetNamePreDirectional>
93                  <addr_type:StreetName>FRANK</addr_type:StreetName>
94                  <addr_type:StreetNamePostType>AVE</addr_type:StreetNamePostType>
95                  <addr_type:StreetNamePostDirectional>NW</addr_type:StreetNamePostDirectional>
96              </addr:CompleteStreetName>
97              <addr:CompleteSubaddress>
98                  <addr_type:SubaddressElement>
99                      <addr_type:SubaddressType>STE</addr_type:SubaddressType>
100                     <addr_type:SubaddressIdentifier>1</addr_type:SubaddressIdentifier>
101                 </addr_type:SubaddressElement>
102             </addr:CompleteSubaddress>
103             <addr_type:CompletePlaceName>
104                 <addr_type:PlaceName PlaceNameType="MunicipalJurisdiction">NORTH
105                     CANTON</addr_type:PlaceName>
106                     <addr_type:PlaceName PlaceNameType="County">STARK </addr_type:PlaceName>
107                 </addr_type:CompletePlaceName>
108                 <addr_type:StateName>OH</addr_type:StateName>
109                 <addr_type:ZipCode>44720</addr_type:ZipCode>
110             </NumberedThoroughfareAddress_type>
111         </RegistrationAddress>
112         <RegistrationForm>nvra</RegistrationForm>
113         <RegistrationMethod>voter-via-mail</RegistrationMethod>
114         <VoterClassification>
115             <Assertion>yes</Assertion>

```

```
116      <Type>eighteen-on-election-day</Type>
117  </VoterClassification>
118  <VoterClassification>
119    <Assertion>yes</Assertion>
120    <Type>united-states-citizen</Type>
121  </VoterClassification>
122  <VoterClassification>
123    <Assertion>yes</Assertion>
124    <OtherType>ohio-resident</OtherType>
125    <Type>other</Type>
126  </VoterClassification>
127  <VoterClassification>
128    <Assertion>yes</Assertion>
129    <OtherType>bmv-authorization</OtherType>
130    <Type>other</Type>
131  </VoterClassification>
132  <VoterId>
133    <AttestNoSuchId>false</AttestNoSuchId>
134    <StringValue>AB879456</StringValue>
135    <Type>drivers-license</Type>
136  </VoterId>
137  <VoterId>
138    <AttestNoSuchId>true</AttestNoSuchId>
139    <Type>ssn4</Type>
140  </VoterId>
141  </VoterRegistration>
142 </VoterRecordsRequest>
```

4.2 Example 2: NVRA Voter Registration Request in JSON

This example shows a fictitious NVRA-style voter registration request for Jane A. Doe in the State of Ohio using JSON, shown in Figure 7.

The form contains the following information:

- I am:** Registering as an Ohio voter Updating my address Updating my name
- 1. Are you a U.S. citizen?** Yes No
- 2. Will you be at least 18 years of age on or before the next general election?** Yes No
If you answered NO to either of the questions, do not complete this form.
- 3. Last Name:** Doe
- First Name:** Jane
- Middle Name or Initial:** A
- 4. House Number and Street (Enter new address if changed):** 3318 BELMONT AVE
- Apt. or Lot #:**
- 5. City or Post Office:** YOUNGSTOWN
- 6. ZIP Code:** 44505
- 7. Additional Mailing Address (if necessary):** 2299 E. STATE ST, SALEM, OH 44460
- 8. County (where you live):** Mahoning
- 9. Birthdate (MM/DD/YYYY) (required):** 11/12/1993
- 10. Ohio Driver's License number OR Last Four Digits of Social Security number (one form of ID required to be listed or provided):** AA999888
- 11. Phone Number (voluntary):** +1 (330) 614-8004
- 12. PREVIOUS ADDRESS IF UPDATING CURRENT REGISTRATION - Previous House Number and Street:** (Blank)
- Previous City or Post Office:** (Blank)
- Previous County:** (Blank)
- Previous State:** Ohio
- 13. CHANGE OF NAME ONLY Former Legal Name:** (Blank)
- Former Signature:** (Blank)
- 14. I declare under penalty of election falsification I am a citizen of the United States, will have lived in this state for 30 days immediately preceding the next election, and will be at least 18 years of age at the time of the general election.**
- Your Signature:** (Signature box)
- Date (MM/DD/YYYY):** (Blank)
- FOR BOARD USE ONLY SEC4010 (rev. 4/15):**
 - City, Village, Twp.
 - Ward
 - Precinct
 - School Dist.
 - Cong. Dist.
 - Senate Dist.
 - House Dist.

Figure 7 - Example NVRA form for an address update request

An example of the JSON statements for the voter registration request is shown below. Again, “AdditionalInfo” is being used in lines 68 through 77 to add information for elements not included in the JSON (and XML) schema, namely for the voter’s preferred language and whether the voter wishes to volunteer as a poll worker.

```

1  {
2    "GeneratedDate": "2017-7-30",
3    "Issuer": "Ohio Secretary of State",
4    "TransactionId": "0bb1bfd7-4316-42be-99d5-9c9e3bb9ccc0",
5    "Type": [
6      "registration"
7    ],
8    "VendorApplicationId": "OLVR 2.0",
9    "VoterRegistration": {
10      "DateOfBirth": "1993-11-12T00:00:00",
11      "VoterId": [
12        {
13          "Type": "drivers-license",

```

```

14         "StringValue": "AA999888",
15         "AttestNoSuchId": false
16     },
17     {
18         "Type": "ssn4",
19         "StringValue": null,
20         "AttestNoSuchId": true
21     }
22 ],
23 "Name": [
24     "FirstName": "JANE",
25     "MiddleName": [
26         "A"
27     ],
28     "LastName": "DOE",
29     "Suffix": null
30 ],
31 "VoterClassification": [
32     {
33         "Assertion": true,
34         "Type": "Item18onelectionday",
35         "OtherType": null
36     },
37     {
38         "Assertion": true,
39         "Type": "unitedstatescitizen",
40         "OtherType": null
41     },
42     {
43         "Assertion": true,
44         "Type": "other",
45         "OtherType": "swear-accuracy"
46     },
47     {
48         "Assertion": true,
49         "Type": "other",
50         "OtherType": "filled-on-own-behalf"
51     },
52     {
53         "Assertion": true,
54         "Type": "other",
55         "OtherType": "ohio-resident"
56     },
57     {
58         "Assertion": true,
59         "Type": "other",
60         "OtherType": "bmv-authorization"
61     },
62     {
63         "Assertion": true,
64         "Type": "other",
65         "OtherType": "meets-all-requirements"
66     }
67 ],
68 "AdditionalInfo": [
69     {
70         "Name": "Language",
71         "StringValue": "en_US"
72     },
73     {
74         "Name": "IsPollWorker",
75         "StringValue": "false"

```

```

76      }
77  ],
78  "ContactMethod": [
79    {
80      "Type": "phone",
81      "Value": "(330) 614-8004"
82    },
83    {
84      "Type": "email",
85      "Value": "JDOE@TESTEMAIL.COM"
86    }
87  ],
88  "RegistrationAddress": {
89    "NumberedThoroughfareAddress_type": {
90      "CompleteAddressNumber": {
91        "AddressNumber": "3818"
92      },
93      "CompleteStreetName": {
94        "StreetNamePreDirectional": {
95          "Value": null
96        },
97        "StreetName": "BELMONT",
98        "StreetNamePostType": {
99          "Value": "AVE"
100       },
101      "StreetNamePostDirectional": {
102        "Value": null
103      }
104    },
105    "CompletePlaceName": [
106      {
107        "PlaceName": [
108          {
109            "PlaceNameTypeSpecified": true,
110            "PlaceNameType": "MunicipalJurisdiction",
111            "Value": "YOUNGSTOWN"
112          },
113          {
114            "PlaceNameTypeSpecified": true,
115            "PlaceNameType": "County",
116            "Value": "78"
117          }
118        ]
119      }
120    ],
121    "StateName": [
122      "OH"
123    ],
124    "ZipCode": [
125      44505
126    ],
127    "AddressId": 1368029993
128  }
129},
130 "RegistrationMethod": "voterviamail",
131 "RegistrationFormSpecified": true,
132 "RegistrationForm": "other",
133 "OtherRegistrationForm": "4010",
134 "MailingAddress": {
135   "NumberedThoroughfareAddress_type": {
136     "CompleteAddressNumber": {
137       "AddressNumber": "2299"

```

```
138 },
139 "CompleteStreetName": {
140     "StreetNamePreDirectional": {
141         "Value": "E"
142     },
143     "StreetName": "STATE",
144     "StreetNamePostType": {
145         "Value": "ST"
146     },
147     "StreetNamePostDirectional": {
148         "Value": null
149     }
150 },
151 "CompletePlaceName": [
152     {
153         "PlaceName": [
154             {
155                 "PlaceNameTypeSpecified": true,
156                 "PlaceNameType": "MunicipalJurisdiction",
157                 "Value": "SALEM"
158             },
159             {
160                 "PlaceNameTypeSpecified": true,
161                 "PlaceNameType": "County",
162                 "Value": null
163             }
164         ]
165     }
166 ],
167 "StateName": [
168     "OH"
169 ],
170 "ZipCode": [
171     44460
172 ],
173     }
174   }
175 }
176 }
```

4.3 Example 3: NVRA Voter Records Response in XML

Form No. 10-J. Prescribed by Secretary of State (06-14)

ACKNOWLEDGMENT NOTICE
R.C. 3501.01(V), 3503.19(C)(1)

Your application to register to vote or update your registration has been received and accepted. For the purpose of voting, you are assigned to:

Precinct	AKRON 4-F
City, Township, Village	Akron
Your polling place is:	PERKINS BUILDING
	630 MULL AVE
	AKRON, OH 44313

Your application to register to vote or update your registration has been rejected because:

<input type="checkbox"/> Form was not signed	<input type="checkbox"/> Address was incomplete
<input type="checkbox"/> Name given was incomplete	<input type="checkbox"/> Birth date was not supplied
<input type="checkbox"/> Other required information not provided:	

Figure 8 - Populated NVRA voter registration response form

This final example shows a fictitious digital NVRA-style registration response, successful, in the State of Ohio using XML. An example of the populated response form is shown in Figure 8. The XML for the voter registration response that contains the information used to populate the form is as follows:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- DISCLAIMER: All persons and places listed below are fictitious -->
3  <VoterRecordsResponse xsi:type="RegistrationSuccess" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5    xmlns="NIST_V0_voter_records_interchange.xsd"
6    xsi:schemaLocation="NIST_V0_voter_records_interchange.xsd
7      NIST_V0_voter_records_interchange.xsd"
8    xmlns:addr="http://www.fgdc.gov/schemas/address/addr"
9    xmlns:addr_type="http://www.fgdc.gov/schemas/address/addr_type">
10   <Action>registration-created</Action>
11   <Districts>
12     <Name>Akron</Name>
13     <Type>municipality</Type>
14   </Districts>
15   <EffectiveDate>2017-07-31</EffectiveDate>
16   <ElectionAdministration>
17     <ContactMethod>
18       <Type>phone</Type>
19       <Value>3306435200</Value>
20     </ContactMethod>
21     <Location>
```

```
22      <Address>
23          <GeneralAddressClass_type>
24              <addr:USPSGeneralDeliveryPoint>470 GRANT ST</addr:USPSGeneralDeliveryPoint>
25                  <addr_type:PlaceStateZip>AKRON, OH 44311</addr_type:PlaceStateZip>
26          </GeneralAddressClass_type>
27      </Address>
28  </Location>
29      <Name>Summit</Name>
30  </ElectionAdministration>
31  <PollingPlace>
32      <Location>
33          <Address>
34              <GeneralAddressClass_type>
35                  <addr:USPSGeneralDeliveryPoint>630 MULL AVE</addr:USPSGeneralDeliveryPoint>
36                      <addr_type:PlaceStateZip>AKRON, OH 44313</addr_type:PlaceStateZip>
37              </GeneralAddressClass_type>
38          </Address>
39      </Location>
40      <Name>PERKINS BUILDING</Name>
41      <Type>polling-place</Type>
42  </PollingPlace>
43  <Locality>
44      <Name>AKRON 4-F</Name>
45      <Type>precinct</Type>
46  </Locality>
47 </VoterRecordsResponse>
```

Appendix A—Acronyms

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

CDF	Common Data Format
EAC	Election Assistance Commission
EAVS	EAC Election Administration and Voting Survey
FIPS	Federal Information Processing Standard
FPCA	Federal Post Card Application
FWAB	Federal Write-in Absentee Ballot
JSON	JavaScript Object Notation
MMS	Multimedia Messaging Service
MIME	Multipurpose Internet Mail Extensions
NIST	National Institute of Standards and Technology
NVRA	National Voter Registration Act
OCD-ID	Open Civic Data Identifiers
OVR	Online Voter Registration
SMS	Short Message Service
SP	Special Publication
UML	Unified Modeling Language
UOCAVA	Uniform and Overseas Citizens Assistance in Voting Act
VR	Voter Registration
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:*”. Synonyms are preceded with “*Syn.:*”.

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, administrative divisions in which voters are entitled to vote in contests that are specific to that division, such as those for state senators and delegates.
Polling place:	Location at which voters cast 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 sometimes in 1-to-1 correspondence with a precinct but in some cases, may represent multiple precincts as with 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.
Registration assistant:	An organization whose purpose includes assisting voters in registering to vote.
Registration proxy:	An organization that submits a voter registration request on behalf of the voter, e.g., a DMV office that submits a voter registration request for a voter.
Registration witness:	An individual who witnesses a voter’s registration, i.e., the voter signing his/her registration form.
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.
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.

Appendix C—References

- [1] W3C, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation, November 26, 2008, <http://www.w3.org/TR/xml/> [accessed 2/1/2016].
- [2] JavaScript Object Notation, <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> [accessed 2/1/2016].
- [3] Election Assistance Commission, *Election Administration and Voting Survey*, http://www.eac.gov/research/election_administration_and_voting_survey.aspx [accessed 2/1/2016].
- [4] Federal Voting Assistance Program (FVAP), Federal Post Card Application (FPCA), <https://www.fvap.gov/eo/overview/materials/forms> [accessed 2/1/2016].
- [5] Object Management Group (OMG), *UML Specification version 1.1* (OMG document ad/97-08-11) September 22, 2011, <http://omg.org/> [accessed 2/1/2016].
- [6] Federal Geographic Data Committee (FGDC), *United States Thoroughfare, Landmark, And Postal Address Data Standard*, http://www.fgdc.gov/standards/projects/FGDC-standards-projects/address-data/index_html [accessed 2/1/2016].
- [7] Electronic Registration Information Center (ERIC), <http://www.ericstates.org/> [accessed 9/5/2017].
- [8] Election Assistance Commission, Election Administration and Voting Survey (EAVS), <http://www.eac.gov> [accessed 9/5/2017].
- [9] Federal Geographic Data Committee (FGDC), <https://www.fgdc.gov/>.
- [10] U.S. Postal Service (USPS), “Postal Addressing Standards” Publication 28, April 2010, <http://pe.usps.gov/cpim/ftp/pubs/Pub28/Pub28.pdf> [accessed 9/5/2017].
- [11] Open Civic Data, *OCD Identifiers*, <http://opencivicdata.readthedocs.org/en/latest/ocdids.html> [accessed 2/1/2016].

Appendix D—File Download Locations

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

<http://vote.nist.gov>

These files are:

- This specification,
- UML model,
- XML and JSON schemas,
- Example files, and
- Validation tools.

Appendix E—XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified"
targetNamespace="NIST_V0_voter_records_interchange.xsd" version="0.0"
xmlns="NIST_V0_voter_records_interchange.xsd"
xmlns:addr="http://www.fgdc.gov/schemas/address/addr"
  <!-- ===== Imports ===== -->
  <xsd:import namespace="http://www.fgdc.gov/schemas/address/addr"
schemaLocation="https://www.fgdc.gov/schemas/address/addr.xsd"/>
  <!-- ===== Roots ===== -->
  <xsd:element name="VoterRecordsRequest" type="VoterRecordsRequest"/>
  <xsd:element name="VoterRecordsResponse" type="VoterRecordsResponse"/>
  <!-- ===== Enumerations ===== -->
  <xsd:simpleType name="AssertionValue">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="no"/>
      <xsd:enumeration value="yes"/>
      <xsd:enumeration value="unknown"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="BallotReceiptMethod">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="email"/>
      <xsd:enumeration value="email-or-online"/>
      <xsd:enumeration value="fax"/>
      <xsd:enumeration value="mail"/>
      <xsd:enumeration value="online"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ContactMethodType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="email"/>
      <xsd:enumeration value="phone"/>
      <xsd:enumeration value="other"/>
    </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="PhoneCapability">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="fax"/>
      <xsd:enumeration value="mms"/>
      <xsd:enumeration value="sms"/>
      <xsd:enumeration value="voice"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="RegistrationError">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="identity-lookup-failed"/>
      <xsd:enumeration value="incomplete"/>
      <xsd:enumeration value="incomplete-address"/>
      <xsd:enumeration value="incomplete-birth-date"/>
      <xsd:enumeration value="incomplete-name"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

<xsd:enumeration value="incomplete-signature"/>
<xsd:enumeration value="ineligible"/>
<xsd:enumeration value="invalid-form"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RegistrationForm">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="f pca"/>
    <xsd:enumeration value="nvra"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RegistrationHelperType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="assistant"/>
    <xsd:enumeration value="witness"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RegistrationMethod">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="armed-forces-recruitment-office"/>
    <xsd:enumeration value="motor-vehicle-office"/>
    <xsd:enumeration value="other-agency-designated-by-state"/>
    <xsd:enumeration value="public-assistance-office"/>
    <xsd:enumeration value="registration-drive-from-advocacy-group-or-political-party"/>
    <xsd:enumeration value="state-funded-agency-serving-persons-with-disabilities"/>
    <xsd:enumeration value="voter-via-election-registrars-office"/>
    <xsd:enumeration value="voter-via-email"/>
    <xsd:enumeration value="voter-via-fax"/>
    <xsd:enumeration value="voter-via-internet"/>
    <xsd:enumeration value="voter-via-mail"/>
    <xsd:enumeration value="unknown"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RegistrationProxyType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="armed-forces-recruitment-office"/>
    <xsd:enumeration value="motor-vehicle-office"/>
    <xsd:enumeration value="other-agency-designated-by-state"/>
    <xsd:enumeration value="public-assistance-office"/>
    <xsd:enumeration value="registration-drive-from-advocacy-group-or-political-party"/>
    <xsd:enumeration value="state-funded-agency-serving-persons-with-disabilities"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RegistrationRequestType">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The specific type of message being sent. Each type indicates a desired result by the system receiving the message. The meanings of each type will need to be explicitly stated.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">

```

This attribute is optional, and if not specified (or other), the required Action attribute will determine the message semantics generically.

For example: 'address update' indicates that only the voter's address should be updated on an existing voter registration.

'address update' requests would specify the 'update' RegistrationAction.

</xsd:documentation>

</xsd:annotation>

<xsd:restriction base="xsd:string">

```

<xsd:enumeration value="ballot-request"/>
<xsd:enumeration value="registration"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ReportingUnitType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="ballot-batch"/>
<xsd:enumeration value="ballot-style-area"/>
<xsd:enumeration value="borough"/>
<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="SignatureSource">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="dmv"/>
<xsd:enumeration value="local"/>
<xsd:enumeration value="state"/>
<xsd:enumeration value="voter"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SignatureType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="dynamic"/>
<xsd:enumeration value="electronic"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SuccessAction">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="address-updated"/>
<xsd:enumeration value="name-updated"/>
<xsd:enumeration value="registration-cancelled"/>
<xsd:enumeration value="registration-created"/>
<xsd:enumeration value="registration-updated"/>
<xsd:enumeration value="status-updated"/>
<xsd:enumeration value="other"/>

```

```

    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="VoterClassificationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="activated-national-guard"/>
      <xsd:enumeration value="active-duty"/>
      <xsd:enumeration value="active-duty-spouse-or-dependent"/>
      <xsd:enumeration value="citizen-abroad-intent-to-return"/>
      <xsd:enumeration value="citizen-abroad-return-uncertain"/>
      <xsd:enumeration value="citizen-abroad-never-resided"/>
      <xsd:enumeration value="deceased"/>
      <xsd:enumeration value="declared-incompetent"/>
      <xsd:enumeration value="eighteen-on-election-day"/>
      <xsd:enumeration value="felon"/>
      <xsd:enumeration value="permanently-denied"/>
      <xsd:enumeration value="protected-voter"/>
      <xsd:enumeration value="restored-felon"/>
      <xsd:enumeration value="united-states-citizen"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="VoterIdType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="drivers-license"/>
      <xsd:enumeration value="local-voter-registration-id"/>
      <xsd:enumeration value="ssn"/>
      <xsd:enumeration value="ssn4"/>
      <xsd:enumeration value="state-id"/>
      <xsd:enumeration value="state-voter-registration-id"/>
      <xsd:enumeration value="unspecified-document"/>
      <xsd:enumeration value="unspecified-document-with-name-and-address"/>
      <xsd:enumeration value="unspecified-document-with-photo-identification"/>
      <xsd:enumeration value="unknown"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- ===== Interfaces Defined ===== -->
  <!-- == Interface Address == -->
  <xsd:group name="Address">
    <xsd:choice>
      <xsd:element name="CommunityAddress_type" type="addr:CommunityAddress_type"/>
      <xsd:element name="FourNumberAddressRange_type" type="addr:FourNumberAddressRange_type"/>
      <xsd:element name="GeneralAddressClass_type" type="addr:GeneralAddressClass_type"/>
      <xsd:element name="IntersectionAddress_type" type="addr:IntersectionAddress_type"/>
      <xsd:element name="LandmarkAddress_type" type="addr:LandmarkAddress_type"/>
      <xsd:element name="NumberedThoroughfareAddress_type"
type="addr:NumberedThoroughfareAddress_type"/>
      <xsd:element name="TwoNumberAddressRange_type" type="addr:TwoNumberAddressRange_type"/>
      <xsd:element name="USPSGeneralDeliveryOffice_type"
type="addr:USPSGeneralDeliveryOffice_type"/>
      <xsd:element name="USPSPostalDeliveryBox_type" type="addr:USPSPostalDeliveryBox_type"/>
      <xsd:element name="USPSPostalDeliveryRoute_type" type="addr:USPSPostalDeliveryRoute_type"/>
      <xsd:element name="UnnumberedThoroughfareAddress_type"
type="addr:UnnumberedThoroughfareAddress_type"/>
    </xsd:choice>
  </xsd:group>
  <!-- ===== Classes ===== -->
  <xsd:complexType name="AdditionalInfo">
    <xsd:sequence>
      <xsd:element name="FileValue" type="File" minOccurs="0"/>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="StringValue" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ContactMethod">
        <xsd:sequence>
            <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Type" type="ContactMethodType"/>
            <xsd:element name="Value" type="xsd:string">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        The value of the ContactMethod. This will be the text value of the phone number,
                        email address, or other mechanism. The values must be free of any formatting characters, such as
                        parentheses or dashes for a phone number.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ElectionAdministration">
        <xsd:sequence>
            <xsd:element name="ContactMethod" type="ContactMethod" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="Location" type="Location" minOccurs="0"/>
            <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Uri" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ExternalIdentifier">
        <xsd:sequence>
            <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Type" type="IdentifierType"/>
            <xsd:element name="Value" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="File">
        <xsd:simpleContent>
            <xsd:extension base="xsd:base64Binary">
                <xsd:attribute name="fileName" type="xsd:string"/>
                <xsd:attribute name="mimeType" type="xsd:string"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="Image">
        <xsd:complexContent>
            <xsd:extension base="File">
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="LatLng">
        <xsd:sequence>
            <xsd:element name="Latitude" type="xsd:float"/>
            <xsd:element name="Longitude" type="xsd:float"/>
            <xsd:element name="Source" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Location">
        <xsd:sequence>
            <xsd:element name="Address" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>

```

```

    </xsd:element>
    <xsd:element name="Directions" type="xsd:string" minOccurs="0"/>
    <xsd:element name="LatLng" type="LatLang" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Name">
    <xsd:sequence>
        <xsd:element name="FirstName" type="xsd:string" minOccurs="0"/>
        <xsd:element name="FullName" 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="Prefix" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Suffix" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Party">
    <xsd:sequence>
        <xsd:element name="Abbreviation" type="xsd:string" minOccurs="0"/>
        <xsd:element name="ExternalIdentifier" type="ExternalIdentifier" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="Name" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PhoneContactMethod">
    <xsd:complexContent>
        <xsd:extension base="ContactMethod">
            <xsd:sequence>
                <xsd:element name="Capability" type="PhoneCapability" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RegistrationAcknowledgement">
    <xsd:complexContent>
        <xsd:extension base="VoterRecordsResponse">
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RegistrationHelper">
    <xsd:sequence>
        <xsd:element name="Address" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="Name" type="Name" minOccurs="0"/>
        <xsd:element name="Phone" type="PhoneContactMethod" minOccurs="0"/>
        <xsd:element name="Signature" type="Signature" minOccurs="0"/>
        <xsd:element name="Type" type="RegistrationHelperType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RegistrationProxy">
    <xsd:sequence>
        <xsd:element name="Address" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

    </xsd:element>
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OriginTransactionId" type="xsd:string" minOccurs="0"/>
    <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Phone" type="PhoneContactMethod" minOccurs="0"/>
    <xsd:element name="TimeStamp" type="xsd:date" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                The date this source received the request.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="Type" type="RegistrationProxyType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RegistrationRejection">
    <xsd:complexContent>
        <xsd:extension base="VoterRecordsResponse">
            <xsd:sequence>
                <xsd:element name="AdditionalDetails" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="Error" type="RegistrationError" minOccurs="0"
maxOccurs="unbounded"/>
                    <xsd:element name="OtherError" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
<xsd:complexType name="RegistrationSuccess">
    <xsd:complexContent>
        <xsd:extension base="VoterRecordsResponse">
            <xsd:sequence>
                <xsd:element name="Action" type="SuccessAction" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="District" type="ReportingUnit" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="EffectiveDate" type="xsd:date" minOccurs="0"/>
                <xsd:element name="ElectionAdministration" type="ElectionAdministration"
minOccurs="0"/>
                    <xsd:element name="Locality" type="ReportingUnit" minOccurs="0" maxOccurs="unbounded"/>
                    <xsd:element name="PollingPlace" type="ReportingUnit" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
<xsd:complexType name="ReportingUnit">
    <xsd:sequence>
        <xsd:element name="ExternalIdentifier" type="ExternalIdentifier" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="IsDistracted" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="Location" type="Location" minOccurs="0"/>
        <xsd:element name="Name" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Name of the reporting unit.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="ReportingUnitType">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Type of reporting unit, e.g., state, jurisdiction, district, etc.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

This field is a key into the NIST maintained registry of GpUnit types.

The key specifies the geo-political category of the locality, the type of locality, and optionally a sub-type.

If an 'Other' type or subtype is specified, then it will be defined via the OtherType value.

```

        </xsd:documentation>
        </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Signature">
    <xsd:sequence>
        <xsd:element name="Date" type="xsd:date" minOccurs="0"/>
        <xsd:element name="FileValue" type="Image" minOccurs="0"/>
        <xsd:element name="OtherSource" type="xsd:string" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Source" type="SignatureSource" minOccurs="0"/>
        <xsd:element name="Type" type="SignatureType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VoterClassification">
    <xsd:sequence>
        <xsd:element name="Assertion" type="AssertionValue"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="VoterClassificationType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VoterId">
    <xsd:sequence>
        <xsd:element name="AttestNoSuchId" type="xsd:boolean" minOccurs="0"/>
        <xsd:element name="DateOfIssuance" type="xsd:date" minOccurs="0"/>
        <xsd:element name="FileValue" type="File" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="StringValue" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="VoterIdType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VoterRecordsRequest">
    <xsd:sequence>
        <xsd:element name="GeneratedDate" type="xsd:date">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    The date of this message's creation
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Issuer" type="xsd:string" minOccurs="0"/>
        <xsd:element name="OtherType" type="xsd:string" minOccurs="0"/>
        <xsd:element name="TransactionId" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Type" type="RegistrationRequestType" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    The specific type of request that is being made. Each type indicates a desired result by the system receiving the message. The meanings of each type will need to be explicitly stated.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

This attribute is optional, and if not specified (or other), the required Action attribute will determine the message semantics generically.

For example: 'address update' indicates that only the voter's address should be updated on an existing voter registration.

'address update' requests would specify the 'update' RegistrationAction.

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="VendorApplicationId" type="xsd:string" minOccurs="0"/>
<xsd:element name="VoterRegistration" type="VoterRegistration"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VoterRecordsResponse" abstract="true">
    <xsd:sequence>
        <xsd:element name="TransactionId" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VoterRegistration">
    <xsd:sequence>
        <xsd:element name="AdditionalInfo" type="AdditionalInfo" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="BallotReceiptPreference" type="BallotReceiptMethod" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    The voter's preference on how to receive their ballot in order from their most
preferred method to least.
                </xsd:documentation>
            </xsd:annotation>
            This property is only used if the VoterRecordsRequest Type specifies: ballot-request.
        </xsd:element>
        <xsd:element name="ContactMethod" type="ContactMethod" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Contact methods, listed in order of contact preference.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="DateOfBirth" type="xsd:date" minOccurs="0"/>
        <xsd:element name="Ethnicity" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Gender" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Older systems may not understand values other than 'Male' or 'Female' (the only
choices available on FPCA)
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="LastDateOfUSResidency" type="xsd:date" minOccurs="0"/>
        <xsd:element name="MailForwardingAddress" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="MailingAddress" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="Name" type="Name"/>
        <xsd:element name="OtherRegistrationForm" type="xsd:string" minOccurs="0"/>
        <xsd:element name="OtherRegistrationMethod" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="OverseasEmployer" type="xsd:string" minOccurs="0"/>
<xsd:element name="Party" type="Party" minOccurs="0"/>
<xsd:element name="PreviousName" type="Name" minOccurs="0"/>
<xsd:element name="PreviousRegistrationAddress" minOccurs="0">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:group minOccurs="0" maxOccurs="1" ref="Address"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="PreviousSignature" type="Signature" minOccurs="0"/>
<xsd:element name="RegistrationAddress">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:group minOccurs="1" maxOccurs="1" ref="Address"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="RegistrationAddressIsMailingAddress" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="RegistrationForm" type="RegistrationForm" minOccurs="0"/>
<xsd:element name="RegistrationHelper" type="RegistrationHelper" minOccurs="0"
maxOccurs="unbounded"/>
<xsd:element name="RegistrationMethod" type="RegistrationMethod"/>
<xsd:element name="RegistrationProxy" type="RegistrationProxy" minOccurs="0"/>
<xsd:element name="SelectedLanguage" type="xsd:language" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            The language specified by the voter, if any.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="VoterClassification" type="VoterClassification" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="VoterId" type="VoterId" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Appendix F—JAVA Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "definitions": {
    "FGDCAddressStandard.CommunityAddress_type": {
      "required": [
        "@type"
      ],
      "additionalProperties": false,
      "properties": {
        "@type": {
          "enum": [
            "FGDCAddressStandard.CommunityAddress_type"
          ],
          "type": "string"
        }
      },
      "type": "object"
    },
    "FGDCAddressStandard.FourNumberAddressRange_type": {
      "required": [
        "@type"
      ],
      "additionalProperties": false,
      "properties": {
        "@type": {
          "enum": [
            "FGDCAddressStandard.FourNumberAddressRange_type"
          ],
          "type": "string"
        }
      },
      "type": "object"
    },
    "FGDCAddressStandard.GeneralAddressClass_type": {
      "required": [
        "@type"
      ],
      "additionalProperties": false,
      "properties": {
        "@type": {
          "enum": [
            "FGDCAddressStandard.GeneralAddressClass_type"
          ],
          "type": "string"
        }
      },
      "type": "object"
    },
    "FGDCAddressStandard.IntersectionAddress_type": {
      "required": [
        "@type"
      ],
      "additionalProperties": false,
      "properties": {
        "@type": {
          "enum": [
            "FGDCAddressStandard.IntersectionAddress_type"
          ],
          "type": "string"
        }
      }
    }
  }
}
```

```
        },
        "type": "object"
    },
    "FGDCAddressStandard.LandmarkAddress_type": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "FGDCAddressStandard.LandmarkAddress_type"
                ],
                "type": "string"
            }
        },
        "type": "object"
    },
    "FGDCAddressStandard.NumberedThoroughfareAddress_type": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "FGDCAddressStandard.NumberedThoroughfareAddress_type"
                ],
                "type": "string"
            }
        },
        "type": "object"
    },
    "FGDCAddressStandard.TwoNumberAddressRange_type": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "FGDCAddressStandard.TwoNumberAddressRange_type"
                ],
                "type": "string"
            }
        },
        "type": "object"
    },
    "FGDCAddressStandard.USPSGeneralDeliveryOffice_type": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "FGDCAddressStandard.USPSGeneralDeliveryOffice_type"
                ],
                "type": "string"
            }
        },
        "type": "object"
    }
```

```

},
"FGDCAddressStandard.USPSPostalDeliveryBox_type": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "FGDCAddressStandard.USPSPostalDeliveryBox_type"
      ],
      "type": "string"
    }
  },
  "type": "object"
},
"FGDCAddressStandard.USPSPostalDeliveryRoute_type": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "FGDCAddressStandard.USPSPostalDeliveryRoute_type"
      ],
      "type": "string"
    }
  },
  "type": "object"
},
"FGDCAddressStandard.UnnumberedThoroughfareAddress_type": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "FGDCAddressStandard.UnnumberedThoroughfareAddress_type"
      ],
      "type": "string"
    }
  },
  "type": "object"
},
"VRI.AdditionalInfo": {
  "required": [
    "@type",
    "Name"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.AdditionalInfo"
      ],
      "type": "string"
    }
  },
  "FileValue": {
    "oneOf": [
      {
        "type": "string"
      }
    ]
  }
}

```

```

        "$ref": "#/definitions/VRI.File"
    },
    {
        "$ref": "#/definitions/VRI.Image"
    }
]
},
{
    "Name": {
        "type": "string"
    },
    "StringValue": {
        "type": "string"
    }
},
{
    "type": "object"
},
"VRI.ContactMethod": {
    "required": [
        "@type",
        "Type",
        "Value"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.ContactMethod"
            ],
            "type": "string"
        },
        "OtherType": {
            "type": "string"
        },
        "Type": {
            "$ref": "#/definitions/VRI.ContactMethodType"
        },
        "Value": {
            "type": "string"
        }
    },
    "type": "object"
},
"VRI.ElectionAdministration": {
    "required": [
        "@type"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.ElectionAdministration"
            ],
            "type": "string"
        },
        "ContactMethods": {
            "items": {
                "oneOf": [
                    {
                        "$ref": "#/definitions/VRI.ContactMethod"
                    },
                    {
                        "$ref": "#/definitions/VRI.PhoneContactMethod"
                    }
                ]
            }
        }
    }
}

```

```

        }
    ],
    "minItems": 0,
    "type": "array"
},
"Location": {
    "$ref": "#/definitions/VRI.Location"
},
"Name": {
    "type": "string"
},
"Uri": {
    "items": {
        "type": "string",
        "format": "uri"
    },
    "minItems": 0,
    "type": "array"
},
"type": "object"
},
"VRI.ExternalIdentifier": {
    "required": [
        "@type",
        "Type",
        "Value"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.ExternalIdentifier"
            ],
            "type": "string"
        },
        "OtherType": {
            "type": "string"
        },
        "Type": {
            "$ref": "#/definitions/VRI.IdentifierType"
        },
        "Value": {
            "type": "string"
        }
    },
    "type": "object"
},
"VRI.File": {
    "required": [
        "@type",
        "Data"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.File"
            ],
            "type": "string"
        }
    }
}

```

```
"Data": {
    "type": "string",
    "format": "byte"
},
"FileName": {
    "type": "string"
},
"MimeType": {
    "type": "string"
}
},
"type": "object"
},
"VRI.Image": {
    "required": [
        "@type",
        "Data"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.Image"
            ],
            "type": "string"
        },
        "Data": {
            "type": "string",
            "format": "byte"
        },
        "FileName": {
            "type": "string"
        },
        "MimeType": {
            "type": "string"
        }
    },
    "type": "object"
},
"VRI.LatLng": {
    "required": [
        "@type",
        "Latitude",
        "Longitude"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.LatLng"
            ],
            "type": "string"
        },
        "Latitude": {
            "type": "number"
        },
        "Longitude": {
            "type": "number"
        },
        "Source": {
            "type": "string"
        }
    }
}
```

```

        },
        "type": "object"
    },
    "VRI.Location": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "VRI.Location"
                ],
                "type": "string"
            },
            "Address": {
                "$ref": "#/definitions/VRI.Address"
            },
            "Directions": {
                "type": "string"
            },
            "LatLng": {
                "$ref": "#/definitions/VRI.LatLng"
            }
        },
        "type": "object"
    },
    "VRI.Name": {
        "required": [
            "@type"
        ],
        "additionalProperties": false,
        "properties": {
            "@type": {
                "enum": [
                    "VRI.Name"
                ],
                "type": "string"
            },
            "FirstName": {
                "type": "string"
            },
            "FullName": {
                "type": "string"
            },
            "LastName": {
                "type": "string"
            },
            "MiddleName": {
                "items": {
                    "type": "string"
                },
                "minItems": 0,
                "type": "array"
            },
            "Prefix": {
                "type": "string"
            },
            "Suffix": {
                "type": "string"
            }
        },
        "type": "object"
    }
},

```

```

    "type": "object"
},
"VRI.Party": {
  "required": [
    "@type",
    "Name"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.Party"
      ],
      "type": "string"
    },
    "Abbreviation": {
      "type": "string"
    },
    "ExternalIdentifiers": {
      "items": {
        "$ref": "#/definitions/VRI.ExternalIdentifier"
      },
      "minItems": 0,
      "type": "array"
    },
    "Name": {
      "type": "string"
    }
  },
  "type": "object"
},
"VRI.PhoneContactMethod": {
  "required": [
    "@type",
    "Type",
    "Value"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.PhoneContactMethod"
      ],
      "type": "string"
    },
    "Capability": {
      "items": {
        "$ref": "#/definitions/VRI.PhoneCapability"
      },
      "minItems": 0,
      "type": "array"
    },
    "OtherType": {
      "type": "string"
    },
    "Type": {
      "$ref": "#/definitions/VRI.ContactMethodType"
    },
    "Value": {
      "type": "string"
    }
  }
},

```

```

    "type": "object"
},
"VRI.RegistrationAcknowledgement": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.RegistrationAcknowledgement"
      ],
      "type": "string"
    },
    "Signature": {
      "$ref": "#/definitions/Xmldsig.Signature"
    },
    "TransactionId": {
      "type": "string"
    }
  },
  "type": "object"
},
"VRI.RegistrationHelper": {
  "required": [
    "@type",
    "Type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.RegistrationHelper"
      ],
      "type": "string"
    },
    "Address": {
      "$ref": "#/definitions/VRI.Address"
    },
    "Name": {
      "$ref": "#/definitions/VRI.Name"
    },
    "Phone": {
      "$ref": "#/definitions/VRI.PhoneContactMethod"
    },
    "Signature": {
      "$ref": "#/definitions/VRI.Signature"
    },
    "Type": {
      "$ref": "#/definitions/VRI.RegistrationHelperType"
    }
  },
  "type": "object"
},
"VRI.RegistrationProxy": {
  "required": [
    "@type",
    "Type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {

```

```

    "enum": [
      "VRI.RegistrationProxy"
    ],
    "type": "string"
  },
  "Address": {
    "$ref": "#/definitions/VRI.Address"
  },
  "Name": {
    "type": "string"
  },
  "OriginTransactionId": {
    "type": "string"
  },
  "OtherType": {
    "type": "string"
  },
  "Phone": {
    "$ref": "#/definitions/VRI.PhoneContactMethod"
  },
  "TimeStamp": {
    "type": "string",
    "format": "date"
  },
  "Type": {
    "$ref": "#/definitions/VRI.RegistrationProxyType"
  }
},
"type": "object"
},
"VRI.RegistrationRejection": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.RegistrationRejection"
      ],
      "type": "string"
    },
    "AdditionalDetails": {
      "items": {
        "type": "string"
      },
      "minItems": 0,
      "type": "array"
    },
    "Error": {
      "items": {
        "$ref": "#/definitions/VRI.RegistrationError"
      },
      "minItems": 0,
      "type": "array"
    },
    "OtherError": {
      "items": {
        "type": "string"
      },
      "minItems": 0,
      "type": "array"
    }
  }
}

```

```

},
"Signature": {
  "$ref": "#/definitions/Xmldsig.Signature"
},
"TransactionId": {
  "type": "string"
}
},
"type": "object"
},
"VRI.RegistrationSuccess": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.RegistrationSuccess"
      ],
      "type": "string"
    },
    "Action": {
      "items": {
        "$ref": "#/definitions/VRI.SuccessAction"
      },
      "minItems": 0,
      "type": "array"
    },
    "District": {
      "items": {
        "$ref": "#/definitions/VRI.ReportingUnit"
      },
      "minItems": 0,
      "type": "array"
    },
    "EffectiveDate": {
      "type": "string",
      "format": "date"
    },
    "ElectionAdministration": {
      "$ref": "#/definitions/VRI.ElectionAdministration"
    },
    "Locality": {
      "items": {
        "$ref": "#/definitions/VRI.ReportingUnit"
      },
      "minItems": 0,
      "type": "array"
    },
    "PollingPlace": {
      "$ref": "#/definitions/VRI.ReportingUnit"
    },
    "Signature": {
      "$ref": "#/definitions/Xmldsig.Signature"
    },
    "TransactionId": {
      "type": "string"
    }
},
"type": "object"
}
,
```

```

"VRI.ReportingUnit": {
  "required": [
    "@type",
    "Type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.ReportingUnit"
      ],
      "type": "string"
    },
    "ExternalIdentifiers": {
      "items": {
        "$ref": "#/definitions/VRI.ExternalIdentifier"
      },
      "minItems": 0,
      "type": "array"
    },
    "IsDistricted": {
      "type": "boolean"
    },
    "Location": {
      "$ref": "#/definitions/VRI.Location"
    },
    "Name": {
      "type": "string"
    },
    "OtherType": {
      "type": "string"
    },
    "Type": {
      "$ref": "#/definitions/VRI.ReportingUnitType"
    }
  },
  "type": "object"
},
"VRI.Signature": {
  "required": [
    "@type"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.Signature"
      ],
      "type": "string"
    },
    "Date": {
      "type": "string",
      "format": "date"
    },
    "FileValue": {
      "$ref": "#/definitions/VRI.Image"
    },
    "OtherSource": {
      "type": "string"
    },
    "OtherType": {
      "type": "string"
    }
  }
}

```

```

        },
        "Source": {
            "$ref": "#/definitions/VRI.SignatureSource"
        },
        "Type": {
            "$ref": "#/definitions/VRI.SignatureType"
        }
    },
    "type": "object"
},
"VRI.VoterClassification": {
    "required": [
        "@type",
        "Assertion",
        "Type"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.VoterClassification"
            ],
            "type": "string"
        },
        "Assertion": {
            "$ref": "#/definitions/VRI.AssertionValue"
        },
        "OtherType": {
            "type": "string"
        },
        "Type": {
            "$ref": "#/definitions/VRI.VoterClassificationType"
        }
    },
    "type": "object"
},
"VRI.VoterId": {
    "required": [
        "@type",
        "Type"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.VoterId"
            ],
            "type": "string"
        },
        "AttestNoSuchId": {
            "type": "boolean"
        },
        "DateOfIssuance": {
            "type": "string",
            "format": "date"
        },
        "FileValue": {
            "oneOf": [
                {
                    "$ref": "#/definitions/VRI.File"
                },
                {

```

```

        "$ref": "#/definitions/VRI.Image"
    }
],
},
"OtherType": {
    "type": "string"
},
"StringValue": {
    "type": "string"
},
>Type": {
    "$ref": "#/definitions/VRI.VoterIdType"
}
},
"type": "object"
},
"VRI.VoterRecordsRequest": {
    "required": [
        "@type",
        "GeneratedDate",
        "Type",
        "VoterRegistration"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.VoterRecordsRequest"
            ],
            "type": "string"
        },
        "GeneratedDate": {
            "type": "string",
            "format": "date"
        },
        "Issuer": {
            "type": "string"
        },
        "OtherType": {
            "type": "string"
        },
        "Signature": {
            "$ref": "#/definitions/Xmldsig.Signature"
        },
        "TransactionId": {
            "type": "string"
        },
        "Type": {
            "items": {
                "$ref": "#/definitions/VRI.RegistrationRequestType"
            },
            "minItems": 1,
            "type": "array"
        },
        "VendorApplicationId": {
            "type": "string"
        },
        "VoterRegistration": {
            "$ref": "#/definitions/VRI.VoterRegistration"
        }
    },
    "type": "object"
}

```

```

},
"VRI.VoterRegistration": {
  "required": [
    "@type",
    "RegistrationAddress",
    "RegistrationMethod",
    "Name"
  ],
  "additionalProperties": false,
  "properties": {
    "@type": {
      "enum": [
        "VRI.VoterRegistration"
      ],
      "type": "string"
    },
    "AdditionalInfos": {
      "items": {
        "$ref": "#/definitions/VRI.AdditionalInfo"
      },
      "minItems": 0,
      "type": "array"
    },
    "BallotReceiptPreference": {
      "items": {
        "$ref": "#/definitions/VRI.BallotReceiptMethod"
      },
      "minItems": 0,
      "type": "array"
    },
    "ContactMethods": {
      "items": {
        "oneOf": [
          {
            "$ref": "#/definitions/VRI.ContactMethod"
          },
          {
            "$ref": "#/definitions/VRI.PhoneContactMethod"
          }
        ]
      },
      "minItems": 0,
      "type": "array"
    },
    "DateOfBirth": {
      "type": "string",
      "format": "date"
    },
    "Ethnicity": {
      "type": "string"
    },
    "Gender": {
      "type": "string"
    },
    "LastDateOfUSResidency": {
      "type": "string",
      "format": "date"
    },
    "MailForwardingAddress": {
      "$ref": "#/definitions/VRI.Address"
    },
    "MailingAddress": {

```

```

        "$ref": "#/definitions/VRI.Address"
    },
    "Name": {
        "$ref": "#/definitions/VRI.Name"
    },
    "OtherRegistrationForm": {
        "type": "string"
    },
    "OtherRegistrationMethod": {
        "type": "string"
    },
    "OverseasEmployer": {
        "type": "string"
    },
    "Party": {
        "$ref": "#/definitions/VRI.Party"
    },
    "PreviousName": {
        "$ref": "#/definitions/VRI.Name"
    },
    "PreviousRegistrationAddress": {
        "$ref": "#/definitions/VRI.Address"
    },
    "PreviousSignature": {
        "$ref": "#/definitions/VRI.Signature"
    },
    "RegistrationAddress": {
        "$ref": "#/definitions/VRI.Address"
    },
    "RegistrationAddressIsMailingAddress": {
        "type": "boolean"
    },
    "RegistrationForm": {
        "$ref": "#/definitions/VRI.RegistrationForm"
    },
    "RegistrationHelpers": {
        "items": {
            "$ref": "#/definitions/VRI.RegistrationHelper"
        },
        "minItems": 0,
        "type": "array"
    },
    "RegistrationMethod": {
        "$ref": "#/definitions/VRI.RegistrationMethod"
    },
    "RegistrationProxy": {
        "$ref": "#/definitions/VRI.RegistrationProxy"
    },
    "SelectedLanguage": {
        "type": "string"
    },
    "Signature": {
        "$ref": "#/definitions/VRI.Signature"
    },
    "VoterClassifications": {
        "items": {
            "$ref": "#/definitions/VRI.VoterClassification"
        },
        "minItems": 0,
        "type": "array"
    },
    "VoterIds": {

```

```

    "items": {
      "$ref": "#/definitions/VRI.VoterId"
    },
    "minItems": 0,
    "type": "array"
  },
  "type": "object"
},
"VRI.AssertionValue": {
  "enum": [
    "no",
    "unknown",
    "yes"
  ],
  "type": "string"
},
"VRI.BallotReceiptMethod": {
  "enum": [
    "email",
    "email-or-online",
    "fax",
    "mail",
    "online"
  ],
  "type": "string"
},
"VRI.ContactMethodType": {
  "enum": [
    "email",
    "other",
    "phone"
  ],
  "type": "string"
},
"VRI.IdentifierType": {
  "enum": [
    "fips",
    "local-level",
    "national-level",
    "ocd-id",
    "other",
    "state-level"
  ],
  "type": "string"
},
"VRI.PhoneCapability": {
  "enum": [
    "fax",
    "mms",
    "sms",
    "voice"
  ],
  "type": "string"
},
"VRI.RegistrationError": {
  "enum": [
    "identity-lookup-failed",
    "incomplete",
    "incomplete-address",
    "incomplete-birth-date",
    "incomplete-name",
  ]
}

```

```

    "incomplete-signature",
    "ineligible",
    "invalid-form",
    "other"
],
"type": "string"
},
"VRI.RegistrationForm": {
  "enum": [
    "fPCA",
    "NVRA",
    "other"
  ],
  "type": "string"
},
"VRI.RegistrationHelperType": {
  "enum": [
    "assistant",
    "witness"
  ],
  "type": "string"
},
"VRI.RegistrationMethod": {
  "enum": [
    "armed-forces-recruitment-office",
    "motor-vehicle-office",
    "other",
    "other-agency-designated-by-state",
    "public-assistance-office",
    "registration-drive-from-advocacy-group-or-political-party",
    "state-funded-agency-serving-persons-with-disabilities",
    "unknown",
    "voter-via-election-registrars-office",
    "voter-via-email",
    "voter-via-fax",
    "voter-via-internet",
    "voter-via-mail"
  ],
  "type": "string"
},
"VRI.RegistrationProxyType": {
  "enum": [
    "armed-forces-recruitment-office",
    "motor-vehicle-office",
    "other",
    "other-agency-designated-by-state",
    "public-assistance-office",
    "registration-drive-from-advocacy-group-or-political-party",
    "state-funded-agency-serving-persons-with-disabilities"
  ],
  "type": "string"
},
"VRI.RegistrationRequestType": {
  "enum": [
    "ballot-request",
    "other",
    "registration"
  ],
  "type": "string"
},
"VRI.ReportingUnitType": {
  "enum": [

```

```

    "ballot-batch",
    "ballot-style-area",
    "borough",
    "city",
    "city-council",
    "combined-precinct",
    "congressional",
    "county",
    "county-council",
    "drop-box",
    "judicial",
    "municipality",
    "other",
    "polling-place",
    "precinct",
    "school",
    "special",
    "split-precinct",
    "state",
    "state-house",
    "state-senate",
    "town",
    "township",
    "utility",
    "village",
    "vote-center",
    "ward",
    "water"
  ],
  "type": "string"
},
"VRI.SignatureSource": {
  "enum": [
    "dmv",
    "local",
    "other",
    "state",
    "voter"
  ],
  "type": "string"
},
"VRI.SignatureType": {
  "enum": [
    "dynamic",
    "electronic",
    "other"
  ],
  "type": "string"
},
"VRI.SuccessAction": {
  "enum": [
    "address-updated",
    "name-updated",
    "other",
    "registration-cancelled",
    "registration-created",
    "registration-updated",
    "status-updated"
  ],
  "type": "string"
},
"VRI.VoterClassificationType": {

```

```

"enum": [
    "activated-national-guard",
    "active-duty",
    "active-duty-spouse-or-dependent",
    "citizen-abroad-intent-to-return",
    "citizen-abroad-never-resided",
    "citizen-abroad-return-uncertain",
    "deceased",
    "declared-incompetent",
    "eighteen-on-election-day",
    "felon",
    "other",
    "permanently-denied",
    "protected-voter",
    "restored-felon",
    "united-states-citizen"
],
"type": "string"
},
"VRI.VoterIdType": {
    "enum": [
        "drivers-license",
        "local-voter-registration-id",
        "other",
        "ssn",
        "ssn4",
        "state-id",
        "state-voter-registration-id",
        "unknown",
        "unspecified-document",
        "unspecified-document-with-name-and-address",
        "unspecified-document-with-photo-identification"
    ],
    "type": "string"
},
"VRI.Address": {
    "required": [
        "@type"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "VRI.Address"
            ],
            "type": "string"
        }
    },
    "type": "object"
},
"Xmldsig.Signature": {
    "required": [
        "@type"
    ],
    "additionalProperties": false,
    "properties": {
        "@type": {
            "enum": [
                "Xmldsig.Signature"
            ],
            "type": "string"
        }
    }
}

```

```
        },
        "type": "object"
    },
},
"oneOf": [
{
    "$ref": "#/definitions/VRI.VoterRecordsRequest"
},
{
    "$ref": "#/definitions/VRI.VoterRecordsResponse"
}
]
```