CATALYST
CONSULTING GROUP, INC.

Deliverable: 2.5 VoteCal System County EMS System Integration and Data Exchange Specifications Document

VoteCal Statewide Voter Registration System Project

State of California, Secretary of State (SOS)

May 19, 2010
Version: 1.0

Authors

This document was prepared by:

Contributors		
Kenneth Miller Catalyst Consulting Group 211 West Wacker Drive Suite 450 Chicago, IL 60606	Jay Varner Catalyst Consulting Group 211 West Wacker Drive Suite 450 Chicago, IL 60606	Scott Hilkert Catalyst Consulting Group 211 West Wacker Drive Suite 450 Chicago, IL 60606
Victor Vergara Catalyst Consulting Group 211 West Wacker Drive Suite 450 Chicago, IL 60606	Kurt Schwartz Catalyst Consulting Group 211 West Wacker Drive Suite 450 Chicago, IL 60606	

Date	Document Version	Document Revision Description	Revision Author
02/23/2010	0.09	Initial Draft	Catalyst Team
03/29/2010	0.10	Pre-draft submission to SOS	Catalyst Team
03/30/2010	0.11	Draft revisions and QC	Catalyst Team
03/31/2010	0.12 and 0.13	Draft revisions	Catalyst Team
04/01/2010	0.14	Draft revisions	Catalyst Team
04/01/2010	0.15	Draft revisions and QC	Catalyst Team
04/02/2010	0.16	Draft revisions and QC	Catalyst Team
04/02/2010	0.17	Pre-draft submission to SOS	Catalyst Team
04/13/2010	0.18 and 0.19	Expanded Content on API’s	Catalyst Team
04/16/2010	0.20	Incorporation of vendor feedback	Catalyst Team
04/23/2010	0.21	Further vendor feedback revisions	Catalyst Team
05/03/2010	0.22	Restructure	Catalyst Team
05/06/2010	0.23	Ensure compatibility with DED ver. 2.0	Catalyst Team
05/18/2010	0.24	Conduct Peer Review	Chad Hoffman
05/19/2010	1.0	Conduct QA and Release to Client for Review	Don Westfall

Table of Contents

1.0 Introduction

The VoteCal project will create a statewide voter registration database and management tool to satisfy federally enacted HAVA requirements. The approach taken in California is to develop what is known as a "bottom-up" system that integrates a state centralized database component with the election management systems (EMS) that are already installed at the 58 counties in California. To function as part of the VoteCal system, these EMS applications require part of their functionality to be remediated. In general, the remediated EMS systems must:

- Provide new or modified user interfaces and business logic to accommodate prescribed VoteCal business processes.
- Include new custom database views and tables in the county EMS database which will be used by VoteCal system components to exchange batch election management information.
- Maintain a live connection to VoteCal though application program interface commands in order to keep voter data in the local EMS database validated and synchronized with official statewide voter data.
- Be configured to translate internal codes and other data items to VoteCal prescribed standard values.

Several EMS vendors serve their existing California customers with proprietary software products for which source code and intellectual property are owned by the vendor. The California Secretary of State (SOS) is contracting with each of these EMS vendors to make modifications to their base EMS product for compatibility with VoteCal and to deploy and configure the modified products for their existing (and future) customer-base in California.

This document, which serves as a technical and functional specification for the EMS remediation process, is a supporting document of the SOW to be provided by SOS.

This specification includes five major components:

- ***Use Cases:*** These describe the specific business processes and system functions that the remediated EMS is required to provide in the context of the overall VoteCal solution. These use cases are found in external attachments.
- ***Database View Definitions:*** These specify database views which must be configured in the EMS main database which will allow direct extraction of batch data by the VoteCal Data Exchange Service which is software provided by SOS for deployment within the county network environment. Specific data validations are provided in the body of this deliverable and standard code values are provided in the appendix.
- ***Database Table Definitions:*** These specify database tables which must be configured in the EMS main database which will receive insertion of batch data from the VoteCal Data Exchange Service.
- ***Application Program Interface Specifications:*** These specify a collection of VoteCal Web services which the remediated EMS will use to synchronize voter registration data and implement the business processes set forth in the use cases.
- Definitions of standard code and field values.

In addition to these five components, this specification document also provides information useful to the EMS vendors in the execution of their remediation process.

1.1 Purpose and Objectives

This VoteCal System County EMS System Integration and Data Exchange Specifications Document serves as a technical specification and resource for EMS vendors to modify or remediate their EMS product for compatibility and integration with VoteCal.

This document is *Deliverable 2.5 VoteCal System County EMS System Integration and Data Exchange Specification Document*. It has been developed to the specifications presented in Deliverable Expectation Document *(DED) 2.5, VoteCal System County EMS System Integration and Data Exchange Specification Document*.

1.2 Scope

This VoteCal System County EMS System Integration and Data Exchange Specifications Document identifies the information necessary so the independent counties and their vendors can identify the required modifications to their EMS for integration with the VoteCal System solution.

1.3 Standards

A large portion of this technical specification covers the Application Program Interface (API) which defines how the county EMS software will communicate with the VoteCal software at the function and data type level. The technical API specification including underlying XML schemas is presented in Web Services Description Language (WSDL) 1.1. Accompanying descriptions of the APIs are provided in Microsoft Compiled HTML Help format (CHM). These are provided as external file attachments. These forms of documentation will be familiar to software developers.

No other standards apply to this document except that, where applicable, existing CalVoter data standards have been used since these are already familiar to the EMS vendors who will use this document.

1.4 Assumptions, Dependencies, and Constraints

This deliverable was developed based on the following assumptions, dependencies, and constraints:

- ❖ This deliverable is created as a technical specification that documents the system interfaces between VoteCal System and the county EMS systems.
- ❖ SOS will create the Statement of Work (SOW) that will be provided to EMS vendors. This deliverable will be presented with the SOW as supporting information.
- ❖ This deliverable is dependent upon the following attachments
    - VoteCal EMS Integration API WSDL version 2.2
    - VoteCal EMS Integration API Reference (chm) version 2.2

- All relevant use cases listed within this deliverable

1.5 Document Control

This document contains a revision history log. When changes occur, the version number will be incremented and the date, name of the person authoring the change, and a description of the change will be recorded in the revision history log of the document.

The due date for the final version of this deliverable is June 2, 2010. Development of the VoteCal application and the EMS remediation will not begin until this deliverable has been submitted and approved. As development progresses, issues may arise requiring revisions or updates to this document. Should this need arise, updates will be scheduled to facilitate incorporation of all relevant information. Events that may drive the need for updates may include:

- Limitations in the EMS system that prevent or preclude interactions as described in this document
- Discovery of additional values maintained at the local level that require expansion or revision to standard code values listed in Appendix A
- System or data compatibility issues
- Enhancements in tables or views to meet existing requirements.

The Web Services Description Language file will be managed as a separate external .WSDL file attachment. Its version number will be referenced within the appropriate section of this document.

The Microsoft Compiled HTML Help content will be managed as a separate external .ZIP file attachment containing a single .CHM file and numerous supporting files. Its version number will be referenced within the appropriate section of this document.

As with other work products of the VoteCal Project, the approved VoteCal System County EMS System Integration and Data Exchange Specification will be placed under configuration management in accordance with the Document Management Plan (a subset of the Project Management Plan). Also, in accordance with the Document Management Plan, the VoteCal System County EMS System Integration and Data Exchange Specification will be stored on the SharePoint server and available to the project team, the Independent Project Oversight Consultant (IPOC), Independent Verification and Validation (IV&V) vendor, and SOS senior management.

2.0 County EMS System Integration and Data Exchange Specifications

This section begins with a description of the remediation process that EMS vendors will participate in, the key milestone dates, and a breakdown of roles and responsibilities.

Following the remediation process description, this section presents detailed requirements and technical specifications.

2.1 County EMS Remediation

The EMS vendors will read and understand this technical specification in the context of a separate statement of work (SOW) document to be provided by SOS. This SOW will call for the remediation of the vendors' existing EMS product including software development services, testing, participation in a pilot program with select counties, data conversion services, training of their California customer base, and deployment at county sites.

Most of the topics in this section will also be covered in the separate SOW document to be provided by SOS. This information is provided as a reference for understanding the remediation specification in the context of the overall process.

2.1.1 Processes

The remediation process is broken into the following activities. Key dates are provided in section *2.1.2 Key Milestones*

2.1.1.1 Communication

Vendors will be provided with the name of a primary technical contact and several alternates from within the VoteCal team. The vendor will provide a primary technical point of contact reachable by Catalyst. This primary contact may designate additional contacts to receive information relating to the project.

Catalyst will provide a secure file sharing server (either SharePoint or FTP to be determined) and accounts for each vendor to facilitate the dissemination of information.

Regular conference calls and in-person meetings at SOS offices will be scheduled to facilitate the development and integration testing process and to ensure that the workflows required by the API can be supported by each EMS vendor (schedule to follow). Some of the conference calls and in-person meetings will be held with all EMS vendors present to gain statewide consensus on Web Service API and VoteCal Data Batch Service integration topics, while others will be scheduled with individual EMS vendors for routine check-ins and detailed technical analysis of issues pertaining to a vendor's particular EMS application.

2.1.2 Key Milestones

This section lists the key milestones excerpted from the VoteCal Integrated Schedule version 3.46. These tasks have in many cases been renamed from the EMS vendors' perspective and from a milestone date perspective. The WBS column provides a cross reference to the related task in the VoteCal Integrated Schedule.

Tasks are shown in chronological order which generally also relates to the sequence of dependencies. Detailed information on dependencies can be found in the VoteCal Integrated Schedule.

Task Name	End Date	WBS
Provide Review Comments on Draft Deliverable 2.5 EMS Integration and Data Exchange Specifications	4/13/2010	3.14.4
Release SOW for EMS Remediation Contracts	5/4/2010	2.8.2
Complete Deliverable 2.5 EMS Integration and Data Exchange Specifications.	6/2/2010	3.14.8
Complete Deliverable 2.9 Data Conversion and Data Integration Plan	6/2/2010	3.15.8
Receive Bids from EMS Vendors	6/2/2010	2.8.3
Finalize EMS Remediation Contracts, Vendor Start of Work	6/16/2010	2.8.4
EMS Vendor Sandbox Environment Available to Vendors	9/8/2010	4.3.4.1.4
Complete Selection of Pilot Counties	9/16/2010	5.1.10
Submit Draft Deliverable 3.3 Acceptance Test Plan for EMS Conversion	9/21/2010	4.4.4
Provide Review Comments on Deliverable 3.3 Acceptance Test Plan for EMS Conversion	10/5/2010	4.4.6
Complete Deliverable 3.3 Acceptance Test Plan for EMS Conversion	10/21/2010	4.4.10
Completion of Design, build, and vendor self testing of EMS	11/5/2010	2.8.5
Deploy EMS to Pilot County Test Environment	11/5/2010	2.8.6
Confirm EMS Readiness to Test, Start Tests of Remediated EMS Systems	12/2/2010	5.7.1
Complete Tests of Remediated EMS Systems (simulated live election cycle)	1/4/2011	5.7.2
Provide Certification of Remediated Systems	1/11/2011	5.7.3
Complete Training of Pilot County Election Staff on EMS Systems	2/16/2011	6.3.6
Complete VoteCal User Acceptance Test	4/1/2011	5.3.7
Promote EMS systems from Test to Production Environment	4/18/2011	6.4.3
Rerun Data Conversion to Production Environment, Start Pilot Tests	4/19/2011	6.4.4
Complete Pilot Testing Through Live Election Cycle	6/9/2011	6.4.5
Approve Final Pilot Results Report	7/12/2011	6.4.13
Complete Correction of EMS Software & System Defects Discovered in Pilot Test	7/22/2011	6.5.1

Task Name	End Date	WBS
Update and Finalize Training Materials	7/29/2011	7.4.1
Complete SOS Remediation-Acceptance Test on Changes after Pilot	8/5/2011	6.5.3
Verify Deployment Checklist for Counties, Start Deployment	8/10/2011	7.5.1
Complete Final Regional Training Session	10/14/2011	7.4.6
Complete EMS Deployment and JIT Training for Counties	11/3/2011	7.9.2

2.1.3 Roles and Responsibilities

The following table shows the breakdown of roles and responsibilities across the main process categories between Catalyst, SOS, EMS vendors, and counties.

Process	Catalyst Role	SOS Role	EMS Vendor Role	County Role
Communication	Maintain File Repository Initiate and Host Conference Calls and Meetings Answer calls for assistance and questions	Participate in Meetings Initiate Communication when issues arise	Participate in Meetings Initiate Communication when issues arise	
Initial EMS Software Development	Provide WSDL Files and assist vendors with import Provide Sample Client application Monitor Progress and provide assistance Develop VoteCal sandbox environment in parallel	Monitor Progress	Actual Software Remediation Work in own Development Environment.	
Continued EMS Software Development using Sandbox	Deploy and Maintain Sandbox Environment, Train EMS Vendors on its use. Monitor Progress and provide assistance	Monitor Progress Provide and Maintain Network Access to Sandbox Environment.	Continued Software Remediation Work. Self Testing using Sandbox Environment	

Process	Catalyst Role	SOS Role	EMS Vendor Role	County Role
Remediated EMS Testing	Write test procedures as part of Deliverable 3.3 Configure Test Environment including VoteCal components Perform System Tests on Remediated EMS Monitor Tests and Document Results Issue Certification Report	Approval of Deliverable 3.3 Monitor Progress Approve Certification Report	Review and provide feedback on Deliverable 3.3 Deploy Remediated EMS in County Test Environment. Assist in deployment of VoteCal components in County Test Environment. Assist in Performance of System Tests on Remediated EMS	Provide County Test Environment Configure and Maintain Local Area Network for connectivity between EMS and VoteCal
User Acceptance Testing	Assist SOS in its responsibilities Configure UAT Environment including VoteCal components	Write UAT procedures. Perform UAT on Remediated EMS	Assist in Performance of UAT on Remediated EMS	Maintain Local Area Network Perform Remediation Testing on non-VoteCal related features of EMS
Pilot Testing	Advise selection of pilot counties Assist in Migration of EMS Components to County Production Environment Configure VoteCal Production Environment including VoteCal components Provide User Help Desk support Monitor and Report on Results	Select Pilot Counties. Administer Pilot County Contracts Approve Report Provide and Maintain Network Access to Production Environment.	Advise selection of pilot counties Migrate Remediated EMS to County Production Environment. Provide User Help Desk support	Serve as Pilot County Use the system through a Live Election Cycle Maintain Local Area Network

2.2 System Configuration and Modification Specifications

This section presents several core requirements for the remediated EMS software and the supporting technical details. Each core requirement introduces a section of detailed specifications that comprise the overall set of remediated EMS requirements. The summarized EMS vendor core requirements are as follows:

- Implement functionality in the remediated EMS system to fulfill each of the use cases that are provided as attachments to this specification document.
- Develop and implement new database views in the underlying database of their remediated EMS systems from which data will be extracted.
- Provide functionality that allows for configurable translations of local EMS database fields to the VoteCal standardized code values (please see appendix for code values).
- Develop and implement new database tables in the underlying database of their remediated EMS systems into which data will be inserted.
- Implement functionality in the remediated EMS system to interact with VoteCal through the specified set of Web Service APIs.

The first three sub-sections below describe the key specification artifacts:

- Required Use Cases
- Required Database Views and Tables
- Required use of Web Service APIs

The remaining sub-sections reference additional relevant information for the EMS remediation process.

2.2.1 Use Cases

Core Requirement: The EMS vendors are required to implement functionality in the remediated EMS system to fulfill each of the use cases that are provided as attachments to this specification document. The vendor is given latitude in design of user interfaces and internal business logic as long as the use case is covered. The vendor is encouraged to make design decisions that will allow the overall user experience to remain as familiar as possible for their customers.

The following list is an inventory of use cases that must be implemented or supported by the remediated EMS software. The inventory list references the specific version of the use case that is associated with the current version of this specification document.

Use Case #	Use Case Name	Version
01.03.01	Record Voter Registration Information through EMS	1.7
01.11.01	Initiate Voter Registration and Run Local EMS Search	1.6
01.11.02	Transfer Voter through EMS	1.5
01.11.04	Enter Declined Voter through EMS	1.4
01.13.01	Check for Duplicate Driver License or State ID Rejections	1.4
01.14.01	Respond to Notice of Data Deficiency	1.5

Use Case #	Use Case Name	Version
01.19.01	Handle Local Voter Registration Contingency	1.5
01.21.01	Run Pre-Registration Check	1.1
01.22.01	Assign Precinct through EMS	1.1
01.23.01	Process Unprecinctable Work Item through EMS	1.1
01.24.01	State Update Record (Register, Update, Transfer)	1.1
01.26.01	Process VoteCal Validations and Store VR Information	1.1
02.02.01	Search Voter through EMS Interface	1.8
02.04.01	Retrieve Voter Details through Web Service	1.3
03.01.01	Process CDPH Death Records File	1.6
03.02.02	Accept or Reject Death Record Match Case through EMS	1.3
03.03.02	Undo Death Record Match Case through EMS	1.3
03.11.01	Process CDCR Felon Record File	1.6
03.12.02	Accept or Reject CDCR Felon Record Match Case through EMS	1.3
03.13.02	Undo CDCR Felon Match Case through EMS	1.3
03.21.01	Process DMV COA Record File	1.9
03.22.01	Process DMV COA Work Item through EMS	1.10
03.32.01	Process NCOA Record File	1.11
03.33.01	Process NCOA Work Item through EMS	1.8
03.41.01	Process Duplicate Voter Detection Job	1.7
03.42.02	Accept or Reject Duplicate Voter Match Case through EMS	1.1
03.43.02	Undo Accepted Duplicate Voter Match Case through EMS	1.1
03.50.01	Undo Transfer Out through EMS	1.7
03.51.02	Review Unresolved List Maintenance Issues through EMS	1.1
03.56.01	Process CASS Record File	1.1
04.06.01	EMS Processes Message Queue	1.5
04.10.01	Process Election Participation Batch	1.7
04.12.01	Process Provisional Voting Batch	1.7
04.13.01	Process Precinct-District Mapping Batch	1.7
04.14.01	Process Voter Record Update Batch	1.6
04.15.01	Process Election Polling Place Batch	1.6
04.16.01	Process Voter Activity Batch	1.7
04.17.01	Send Image Batch	1.5
04.18.01	Process Batch Data Exchange	1.3
04.19.01	Create or Update Vote-by-Mail Record	1.3
04.20.01	Process DMV Signature Request	1.1
04.21.01	Compare County and VoteCal Voter Data	1.3

Use Case #	Use Case Name	Version
04.22.01	Generate Voter Data Comparison Extract	1.0
06.01.01	Generate Election Roster	1.5
07.18.01	Generate Report or Correspondence	1.9
05.08.01	Record Issuance of Affidavits for Organization	1.6
05.09.01	Determine Organization that was Issued an Affidavit	1.6
05.26.02	Edit Affidavit Issuance Record	1.1

2.2.2 Data

In order to communicate effectively with VoteCal, the local EMS systems must conform to specific data standards and code translations as defined in the following sections. Data will be transferred between the two systems both in real-time through VoteCal API's, and through batch processes managed through the Data Exchange Service installed at the local level.

2.2.2.1 Transferred from EMS to VoteCal in Batch (Database Views)

Core Requirement: The EMS vendors must develop and implement new database views in the underlying database of their remediated EMS systems from which data will be extracted.

Core Requirement: The EMS vendor must provide functionality that allows for configurable translations of local EMS database fields to the VoteCal standardized code values listed in Appendix A.

The following views within the local county EMS database will be accessed by the VoteCal Data Exchange Service running on a server at the county that has access to the local county EMS database. The vendor of the local county system is required to develop these specific database views and ensure all data elements and naming conventions conform to VoteCal standards. All of this information will be accessed using a read-only connection.

The VoteCal Data Exchange Service will connect directly to county EMS databases using ADO.NET. In the unlikely event that a specific ADO.NET adapter is not available for the database software used by the county EMS application, a generic ADO.NET ODBC connection will be used. The EMS vendors must provide non-expiring database credentials for the VoteCal Data Exchange Service installed at each county with read/write access to specific new tables and views specified in this section.

When the Notes column of a particular Field in one of the database view definitions below references one of tables of standardized codes (e.g. County Code, Registration Source Code, Registration Method Code, etc.) it is the responsibility of the EMS vendor to translate their internal code values to the VoteCal standardized code values during deployment of the remediated EMS. The remediated EMS software must also include a mechanism allowing future modifications to the translated code values by a county system administrator. The appendix provides the present known standardized code values. These tables will be updated through collaboration with the EMS vendors to leverage existing code definitions already in use in the state.

The Required column indicates whether or not the associated field is a required value in the VoteCal system. In cases where a field is flagged as not required ('N' in the required field) the field should be left null if no values are available, but not eliminated completely from the data set.

VoteCal will truncate field values that exceed maximum field length (Max Len column) and flag the condition as a data deficiency. Maximum field lengths indicated may be subject to revision in later versions of this document as feedback is gathered from the EMS vendors.

All view fields will be represented as unicode strings. The column titled Formatting indicates the data type to which the string will be converted in VoteCal. Views must be constricted so that their contents are convertible to these types.

2.2.2.1.1 Election Definition View

This view provides definition of basic election information. On the local EMS system, this view should be named VOTECAL_ELECTION_DEFINITION.

Table 2-1 Election Definition View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
County_Election_Id	Y	Alphanumeric	10	This field will be used as a compound key with the Jurisdiction ID to provide a unique election identity within VoteCal
Election_Date	Y	Date	10	The date this election is to be held in MM-DD-YYYY format.
Election_Title_1	Y	Alphanumeric	60	The title assigned to this election by the local jurisdiction.
Election_Title_2	N	Alphanumeric	60	
Election_Type	Y	Alphanumeric	2	Per Table A-7, the type of election (Primary, General, etc)

2.2.2.1.2 Election Participation View

This view provides detail on a voter's eligibility for and participation in an election. On the local EMS system, this view should be named VOTECAL_ELECTION_PARTICIPATION.

Table 2-2 Election Participation View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
County_Election_Id	Y	Alphanumeric	10	Primary key from the local EMS Elections table
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from the local EMS Voter table
State_Voter_Id_Primary_Key	Y	Guid		VoteCal primary key associated to this voter
Voting_Method	Y	Alphanumeric	2	Per Table A-14, the method used to cast this ballot (VBM, Poll, etc)
Party	N	Alphanumeric	3	Per Table A-6, the party designated by this voter for partisan elections.
Participated	Y	Alphanumeric	1	‘Y’ indicates that this voter voted, ‘N’ indicates that the voter was eligible but did not participate.
Consolidation_Abbreviation	Y	Alphanumeric	12	The precinct consolidation assigned to this voter for this election. Values provided must match values in the PRECINCT CONSOLIDATION VIEW for this election.

2.2.2.1.3 Vote-by-Mail View

This view provides detail on Vote-by-Mail (VBM) applications for a specific election. On the local EMS system, this view should be named VOTECAL_VOTE_BY_MAIL.

Table 2-3 Vote-by-Mail View

Field	Required?	Formatting	Max Length	Notes
VBM_Id_Primary_Key	Y	Alphanumeric	10	Primary key from local EMS system for this VBM request. This may be the primary key from a VBM table, a local voter ID, or any meaningful link to this specific record.
State_Voter_Id_Primary_Key	Y	Guid	16	VoteCal primary key associated to this voter
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from local EMS Voter table
County_Election_Id	Y	Alphanumeric	10	Primary key designating this election within the local EMS Elections table
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
VBM_Category	Y	Alphanumeric	2	Per Table A-16, the VBM category (Permanent, UOCAVA, etc)
Mail_Address_Line_1	N	Alphanumeric	60	If this registrant provided a mailing address specific to this application.
Mail_Address_Line_2	N	Alphanumeric	60	
Mail_Address_Line_3	N	Alphanumeric	60	
Mail_Address_Line_4	N	Alphanumeric	60	
Is_Foreign_Address	N	Alphanumeric	1	'Y' if this is a non-domestic mailing address, else 'Y'
Application_Date	Y	Date	10	Date this application was initiated in MM-DD-YYYY format.
Application_Method	Y	Alphanumeric	2	Per Table A-17, the method of application (mail, in-person, etc)

Field	Required?	Formatting	Max Length	Notes
Application_Date_Rejected	N	Date	10	If application was rejected, the date of that rejection in MM-DD-YYYY format. If accepted, this field should be null.
Application_Reason_Rejected	N	Alphanumeric	3	Per Table A-18, the reason this application was rejected.
Date_Transmitted	N	Date	10	Date the ballot was sent to the voter in MM-DD-YYYY format
Method_Transmitted	N	Alphanumeric	1	Per Table A-19, the method used to send this ballot.
Date_Signed	N	Date	10	Date the ballot was signed in MM-DD-YYYY format.
Date_Received	N	Date	10	Date the ballot was received at the local jurisdiction in MM-DD-YYYY format.
Return_Method	N	Alphanumeric	1	Per Table A-20, the method used to return this ballot (Mail, In-Person, etc)
Voting_Method	N	Alphanumeric	1	Per Table A-21
Rejection_Reason	N	Alphanumeric	2	Per Table A-22, the reason this ballot was rejected upon receipt and review (Not Signed, etc)
Rejection_Date	N	Date	10	If rejected, the date this ballot was rejected.

2.2.2.1.4 District Detail View

This view will provide details on districts. On the local EMS system, this view should be named VOTECAL_DISTRICT_DETAIL.

Table 2-4 District Detail View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.

Field Name	Required?	Formatting	Max Length	Notes
District_Type	Y	Alphanumeric	3	Per Table A-23, the type of district (Congressional, Senatorial, etc)
State_District_Code	N	Alphanumeric	5	The district code assigned to this district by the State. NOTE: These codes are still under development, and will be provided in later revisions to this document
Name_1	Y	Alphanumeric	50	Full name of precinct or district (Example: Santa Clara Water District)
Name_2	N	Alphanumeric	50	
Abbreviation	Y	Alphanumeric	12	The abbreviated name of this district or precinct. Must be unique within each jurisdiction. This will be used as a compound key with the jurisdiction id to form a unique identifier within VoteCal.

2.2.2.1.5 Precinct Detail View

This view will provide details related to precincts. If the jurisdiction maintains precincts and portions of precincts (e.g., parts or splits), one record should be submitted for the 'whole' and separate records for each portion. On the local EMS system, this view should be named VOTECAL_PRECINCT_DETAIL.

Table 2-5 Precinct Detail View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
Precinct_Number	Y	Alphanumeric	9	The numeric representation for this precinct. Combined with the Precinct Part, must be unique within each jurisdiction.

Field Name	Required?	Formatting	Max Length	Notes
Precinct_Part	Y	Alphanumeric	4	The numeric representation for this precinct portion (part/split). The 'whole' precinct should be designated with a 'part' code of '0000'.
Name	Y	Alphanumeric	35	Full name associated with this precinct
Abbreviation	Y	Alphanumeric	20	The abbreviated name of this precinct. Must be unique within each jurisdiction.

2.2.2.1.6 Precinct-District Mapping View

This view provides details related to district assignments by precinct. One row for each district assigned in a many-to-one relationship with precincts. On the local EMS system, this view should be named VOTECAL_PRECINCT_DISTRICT_MAP.

Table 2-6 Precinct-District Mapping View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
Precinct_Part	Y	Alphanumeric	14	Combined values of Precinct and Part (separated by a period) as defined in the Precinct Detail View. The 'whole' precinct should be designated with a 'part' code of '0000' (example: 99999.0000)
District_Abbreviation	Y	Alphanumeric	12	Conforms to District Abbreviation defined in the Precinct-District Detail View
Effective_Date	Y	Date	10	Date the relation between precinct portion and district became effective in MM-DD-YYYY format.

2.2.2.1.7 Precinct Consolidation View

This view provides details related to precinct consolidations for a specific election. One record for each precinct assigned in a many-to-one relationship with the consolidation. On the local EMS system, this view should be named VOTECAL_PRECINCT_CONSOLIDATION.

Table 2-7 Precinct Consolidation View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
County_Election_Id	Y	Alphanumeric	10	Primary key designating this election within the local EMS Elections table
Consolidation_Name	Y	Alphanumeric	64	A unique name assigned to this consolidation for this election.
Consolidation_Type	Y	Alphanumeric	3	VBM/REG
Consolidation_Abbreviation	Y	Alphanumeric	12	A unique abbreviation of this Consolidation’s assigned name.
Precinct_Part	Y	Alphanumeric	14	Combined values of Precinct and Part (separated by a period) being consolidated as defined in the Precinct Detail View.
Poll_Code	N	Alphanumeric	10	A unique identifier for the assigned polling place consistent with Poll_Code from the Election Poll View.

2.2.2.1.8 Election Poll View

This view provides details related to the pool of available polling places. On the local EMS system, this view should be named VOTECAL_ELECTION_POLL.

Table 2-8 Election Poll View

Field Name	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
Poll_Code	Y	Alphanumeric	10	The abbreviated name or code assigned to this polling place. Must be unique within each jurisdiction.
Poll_Name_1	Y	Alphanumeric	64	Full name assigned to this polling place.
Poll_Name_2	N	Alphanumeric	64	
Address_Line_1	Y	Alphanumeric	65	Free form address of this polling place.
Address_Line_2	Y	Alphanumeric	65	Free form address of this polling place.
City	Y	Alphanumeric	60	The city associated to the address of this polling place.
State	Y	Alphanumeric	2	The USPS state code associated to the address of this polling place.
Zip	Y	Alphanumeric	5	The five-digit zip code associated to the address of this polling place.
Accessible	Y	Alphanumeric	1	‘Y’ if handicapped accessible, ‘N’ if not

2.2.2.1.9 Document View

This view provides details related document images and associated metadata.
Acceptable image formats are listed in Appendix A, Table A-25. Multi-page TIF files are only allowed for multi-page documents of the same category associated to a single voter record (e.g. a two-page correspondence). On the local EMS system, this view should be named VOTECAL_DOCUMENT.

Table 2-9 Document View

Field Name	Required?	Formatting	Max Length	Notes
State_Voter_Id_Primary_Key	N	Guid	16	VoteCal primary key associated to this voter
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from local EMS Voter table
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
Document_Type	Y	Alphanumeric	1	Per Table A-24, the type of document (Affidavit, Signature, etc)
Image_Format	Y	Alphanumeric	1	Per Table A-25, the format of this image (TIF, JPG, etc). **SPECIAL NOTE: Multi-page TIF files will only be accepted for individual multi-page documents (e.g. a two-page correspondence to an individual registrant).**
Document_Date	Y	Datetime		For historic collections of documents such as affidavits, this date should clearly reflect the sequence of submission in MM-DD-YYYY HH:MM:SS format.
Image	Y	Binary		The binary representation of the image.

2.2.2.1.10Voter View

This view provides all pertinent details related to a voter's registration record. On the local EMS system, this view should be named VOTECAL_VOTER.

Table 2-10 Voter View

Field	Required?	Formatting	Max Len	Notes
State_Voter_Id_Primary_Key	N	Guid	16	VoteCal primary key associated to this voter

Field	Required?	Formatting	Max Len	Notes
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary Key from local EMS Voter table
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
UID	N	Alphanumeric	15	VoteCal unique identifier
Drivers_License	N	Alphanumeric	24	This registrant's driver's license or State-issued ID number, if available.

DL_Verification_Status	N	Alphanumeric	1	Y = Verified
				N = Not Verified
				P = Pending Verification

DL_Verification_Date	N	Date	10	The date this registrant's drivers license was verified in MM-DD-YYYY format
SSN4	N	Alphanumeric	4	Last four digits of SSN (if available)

SSN_Verification_Status	N	Alphanumeric	1	Y = Verified
				N = Not Verified
				P = Pending Verification

SSN_Verification_Date	N	Date	10	The date this registrant's Social Security Number was verified in MM-DD-YYYY format.
Name_Prefix	N	Alphanumeric	5	From Table A-34, the name prefix associated to this registrant.
First_Name	N	Alphanumeric	26	First name of this registrant.
Middle_Name	N	Alphanumeric	26	Middle name or initial of this registrant
Last_Name	Y	Alphanumeric	32	Last name of this registrant.

Field	Required?	Formatting	Max Len	Notes
Name_Suffix	N	Alphanumeric	4	Per Table A-10, the name suffix associated to this registrant. If no suffix is specified, this field remains null.
Voter_Status	Y	Alphanumeric	1	Per Table A-26, the status of this registrant.
Status_Reason	N	Alphanumeric	5	If a non-active status, the reason associated to this registrant's status per Table A-27. Status reasons for Active voters are optional.
Date_of_Birth	N	Date	10	Date of birth for this registrant in MM-DD-YYYY format (if available).
Alternate_Name	N	Alphanumeric	72	Any alternate names maintained for this registrant.
Affidavit_Number	Y	Alphanumeric	15	The number associated to the Registration Affidavit for this registrant.
Affidavit_Date_Received	Y	Date	10	The date this registrant's affidavit was received by the local jurisdiction in MM-DD-YYYY format.
Affidavit_Date_Executed	Y	Date	10	The date this registrant's affidavit was executed in MM-DD-YYYY format.
Registration_Date	Y	Date	10	The effective date of this registration.
Original_Registration_Date	Y	Date	10	The original date of registration for this registrant. If no previous registration date is available, use registration date.
Gender	N	Alphanumeric	1	Per Table A-8 (M, F)

Field	Required?	Formatting	Max Len	Notes
Accessibility_Code	N	Alphanumeric	5	Per Table A-28, the type of assistance this registrant requires. If multiple types of assistance are required, you may enter up to five single-digit codes in this field.
Language_Code	N	Alphanumeric	1	Per Table A-4, the language preference of this registrant.
Registration_Source	Y	Alphanumeric	3	Per Table A-2, this registrants source of registration (DMV, Other Agency, etc)
Registration_Method	Y	Alphanumeric	1	Per Table A-3, the method of registration (Counter, Mail, etc).
UOCAVA_Code	N	Alphanumeric	2	Per Table A-29, the UOCAVA category applicable to this registrant, or null.
Party_Code	Y	Alphanumeric	3	Per Table A-6, the partisan preference of this registrant.
Confidentiality_Code	N	Alphanumeric	1	Per Table A-32, the confidentiality code that applies to this registrant.
Confidentiality_App_Date	N	Date	10	Date this registrant was assigned a confidential status in MM-DD-YYYY format.
Place_of_Birth_Code	N	Alphanumeric	3	Per Table A-5, the place of birth for this registrant (if available).
Place_of_Birth_Text	N	Alphanumeric	30	If no entry in Table A-5 matches this registrant’s information, a free-form place of birth.
ID_Requirement	N	Alphanumeric	1	Per Table A-9, the status of ID requirements for voters who register by mail. If registration method is not by-mail, this field should remain null.

Field	Required?	Formatting	Max Len	Notes
Address_Line_1	N	Alphanumeric	40	Free Form Residential for non-standard addresses. This field should remain null for standardized addresses.
Address_Line_2	N	Alphanumeric	40	Free Form Residential for non-standard addresses. This field should remain null for standardized addresses.
House_Number	N	Alphanumeric	10	The building number associated with the registrant's current residence, not to be confused with the number of the street. If a non-standard (free-form) address is provided, these parsed field values should be null.
House_Suffix	N	Alphanumeric	5	A suffix, such as "A" associated to this registrant's residential address. Example: 123 A Main St
House_Fraction	N	Alphanumeric	4	A fractional value associated to this registrant's residential address. Example: 123 ½ Main St
Pre_Direction	N	Alphanumeric	2	Per Table A-12, the pre-directional value assigned to this registrant's residential address.
Post_Direction	N	Alphanumeric	2	Per Table A-12, the post-directional value assigned to this registrant's residential address.
Street_Name	N	Alphanumeric	30	The name (or number) of the street associated with the registrant's current residence address.
Street_Suffix	N	Alphanumeric	4	Per Table A-13, the street suffix associated to this registrant's residential address (ST, DR, etc)

Field	Required?	Formatting	Max Len	Notes
Unit_Type	N	Alphanumeric	4	Per Table A-30, the type of unit located within a multi-residential community such as, Apartment, Building, Lot, etc.
Unit_Number	N	Alphanumeric	12	The alphanumeric representation of the unit number associated with the registrant's current residence building number.
City	N	Alphanumeric	30	The name of the city associated with the registrant's current residence address.
State	N	Alphanumeric	2	The USPS state code for the registrant's current residence address. This will typically be "CA" for registrant data stored in County VR Systems. If outside of California, this field must conform to the State codes provided in Table A-11
Zip_Code	N	Alphanumeric	5	The five-digit zip code for the registrant's current address.
Zip_Code_4	N	Alphanumeric	4	The four-digit zip+4 codes for the registrants current address.
Mail_Address_Line_1	N	Alphanumeric	40	Free form mailing. Use only if mailing address is different than the residential address
Mail_Address_Line_2	N	Alphanumeric	40	Free form mailing. Use only if mailing address is different than the residential address.
Mail_Address_Line_3	N	Alphanumeric	40	Free form mailing. Use only if mailing address is different than the residential address.
Mail_Address_Line_4	N	Alphanumeric	40	Free form mailing. Use only if mailing address is different than the residential address.
Precinct_Part	Y	Alphanumeric	14	Precinct Number and Precinct Part, separated by a period as defined in the Precinct Detail View (example: 99999.02)

Field	Required?	Formatting	Max Len	Notes
Telephone1	N	Alphanumeric	16	The 10-digit primary telephone number associated to this registrant immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111
Telephone1Type	N	Alphanumeric	1	A value from Table A-33 indicating the type of phone.
Telephone2	N	Alphanumeric	16	A 10-digit alternate telephone number associated to this registrant immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111
Telephone2Type	N	Alphanumeric	1	A value from Table A-33 indicating the type of phone.
Telephone3	N	Alphanumeric	16	A 10-digit alternate telephone number associated to this registrant immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111
Telephone3Type	N	Alphanumeric	1	A value from Table A-33 indicating the type of phone.
Telephone4	N	Alphanumeric	16	A 10-digit alternate telephone number associated to this registrant immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111
Telephone4Type	N	Alphanumeric	1	A value from Table A-33 indicating the type of phone.
Email_Address	N	Alphanumeric	100	This registrant’s primary email address (if available).

Field	Required?	Formatting	Max Len	Notes
Signature_Image	N	Binary		Binary representation of this registrant's current signature image.

2.2.2.1.11Voter Activity View

This view provides details related to actions taken or revisions performed on an existing registrant's record at the local EMS level. This view will be utilized to track activities that occur at the local level that are not tracked or maintained in VoteCal (e.g. certain mailings). On the local EMS system, this view should be named VOTECAL_VOTER_ACTIVITY.

Table 2-11 Voter Activity View

Field Name	Required?	Formatting	Max Length	Notes
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from the local EMS Voter table
State_Voter_Id_Primary_Key	Y	Guid	16	VoteCal primary key associated to this voter
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table *A*-1 identifying the county that generated or is the intended recipient of a transaction record.
Activity_Id	Y	Numeric		The local primary key associated to this activity record.
Activity_User	Y	Alphanumeric	12	User who performed this activity
Activity_Date	Y	Datetime	19	The date of the most recent edit or change to the voter record in MM-DD-YYYY HH:MM:SS format.
Effective_Date	Y	Datetime	19	The date and time (if available) this activity occurred in MM-DD-YYYY HH:MM:SS format.
Activity_Type	Y	Alphanumeric	5	Per Table A-31, the type code or reason code for the most recent edit or change made to the voter record.

Field Name	Required?	Formatting	Max Length	Notes
Activity_Param1	Y	Alphanumeric	128	A description of the activity (example: VNC SENT), or in the case of an edit, the value that existed prior to the edit being performed. Using an address change as an example, this field should contain the complete previous address in free-form.
Activity_Param2	N	Alphanumeric	128	Additional description of the activity, or in the case of an edit, the updated values corresponding to the activity.
Activity_Description	Y	Alphanumeric	25	This field should contain a brief description of the activity. If no information is available for the most recent edit, this field should return the generic description “Record Edited”

2.2.2.1.12Provisional Voter View

This view provides all pertinent details related to a provisional ballot issued. On the local EMS system, this view should be named VOTECAL_PROVISIONAL_VOTER.

Table 2-12 Provisional Voter View

Field	Required?	Formatting	Max Length	Notes
Provisional_Ballot_Id	Y	Alphanumeric	10	This field must match the ID on any ballot receipt issued, or in lieu of a receipt, should be the primary key from the local EMS system.
County_Voter_Id_Primary_Key	N	Alphanumeric	10	If this ballot was issued to a currently registered voter, the Primary Key from local EMS Voter table. If registration is undetermined, this field should be null

Field	Required?	Formatting	Max Length	Notes
Jurisdiction_Id	Y	Alphanumeric	2	A code from Table A-1 identifying the county that generated or is the intended recipient of a transaction record.
County_Election_Id	Y	Alphanumeric	10	Primary key designating this election within the local EMS Elections table
Drivers_License	N	Alphanumeric	16	This provisional voter's driver's license or State-issued ID number, if available.
SSN4	N	Alphanumeric	4	Last four digits of SSN (if available)
Name_Prefix	N	Alphanumeric	5	From table A-34, a code indicating the name prefix associated to this registrant.
First_Name	N	Alphanumeric	26	First name of this provisional voter.
Middle_Name	N	Alphanumeric	26	Middle name or initial of this provisional voter
Last_Name	Y	Alphanumeric	32	Last name of this provisional voter.
Name_Suffix	N	Alphanumeric	4	Per Table A-10, the name suffix provided by this provisional voter
Date_of_Birth	N	Date	10	Date of birth for this provisional voter in MM-DD-YYYY format (if available).
Gender	N	Alphanumeric	1	Per Table A-8 (M, F)
Language_Code	N	Alphanumeric	1	Per Table A-4, the language preference of this provisional voter.
Party_Code	N	Alphanumeric	3	Per Table A-6, the partisan preference of this provisional voter.
Place_of_Birth_Cod e	N	Alphanumeric	3	Per Table A-5, the place of birth for this registrant (if available).

Field	Required?	Formatting	Max Length	Notes
Counted	N	Alphanumeric	1	'Y' if this ballot was counted, 'N' if it was not.
Rejection_Reason	N	Alphanumeric	2	Per Table A-15, the reason this provisional ballot was rejected.

2.2.2.1.13Affidavit Request View

This view provides detail on individuals and/or organizations requesting the issuance of affidavits. On the local EMS system, this view should be named VOTECAL_AFFIDAVIT_REQUEST.

Table 2-15 Affidavit Request View

Field	Required?	Formatting	Max Length	Notes
Jurisdiction_ID	Y	Alphanumeric	2	The two-digit code identifying the issuing jurisdiction
Local_Request_ID	Y	Alphanumeric	10	Primary key from the local EMS system identifying this request.
Organization_Name	N	Alphanumeric	120	If request is submitted by an organization, the name of that organization
Organization_Address_1	N	Alphanumeric	75	Free-form address of the organization
Organization_Address_2	N	Alphanumeric	75	Free-form address of the organization
Organization_Phone	N	Alphanumeric	16	A 10-digit primary telephone number associated to this organization immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111

Field	Required?	Formatting	Max Length	Notes
Organization_Fax	N	Alphanumeric	16	A 10-digit fax number associated to this organization immediately followed by up to 6-digit extension (if any). Use no parenthesis, dashes, or special characters. Example: 91655533441111
Requestor_Email	N	Alphanumeric	120	This requestor's email address.
RequestDate	Y	Date	10	The date this request was submitted in MM-DD-YYYY format.
IssueDate	Y	Date	10	If issued, the date this request was fulfilled.
Number_Of_Affidavits_Requested	Y	Numeric	10	The total number of affidavits requested.
Distribution_Method_Code	Y	Alphanumeric	2	From Table A-35, a code indicating the distribution method.
Distribution_Method_Text	Y	Alphanumeric	250	A textual description of the distribution method proposed for this request.

2.2.2.1.14Affidavit Issuance View

This view provides detail on groups of affidavits pertaining to a specific request. There may be multiple issuance records associated to a single request record (for instance, requests for affidavits in varying languages). On the local EMS system, this view should be named VOTECAL_AFFIDAVIT_ISSUANCE.

Table 2-14 Affidavit Issuance View

Field	Required?	Formatting	Max Length	Notes
Jurisdiction_ID	Y	Alphanumeric	2	The two-digit code identifying the issuing jurisdiction
Local_Request_ID	Y	Alphanumeric	10	Primary key from the local EMS system identifying this request. This value links this issuance to a specific request.

Field	Required?	Formatting	Max Length	Notes
Alpha_Code	Y	Alphanumeric	2	The alpha code (usually language code) used in the affidavit preamble.
NVRA_Source	Y	Alphanumeric		From Table A-2, the source of registration applicable to this request.
Language_Code	Y	Alphanumeric	2	From Table A-4, the language specified for this range of affidavits.
Beginning_Number	Y	Alphanumeric	10	The beginning number of the range of affidavits issued.
Ending_Number	Y	Alphanumeric	10	The ending number of the range of affidavits issued.
Returned	N	Alphanumeric	1	If this issuance represents a group of returned affidavits, enter ‘Y’, else ‘N’.
Comment	N	Alphanumeric	250	Any comments associated with this issuance.

2.2.2.1.15View Use Cases

An EMS Vendor’s application will support and comply with the following Use Cases by creating the views (previously detailed in this specification) for use by the Batch Data Exchange Service:

Use Case Number	Use Case Name	Views / Tables
04.10.01	Process Election Participation Batch	VOTECAL_ELECTION_PARTICIPATION_VIEW
04.12.01	Process Provisional Voting Batch	VOTECAL_PROVISIONAL_VOTER_VIEW
04.13.01	Process Precinct-District Mapping Batch	VOTECAL_PRECINCT_DISTRICT_MAPPING_VIEW
04.14.01	Process Voter Record Update Batch	VOTECAL_VOTER_VIEW
04.15.01	Process Election Polling Place Batch	VOTECAL_ELECTION_POLL_VIEW
04.16.01	Process Voter Activity Batch	VOTECAL_VOTER_ACTIVITY_VIEW
04.17.01	Send Image Batch	VOTECAL_DOCUMENT_VIEW
04.18.01	Process Batch Data Exchange	*All Views and Tables*

Use Case Number	Use Case Name	Views / Tables
04.21.01	Compare Voter Registrations for Independent County through VoteCal	VOTECAL_VOTER_VIEW

2.2.2.2 Data Stored in EMS Received from VoteCal in Batch (Database Tables)

Core Requirement: The EMS vendors must develop and implement new database tables in the underlying database of their remediated EMS systems into which data will be inserted.

Data will be written to these tables through the VoteCal Data Exchange Service running on a server at the county which has access to the local county EMS database. Structures need to be created and maintained through the local EMS system.

2.2.2.2.1 Official Voter Eligibility Table

Description: VoteCal will update this table with the official list of registration, status, and eligibility which can be used for roster generation. This table may also be utilized for auditing processes to ensure synchronicity between VoteCal and the local EMS system. On the local EMS system, this table should be named VOTECAL_VOTER_ELIGIBILITY. The table shall include the fields defined below with appropriate native data types for what is described under the formatting header. All strings shall be Unicode.

Field Name	Required?	Formatting	Field Length	Notes
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from the local EMS Voter table. Vendor is advised to create an index based on this field and designate as primary key for this table.
State_Voter_Id_Primary_Key	Y	Guid	16	VoteCal primary key associated to this voter
Name_Prefix	N	Alphanumeric	5	From Table A-34, the name prefix associated to this registrant.
First_Name	N	Alphanumeric	26	This registrant's first name
Middle_Name	N	Alphanumeric	26	This registrant's middle name or initial
Last_Name	Y	Alphanumeric	32	This registrant's last name
Name_Suffix	N	Alphanumeric	4	This registrant's name suffix (if any)

Field Name	Required?	Formatting	Field Length	Notes
Address_Line_1	Y	Alphanumeric	40	Free-form address for this registrant
Address_Line_2	Y	Alphanumeric	40	Free-form address for this registrant
ID_Required	Y	Alphanumeric	1	'Y' if registrant is required to show ID, 'N' if not.
Ballot_Issued	Y	Alphanumeric	1	'Y' if a ballot has been issued to this registrant, 'N' if no ballot has been issued.
Generation_Date	Y	Datetime		The date and time this listing was generated in MM-DD-YYYY HH:MM:SS format to facilitate supplemental roster generation.
VBM_Returned	Y	Alphanumeric	1	If a mail ballot has been returned for this registrant then 'Y', else 'N'

2.2.2.2.2 Cass Address Table

Description: VoteCal will update this table with CASS standardized addresses which are available for local use. These individual address records will relate to local voter records. On the local EMS system, this table should be named VOTECAL_CASS_ADDRESS. The table shall include the fields defined below with appropriate native data types for what is described under the formatting header. All strings shall be Unicode.

Field Name	Required?	Formatting	Max Length	Notes
County_Voter_Id_Primary_Key	Y	Alphanumeric	10	Primary key from the local EMS Voter table. Vendor is advised to create an index based on this field and designate as primary key for this table.
State_Voter_Id_Primary_Key	Y	Guid	16	VoteCal primary key associated to this voter.
CASS_Status	Y	Alphanumeric	2	A status code from CASS indicating the status of this address entry.
Cass_Comment	N	Alphaumeric	50	Any comments supplied by CASS related to this address entry.

Field Name	Required?	Formatting	Max Length	Notes
Delivery_Point	N	Alphanumeric	25	The delivery point designation provided by the CASS system.
Carrier_Route	N	Alphanumeric	25	The carrier route designation provided by the CASS system.
House_Number	N	Alphanumeric	9	The building number associated with the registrant's current residence as provided by the CASS system.
House_Suffix	N	Alphanumeric	4	The house suffix associated with the registrant's current residence as provided by the CASS system.
House_Fraction	N	Alphanumeric	4	The house fraction provided by the CASS system.
Pre_Direction	N	Alphanumeric	2	The pre-direction provided by the CASS system.
Post_Direction	N	Alphanumeric	2	The post-direction provided by the CASS system.
Street_Name	N	Alphanumeric	30	The street name provided by the CASS system.
Street_Suffix	N	Alphanumeric	4	The street suffix provided by the CASS system.
Unit_Type	N	Alphanumeric	4	The unit type provided by the CASS system.
Unit_Number	N	Alphanumeric	10	The unit number provided by the CASS system.
City	N	Alphanumeric	25	The city provided by the CASS system.
State	N	Alphanumeric	2	The state provided by the CASS system.
Zip_Code	N	Alphanumeric	5	The zip code provided by the CASS system.
Zip_Code_4	N	Alphanumeric	4	The zip+4 value provided by the CASS system.

2.2.2.3 Data Translation

All data transferred to VoteCal by the local EMS systems must conform to standards defined in this document. Data not meeting these standards will be rejected with an appropriate message being returned. All translations must be performed at the local level prior to submission to VoteCal.

2.2.2.4 Table Use Cases

An EMS Vendor's application will support and comply with the following Use Cases by creating the tables (previously detailed in this specification) for use by the Batch Data Exchange Service:

Use Case Number	Use Case Name	Tables
03.56.01	Process CASS Record File	VOTECAL_CASS_ADDRESS_TABLE
04.18.01	Process Batch Data Exchange	*All Views and Tables*
06.01.01	Generate Election Roster	VOTECAL_VOTER_ELIGIBILITY_TABLE

2.2.3 Application Programming Interface and Method Call Workflows

Core Requirement: The EMS vendors are required to implement functionality in the remediated EMS system to interact with VoteCal through the specified set of Web Service APIs. The use of the APIs specified in this section is for the purpose of implementing business logic and functionality set forth in the use cases of Section 2.2.1.

This section presents the VoteCal application programming interface (the Web Service API) and method (function) call workflows that EMS systems will be required to interact with. The section begins describing how the detailed API documentation exists in the form of external attachments. The section then proceeds with a descriptive overview of the Web Service API methods, parameters, responses, and their related workflows.

Web Service API Reference and WSDL External Attachments

Microsoft Compiled Help Web Service API Reference

A Web Service API Reference is attached to this deliverable (2.5) and will also be included as an attachment to Deliverable 2.3 (VoteCal System Detailed System Design Specification). This reference documentation comes in the form of an MSDN-style Compiled HTML Help File.

This approach to documentation facilitates readability and usability as every Web Service API method contains hyperlinks to parameter and response object class definitions as well as descriptions of class object member properties. The reader can click from cross reference to cross reference rather than thumb through pages of a single word document. Furthermore, this documentation is generated from a combination of the compiled source and comments contained therein, ensuring accuracy and ease of maintainability.

To view the top level list of all Web Service API Method definitions contained in the compiled help reference:

1. Unpack the provided ZIP file.
2. Within the resulting folder, launch the VoteCalEMSVendorIntegration API Compiled HTML Help File.
3. Depending on your browser settings you may need to allow certain security warnings.
4. In the left hand table of contents window expand the top item VoteCal.Management.Web.Services Namespace.
5. Under that, expand the VoteCalEMSIntegrationWebService Class.
6. Under that, expand the VoteCalEMSIntegrationWebService Methods.
7. Under that each distinct API method will appear and can be selected to view its definition and documentation.

SOAP Web Services Description Language (WSDL) File

A SOAP WSDL file for the Web Service API is attached to this Deliverable 2.5 and will also be included as an attachment to Deliverable 2.3.

Local county EMS vendors can import the WSDL file using their Integrated Development Environments (IDE) and automatically create proxy classes which locally represent all exposed functionality of the SOAP XML Web Service within their software.

Instructions for creating a SOAP Web Reference in Visual Studio (C++, C#, VB) are located here:

http://msdn.microsoft.com/en-us/library/tydxdyw9(VS.80).aspx

Instructions for consuming SOAP Web Services using eclipse (J2EE) are located here:

http://www.eclipse.org/webtools/initial-contribution/IBM/evalGuides/WebServicesToolsEval.html

Web Services Overview

The VoteCal Web Service API is a SOAP based Web service interface with methods to sync changes to voter data between a county's local EMS database and the VoteCal database. Web service methods are also provided to facilitate the county elections staff resolution of various issues with a voter's record which VoteCal has identified (Death Record Matches, Felon Record Matches, etc.). This section will summarize the conceptual framework of the Web Service API, the various methods exposed by the Web Service API, and the typical workflows associated with each method in a given use-case scenario.

For additional information on the Web Service API methods and properties of parameter and response objects please refer to the compiled help Web Service API Reference attached to this deliverable.

Functional Areas of Concern

The Web Service API addresses four primary functional areas of concern:

- Communication of **Voter Updates**, **New Registrations**, and **Transfers** which occur at the county and are entered into the county EMS software.
- Through the handling of **Messages**, county EMS application of updates made to a voter's record by VoteCal.
- Through the handling of **Work Items**, resolution of issues with a voter's record detected by VoteCal.
- **Searching** for voter information in the VoteCal database.

General Concepts

The general concepts underpinning the workflows supported by the Web Service API are described in this section.

Communicating County EMS Voter Updates, New Registrations, and Transfers

Anytime a change in voter data is made within the local county EMS the change must be communicated to VoteCal. These changes include routine updates to a voter's record, new registrations received by the county elections staff, and transfers of voters to a county from another county. All of these types of activities are handled by a single Web Service API method, **`UpdateStateVoter`**. Updates to voter data are ideally communicated in real-time, though *near* real time is acceptable (e.g. a job processing service sending updates every 10 seconds - see: 2.2.5 Transaction Timing).

VoteCal also requires a voter's Vote-by-Mail activity occurring at the county to be communicated on a near real-time basis. The **`UpdateStateVoterVoteByMailActivity`** Web Service API method handles the communication of such activity.

VoteCal Messages and Message Types

When updates occur to a voter's record in the VoteCal database, **Messages** are generated and sent to the county EMS system so that these changes can be applied to the voter's record in the county EMS system database. The **`GetOpenMessages`** Web Service API method is used to retrieve all VoteCal Messages for a county which have not yet been handled. The response object for this method contains various complete Message objects, each containing properties detailing the update made to the voter's record by VoteCal, as indicated by the message type, to the extent that the same updates can be applied to the voter in the county EMS database. Messages generated by VoteCal will continue to be returned by the `GetOpenMessages` method until they have been updated (using various API methods detailed in following sections) with a status indicating that they have been handled by the county EMS (i.e. the updates made by VoteCal have been applied to the voter in the county EMS database).

The `GetOpenMessages` method also serves as a heartbeat, informing VoteCal that the county EMS processing system is currently running. As such, the `GetOpenMessages` Web Service API method should be called on a routine basis to retrieve messages (e.g. every 10 seconds - see:
2.2.5 Transaction Timing).

The types of messages generated by VoteCal are as follows:

Message Type	Purpose Overview
`StateVoterStatusUpdated`	Indicates that VoteCal has changed the status of a voter. Since many of the changes to a voter's record involve only status this Message type, rather than the `StateVoterUpdated` type, is used for updates only involving status for the sake of efficiency (i.e. the entire Voter's record does not need to be transmitted to the county EMS).
`StateVoterUpdated`	Indicates that VoteCal has updated some aspect of a voter's record.
`StateVoterActivityCreated`	Indicates that new Voter Activity has been added to the Voter's record in the VoteCal Database.
`NewVoterReceived`	Indicates that a new voter registration has been received by VoteCal for the county receiving the message.
`DMVChangeOfAddressReceived`	Indicates that a DMV change of address record has been received by VoteCal for a voter.
`NationalChangeOfAddressReceived`	Indicates that a National change of address record has been received by for a voter.

VoteCal Work Items and Work Item Types

VoteCal periodically analyzes its database to detect issues with a voter's record (Death Record Matches, Felon Record Matches, etc.). In some cases, an issue can be automatically resolved by an update to the Voter's record by VoteCal, resulting in a Message being generated (as in the previous section). In other cases, some level of human interaction is required to resolve the issue. When human interaction is required for issue resolution, VoteCal generates a **Work Item**.

The **`GetOpenWorkItemsList`** method is used to retrieve all VoteCal Work Items for a county which have not yet been handled. Since Work Items are handled by elections staff (i.e. not in real-time), several work items may accumulate over time. The `GetOpenWorkItemList` method accepts paging and filtration parameters to accommodate for times when large numbers of work items exist. As is the case with the `GetOpenMessages` method, Work Items will continue to be returned by the `GetOpenWorkItemList` method until they have been updated (using various API methods detailed in following sections) with a status indicating the determination made for the Work Item by county elections staff. *Unlike* the `GetOpenMessages` method, the `GetOpenWorkItemList` method only returns high-level open Work Item definitions; enough information to display a UI grid of open Work Items, but not enough information to actually resolve the Work Items. Individual Web Service API methods (detailed in following sections) are provided for each Work Item type to retrieve the full and complete details for the Work Item (e.g. the `GetDeathRecordMatchCaseWorkItems` Web Service method).

The types of Work Items generated by VoteCal are as follows:

Work Item Type	Purpose Overview
`DeathRecordMatchCase`	Generated when death records received from external agencies match a voter's record. County elections staff need to Accept or Reject the match.
`FelonRecordMatchCase`	Generated when felon records from external agencies match a voter's record. County elections staff need to Accept or Reject the match.
`DuplicateRecordMatchCase`	Generated when a duplicate voter is found in the VoteCal Database. County elections staff need to Accept or Reject the match.
`DMVChangeOfAddressRecordMatchCase`	Generated when a `DMVChangeOfAddressReceived` Message could not be automatically processed (e.g. no precinct could be determined for the address) by the county EMS system. County elections staff need to indicate whether the change of address is to be ignored, or whether the elections staff will be manually handling the address update.
`NationalChangeOfAddressRecordMatchCase`	Generated when a `NationalChangeOfAddressReceived` Message could not be automatically processed (e.g. no precinct could be determined for the address) by the county EMS system. County elections staff need to indicate whether the change of address is to be ignored, or whether the elections staff will be manually handling the address update.

Work Item Type	Purpose Overview
TransferOut	Generated when a voter transfer has been initiated by another county (e.g. by a call to UpdateStateVoter with a transferStateVoterId (parameter supplied). County elections staff for the county the voter is transferring *from* need to Approve or Reject the transfer.
DataValidationError	Generated when a call to UpdateStateVoter contains data validation errors. County elections staff must Acknowledge the errors contained in a DataValidationError Work Item.

Error Handling

The response objects for all Web Service API methods contain a general ResponseStatus (Success, Error) and a ResponseStatusReason which can vary based on the method called - refer to the compiled help Web Service API Reference for technical details on the various ResponseStatusReason enumerations. In the case of the UpdateStateVoter method, the DataValidationErrorWorkItem mechanism is additionally utilized to provide detailed information on the nature of data validation errors as related to voter records.

Background Job Processing Windows Service

A background job processing service is typically needed to routinely check for and handle new VoteCal Messages. This service will frequently poll the GetOpenMessages method, modify voter data as dictated by the messages received, and call the various update Web Service API methods (e.g. UpdateStateVoterUpdatedMessages) for each message with a message status indicating that the message has been (and how it was) handled. As detailed in following sections, all update methods for Messages accept multiple Messages to be updated as a parameter (i.e. you can update multiple messages with one call).

Some EMS vendors may choose to use this job processing service to communicate voter data updates as an alternative to communicating the update at the time the update is made (through a call to UpdateStateVoter). This approach can eliminate the potential lag when calling the Web Service API at the time of update. However, this approach also eliminates the option of presenting a county user with the voter status and status reason code as determined by VoteCal directly following the update of an existing voter or registration of a new voter. Some county EMS vendors may opt for a user configurable mixed-mode approach to enhance the efficiency of various user interactions. From a VoteCal perspective, the preferred approach is real-time updates (i.e. communication of an update at the time of the update).

A GetOpenWorkItemCounts (detailed in following sections) method is provided so that the current count of various open work items can be obtained. Some EMS vendors may choose to use this information within the job processing service to send out alerts when a configurable threshold is reached.

EMS Software Desktop or Web Application

At a minimum, a county EMS vendors' desktop or Web application will need to be modified to communicate with the StateVoterSearch method during the data entry process for a new voter registration. As detailed in following sections, various business rules requiring different workflows apply when creating a new voter that is potentially a duplicate. Due to the nature of these business rules, and the parameter structure of the Web Service API, the EMS software must be aware of a potential duplicate voter *prior* to calling UpdateStateVoter. Using the StateVoterSearch method during data entry, the EMS application is provided this ability.

In addition to state voter searches, the EMS vendors' desktop or Web application will need to communicate county voter data updates to VoteCal (through a call to UpdateStateVoter). This assumes that the job processing service has not been solely tasked with this responsibility.

Though county elections staff may use the SOS VoteCal Web application to resolve Work Items, some EMS vendors may choose to use the available Web Service API methods to allow for Work Item resolution by users within their application. This will typically entail development of a screen with a paged grid containing a listing of all open Work Items (through a call to GetOpenWorkItemList, as well as resolution screens for each Work Item type which display the Work Item details in a manner that facilitates resolution of the Work Item by a user (e.g. through calls to GetDeathRecordMatchCaseWorkItems and UpdateDeathRecordMatchCaseWorkItems). As is the case with update methods for Messages, both the get and update methods for Work Items support getting and updating multiple Work Items in a single call. An EMS vendor may choose to use this feature to create rapid Work Item resolution screens, allowing the user to scroll-down through the pertinent details of several Work Items (of a given type) and resolving them in the process.

2.2.3.1 Statewide Voter Search

When county elections staff are entering new registrations into the county EMS system a search must be performed against the VoteCal voter database to determine if the new voter has a duplicate record in another county. The SearchStateVoters Web Service API method provides this functionality.

Methods for Searching the VoteCal Voter Database

For technical details on Web Service API methods and properties of parameter and response objects please refer to the compiled help Web Service API Reference attached to this deliverable.

Web Service API Method

`SearchStateVoters`

Purpose:

- Gives the county EMS the ability to search the VoteCal database for voters to determine whether a new voter registration is a potential duplicate or transfer.

Called When:

- A new voter registration is being entered into a county EMS.

Logic Rules:

- Returns abbreviated voter records and corresponding search match confidence ratings for state voters which match the supplied search parameters to an extent which exceeds the search confidence threshold as determined by SOS.

`GetStateVoters`

Purpose:

- Retrieves the complete voter records from the VoteCal database, including (if specified) associated open and closed Messages, Work Items, and Documents.

Called When:

- County elections staff requests VoteCal Messages, Work Items, or Documents associated with a voter or voters.

Logic Rules:

- To include Messages associated with voters in the returned StateVoters object, set the `includeMessages` parameter to true. The default for this parameter is false.
- To include Work Items associated with voters in the returned StateVoters object, set the `includeWorkItems` parameter to true. The default for this parameter is false.
- To include Documents associated with voters in the returned StateVoters object, set the `includeDocuments` parameter to true. The default for this parameter is false.

Web Service API Method

`VerifyVoterIdentification`

Purpose:

- Gives the county EMS the ability to verify the identification data for a voter during the data entry process for a new voter

Called When:

- A new voter registration is being entered into a county EMS.

Logic Rules:

- The `SuggestedDriversLicense` property of the response object will be populated only if the `VoterIdentificationMatchType` property of the response object equals `AlternateDLSuggested`

2.2.3.2 State Voter Update

For technical details on Web Service API methods and properties of parameter and response objects please refer to the compiled help Web Service API Reference attached to this deliverable.

Web Service API Method

`UpdateStateVoter`

Purpose:

- Communicating to VoteCal updates made to a voter's record in the local county EMS
- Communicating to VoteCal new voter registrations received by the county and entered into the county EMS

Called When:

- An update is made to a voter's record in the county EMS database
- A new voter (voter transfer) is registered in the county EMS database
- (Typically) A `NewVoterReceived` Message is received by the county EMS from VoteCal
- (Typically) A `DMVChangeOfAddressReceived` Message is received by the county EMS from VoteCal
- (Typically) A `NationalChangeOfAddressReceived` Message is received by the county EMS from VoteCal

Logic Rules:

- If a voter being updated does not have a `StateVoterId` or `StateVoterIdPrimaryKey`, VoteCal will treat the voter as a new registration
- If a voter search results were returned from a call to `SearchStateVoters` during the data entry process for a new registration, the matching voters **must** be handled as either a transfer or an ignored potential duplicate. When handling a new registration as a transfer, a `TransferOut` Work Item will be generated for the county the voter is transferring from (for verification purposes). When ignoring a duplicate match case for the new registration a `DuplicateMatchCase` work item will still be generated for the county in which the duplicate voter exists (for verification purposes)
- To indicate a potential duplicate state voter is to be ignored (returned as a match result from a call to the `SearchStateVoters` method during data entry for a new registration), add the duplicate voter's `StateVoterIdPrimaryKey` to the `duplicatesToIgnoreStateVoterIdPrimaryKeys` `Guid` array property of the `UpdateStateVoterProperties` parameter.

`UpdateStateVoter` (cont.)

- To indicate a voter is to be handled as a transfer, set the `transferStateVoterIdPrimaryKey` property of the `UpdateStateVoterProperties` parameter to the `StateVoterIdPrimaryKey` of the voter to be transferred (returned as a match result from a call to the `SearchStateVoters` method during data entry for a new registration)
- VoteCal does not have the street information needed place a new voter (or one who has changed their address) in a precinct. VoteCal must rely on the local county EMS systems for this information. As such, `NewVoterReceived`, `DMVChangeOfAddressReceived`, and `NationalChangeOfAddressReceived` messages **ought to** result in a call to `UpdateStateVoter` so that VoteCal can be updated with the precinct information for the voter (and in the case of new registrations, the `CountyVoterId` and `CountyVoterIdPrimaryKey`).
- To indicate that a state voter update is in response to a `NewVoterReceived` message, set the `refNewVoterReceivedMessageId` property of the `UpdateStateVoterProperties` parameter to the `MessageId` of the `NewVoterReceived` Message
- To indicate that a state voter update is in response to a `DMVChangeOfAddressReceived` Message, set the `refDMVChangeOfAddressReceivedMessageId` property of the `UpdateStateVoterProperties` parameter to the `MessageId` of the `DMVChangeOfAddressReceived` Message
- To indicate that a state voter update is in response to a `NationalChangeOfAddressReceived` Message, set the `refNationalChangeOfAddressReceivedMessageId` property of the `UpdateStateVoterProperties` parameter to the `MessageId` of the `NationalChangeOfAddressReceived` Message
- There will be rare cases when `NewVoterReceived`, `DMVChangeOfAddressReceived`, and `NationalChangeOfAddressReceived` Messages cannot be properly processed by the county EMS due to some error or missing information – in these cases the update status method corresponding to the message type should be used to update the status of the message rather than referencing the Message in a call to `UpdateStateVoter`

`UpdateStateVoter` (cont.)

- The properties of the Voter object parameter are subject to various data validation rules. The data validation rules for each property of the Voter object are detailed in the compiled help Web Service API Reference. If properties of the Voter object supplied in the call to `UpdateStateVoter` fail data validation, a `DataValidationError` Work Item will be generated to be acknowledged by the county elections staff.
  - Data validation errors have two levels of severity: `Deficiency` and `Fatal`. Deficiency data validation errors do *not* prevent a voter from being added to or updated within VoteCal. Fatal errors *do* prevent a voter from being added to or updated within VoteCal.
  - `DataValidationErrorWorkItem` objects contain an array of `DataValidationError` objects. If multiple properties of a Voter object fail validation, the `DataValidationErrorWorkItem` will contain multiple errors.
  - If any property of a Voter object fails validation with a severity level of fatal (`DataValidationError.DataValidationErrorSeverityType = DataValidationErrorSeverityType.Fatal`) the Voter will not be added to or updated within VoteCal.
  - When a `DataValidationErrorWorkItem` is generated in response to a call to `UpdateStateVoter`, the `DataValidationWorkItem` will be included in the `VoterWorkItems` property of the Voter object property of the `UpdateStateVoterResponse` object returned from the call. The `ResponseStatus` and `ResponseStatusReason` of the `UpdateStateVoterResponse` object will also be set to indicate whether or not a data validation error was encountered and whether or not one of the errors was fatal. In the case of new voter registrations, if any of the data validation errors were fatal, a `StateVoterId` and `StateVoterIdPrimaryKey` *will not* be assigned to a voter. The calling EMS can iterate through the array of `DataValidationErrors` to determine if the new voter registration fatal data validation error case applies
- While the `DataValidationErrorWorkItem` mechanism is capable of handling most types of voter data validation errors, a `CountyVoterIdPrimaryKey` is required as a bare-minimum for creation of the `DataValidationErrorWorkItem` (i.e. the work item must refer to some voter in the County EMS database). Should the Voter object supplied as a parameter not have the `CountyVoterIdPrimaryKey` property supplied, a general response error will be returned with a response error reason of `MissingCountyVoterIdPrimaryKey`.

2.2.3.3 Receiving/Retrieving VoteCal Messages and Work Items

For technical details on Web Service API methods and properties of parameter and response objects please refer to the compiled help Web Service API Reference attached to this deliverable.

Web Service API Method

`GetOpenMessages`

Applicable Message Type(s):

- All Message Types

Purpose:

- Retrieves all open VoteCal messages for a county
- Serves as a heartbeat informing VoteCal that the county EMS job processing service is running.

Called When:

- A county's EMS job processing service is ready to receive new VoteCal messages (typically on a routine basis every 10 seconds - see: 2.2.5 Transaction Timing).

Logic Rules:

- Returns all messages with a `MessageStatus` of `Open`
- Messages will be returned continually by this method until they have a `MessageStatus` of `Closed`

Web Service API Method
`UpdateStateVoterStatusUpdatedMessages`

Applicable Message Type(s):

- `StateVoterStatusUpdated`

Purpose:

- To update the `StateVoterStatusUpdatedMessageStatus` of `StateVoterStatusUpdated` Messages, ensuring the message was received and processed by the local county EMS.

Called When:

- The `VoterStatus` and `VoterStatusReasonCode` specified by the Message has been applied to the voter's record in the local county EMS database (`StateVoterStatusUpdatedMessageStatus = StateVoterStatusUpdatedMessageStatus.VoterStatusUpdateApplied`)
- The `VoterStatus` and `VoterStatusReasonCode` specified by the Message could not be applied by the local county EMS (`StateVoterStatusUpdatedMessageStatus = StateVoterStatusUpdatedMessageStatus.CannotApplyVoterStatusUpdate`)

Logic Rules:

- Once the `StateVoterStatusUpdatedMessageStatus` property of the Message has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Message (`MessageStatus`) to `Closed` and will no longer return the Message from calls to `GetOpenMessages`.

Web Service API Method

`UpdateStateVoterUpdatedMessages`

Applicable Message Type(s):

- `StateVoterUpdated`

Purpose:

- To update the `StateVoterUpdatedMessageStatus` of `StateVoterUpdated` Messages, ensuring the message was received and processed by the local county EMS.

Called When:

- The updates in the `Voter` object supplied in the Message have been applied to the voter's record in the local county EMS database (`StateVoterUpdatedMessageStatus = StateVoterUpdatedMessageStatus.VoterUpdateApplied`)
- The updates in the `Voter` object supplied in the Message could not be applied by the local county EMS (`StateVoterUpdatedMessageStatus = StateVoterUpdatedMessageStatus.CannotApplyVoterUpdate`)

Logic Rules:

- Once the `StateVoterUpdatedMessageStatus` property of the Message has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Message (`MessageStatus`) to `Closed` and will no longer return the Message from calls to `GetOpenMessages`.

Web Service API Method

`UpdateStateVoterActivityCreatedMessages`

Applicable Message Type(s):

- `StateVoterActivityCreated`

Purpose:

- To update the `StateVoterActivityCreatedMessageStatus` of `StateVoterActivityCreated` Messages, ensuring the message was received and processed by the local county EMS.

Called When:

- The state voter activity specified by the Message has been added to the voter's record in the local county EMS database (`StateVoterActivityCreatedMessageStatus = StateVoterActivityCreatedMessageStatus .VoterActivityApplied`)
- The state voter activity specified by the Message could not be applied by the local county EMS (`StateVoterActivityCreatedMessageStatus = StateVoterActivityCreatedMessageStatus.CannotApplyVoterActivity`)

Logic Rules:

- Once the `StateVoterActivityCreatedMessageStatus` property of the Message has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Message (`MessageStatus`) to `Closed` and will no longer return the Message from calls to `GetOpenMessages`.

Web Service API Method

UpdateNewVoterReceivedMessages

Applicable Message Type(s):

- NewVoterReceived

Purpose:

- To update the NewVoterReceivedMessageStatus of NewVoterReceived Messages, ensuring the message was received and processed by the local county EMS.

Called When:

- The NewVoter supplied in the Message could not be added to the local county EMS (NewVoterReceivedMessageStatus = NewVoterReceivedMessageStatus.CannotApplyNewVoter)

Logic Rules:

- Once the NewVoterReceivedMessageStatus property of the Message has been updated to a status other than Unhandled, VoteCal will set the overall status of the Message (MessageStatus) to Closed and will no longer return the Message from calls to GetOpenMessages.
- Updates to the NewVoterReceivedMessageStatus of NewVoterReceived Messages typically occur by supplying the MessageId as a reference in calls to UpdateStateVoter. Since VoteCal does not have the information needed to apply a precinct to a new voter, VoteCal must rely on the local County EMS for such information. The UpdateNewVoterReceivedMessages method is still made available for cases when (for some reason) the new voter could not be added to the local county EMS database.

Web Service API Method

`UpdateDMVChangeOfAddressRecordReceivedMessages`

Applicable Message Type(s):

- `DMVChangeOfAddressRecordReceived`

Purpose:

- To update the `DMVChangeOfAddressRecordReceivedMessageStatus` of `DMVChangeOfAddressRecordReceived` Messages, ensuring the Message was received and processed by the local county EMS.

Called When:

- The `ChangeOfAddressInfo` supplied in the Message could not be applied to the voter's record by the local county EMS (`DMVChangeOfAddressRecordReceivedMessageStatus = DMVChangeOfAddressRecordReceivedMessageStatus.CannotApplyDMVChangeOfAddress`)

Logic Rules:

- Once the `DMVChangeOfAddressRecordReceivedMessageStatus` property of the Message has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Message (`MessageStatus`) to `Closed` and will no longer return the Message from calls to `GetOpenMessages`.
- Updates to the `DMVChangeOfAddressRecordReceivedMessageStatus` of `DMVChangeOfAddressRecordReceived` Messages typically occur by supplying the `MessageId` as a reference in calls to `UpdateStateVoter`.. The `UpdateDMVChangeOfAddressReceivedMessages` method is still made available for cases when (for some reason) the DMV change of address could not be applied to the voter's record in the local county EMS database.

Web Service API Method

UpdateNationalChangeOfAddressRecordReceivedMessages

Applicable Message Type(s):

- NationalChangeOfAddressRecordReceived

Purpose:

- To update the NationalChangeOfAddressRecordReceivedMessageStatus of NationalChangeOfAddressRecordReceived Messages, ensuring the Message was received and processed by the local county EMS.

Called When:

- The ChangeOfAddressInfo supplied in the Message could not be applied to the voter's record by the local county EMS (NationalChangeOfAddressRecordReceivedMessageStatus = NationalChangeOfAddressRecordReceivedMessageStatus.CannotApplyNationalChangeOfAddress)

Logic Rules:

- Once the NationalChangeOfAddressRecordReceivedMessageStatus property of the Message has been updated to a status other than Unhandled, VoteCal will set the overall status of the Message (MessageStatus) to Closed and will no longer return the Message from calls to GetOpenMessages.
- Updates to the NationalChangeOfAddressRecordReceivedMessageStatus of NationalChangeOfAddressRecordReceived Messages typically occur by supplying the MessageId as a reference in calls to UpdateStateVoter.. The UpdateNationaChangeOfAddressReceivedMessages method is still made available for cases when (for some reason) the National change of address could not be applied to the voter's record in the local county EMS database.

.

Methods for Handling VoteCal Work Items

For technical details on Web Service API methods and properties of parameter and response objects please refer to the compiled help Web Service API Reference attached to this deliverable.

Web Service API Method
`GetOpenWorkItemList` **Applicable Work Item Type(s):** • All Work Item Types **Purpose:** • Retrieves a paged and filtered list of all open VoteCal Work Items for a county **Called When:** • A county EMS User requests a list of Work Items **Logic Rules:** • Returns all Work Items with a `WorkItemStatus` of `Open` • Work Items will be returned continually by this method until they have a `WorkItemStatus` of `Closed` • For the sake of efficiency, this method returns high-level Work Item objects. To retrieve the full and complete details of a Work Item, call the corresponding get method for the Work Item type.
`GetOpenWorkItemCounts` **Applicable Work Item Type(s):** • All Work Item Types **Purpose:** • Returns the number of open Work Items by type **Called When:** • The number of open Work Items by type is needed, typically for the purpose of alerting elections staff **Logic Rules:** • Returns the number of Work Items by type with a `WorkItemStatus` of `Open`

Web Service API Method
GetDeathRecordMatchCaseWorkItems **Applicable Work Item Type(s):** - DeathRecordMatchCase **Purpose:** - Retrieves an array of DeathRecordMatchCaseWorkItem objects, each of which contain the DeathRecordMatchCase information details needed for county elections staff to make a determination on the match case **Called When:** - County elections staff requests one or more DeathRecordMatchCase Work Items **Logic Rules:** - Returns DeathRecordMatchCaseWorkItem objects based on an array of WorkItemId Guid objects supplied as a parameter.
UpdateDeathRecordMatchCaseWorkItems **Applicable Work Item Type(s):** - DeathRecordMatchCase **Purpose:** - Updates the DeathRecordMatchCaseWorkItemStatus of a DeathRecordMatchCaseWorkItem indicating the determination county elections staff have made on the match case **Called When:** - A DeathRecordMatchCase is accepted (DeathRecordMatchCaseWorkItemStatus = DeathRecordMatchCaseWorkItemStatus.Accepted) - A DeathRecordMatchCase is rejected (DeathRecordMatchCaseWorkItemStatus = DeathRecordMatchCaseWorkItemStatus.Rejected) **Logic Rules:** - Once the DeathRecordMatchCaseWorkItemStatus property of Work Item has been updated to a status other than Unhandled, VoteCal will set the overall status of the Work Item (WorkItemStatus) to Closed and will no longer return the Work Item from calls to GetOpenWorkItemList.

Web Service API Method
GetFelonRecordMatchCaseWorkItems **Applicable Work Item Type(s):** • FelonRecordMatchCase **Purpose:** • Retrieves an array of FelonRecordMatchCaseWorkItem objects, each of which contain the FelonRecordMatchCase information details needed for county elections staff to make a determination on the match case **Called When:** • County elections staff requests one or more FelonRecordMatchCase Work Items **Logic Rules:** • Returns FelonRecordMatchCaseWorkItem objects based on an array of WorkItemId Guid objects supplied as a parameter.
UpdateFelonRecordMatchCaseWorkItems **Applicable Work Item Type(s):** • FelonRecordMatchCase **Purpose:** • Updates the FelonRecordMatchCaseWorkItemStatus of a FelonRecordMatchCaseWorkItem indicating the determination county elections staff have made on the match case **Called When:** • A FelonRecordMatchCase is accepted (FelonRecordMatchCaseWorkItemStatus = FelonRecordMatchCaseWorkItemStatus.Accepted) • A FelonRecordMatchCase is rejected (FelonRecordMatchCaseWorkItemStatus = FelonRecordMatchCaseWorkItemStatus.Rejected) **Logic Rules:** • Once the FelonRecordMatchCaseWorkItemStatus property of Work Item has been updated to a status other than Unhandled, VoteCal will set the overall status of the Work Item (WorkItemStatus) to Closed and will no longer return the Work Item from calls to GetOpenWorkItemList.

Web Service API Method
`GetDuplicateRecordMatchCaseWorkItems` **Applicable Work Item Type(s):** - `DuplicateRecordMatchCase` **Purpose:** - Retrieves an array of `DuplicateRecordMatchCaseWorkItem` objects, each of which contain the `DuplicateRecordMatchCase` information details needed for county elections staff to make a determination on the match case **Called When:** - County elections staff requests one or more `DuplicateRecordMatchCase` Work Items **Logic Rules:** - Returns `DuplicateRecordMatchCaseWorkItem` objects based on an array of `WorkItemId Guid` objects supplied as a parameter.
`UpdateDuplicateRecordMatchCaseWorkItems` **Applicable Work Item Type(s):** - `DuplicateRecordMatchCase` **Purpose:** - Updates the `DuplicateRecordMatchCaseWorkItemStatus` of a `DuplicateRecordMatchCaseWorkItem` indicating the determination county elections staff have made on the match case **Called When:** - A `DuplicateRecordMatchCase` is accepted (`DuplicateRecordMatchCaseWorkItemStatus = DuplicateRecordMatchCaseWorkItemStatus.Accepted`) - A `DuplicateRecordMatchCase` is rejected (`DuplicateRecordMatchCaseWorkItemStatus = DuplicateRecordMatchCaseWorkItemStatus.Rejected`) **Logic Rules:** - Once the `DuplicateRecordMatchCaseWorkItemStatus` property of Work Item has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Work Item (`WorkItemStatus`) to `Closed` and will no longer return the Work Item from calls to `GetOpenWorkItemList`.

Web Service API Method
GetDMVChangeOfAddressRecordMatchCaseWorkItems **Applicable Work Item Type(s):** • DMVChangeOfAddressRecordMatchCase **Purpose:** • Retrieves an array of DMVChangeOfAddressRecordMatchCaseWorkItem objects, each of which contain the DMVChangeOfAddressRecordMatchCase information details needed for county elections staff to make a determination on the match case **Called When:** • County elections staff requests one or more DMVChangeOfAddressRecordMatchCase Work Items **Logic Rules:** • Returns DMVChangeOfAddressRecordMatchCaseWorkItem objects based on an array of WorkItemId Guid objects supplied as a parameter.
UpdateDMVChangeOfAddressRecordMatchCaseWorkItems **Applicable Work Item Type(s):** • DMVChangeOfAddressRecordMatchCase **Purpose:** • Updates the DMVChangeOfAddressRecordMatchCaseWorkItemStatus of a DMVChangeOfAddressRecordMatchCaseWorkItem indicating the determination county elections staff have made on the match case **Called When:** • A DMVChangeOfAddressRecordMatchCase is determined to be valid but will require some manual intervention by elections staff (DMVChangeOfAddressRecordMatchCaseWorkItemStatus = DMVChangeOfAddressRecordMatchCaseWorkItemStatus.StaffUpdating) • A DMVChangeOfAddressRecordMatchCase is determined to be invalid and will be ignored (DMVChangeOfAddressRecordMatchCaseWorkItemStatus = DMVChangeOfAddressRecordMatchCaseWorkItemStatus.Ignored) **Logic Rules:** • Once the DMVChangeOfAddressRecordMatchCaseWorkItemStatus property of Work Item has been updated to a status other than Unhandled, VoteCal will set the overall status of the Work Item (WorkItemStatus) to Closed and will no longer return the Work Item from calls to GetOpenWorkItemList.

Web Service API Method
`GetNationalChangeOfAddressRecordMatchCaseWorkItems` **Applicable Work Item Type(s):** • `NationalChangeOfAddressRecordMatchCase` **Purpose:** • Retrieves an array of `NationalChangeOfAddressRecordMatchCaseWorkItem` objects, each of which contain the `NationalChangeOfAddressRecordMatchCase` information details needed for county elections staff to make a determination on the match case **Called When:** • County elections staff requests one or more `NationalChangeOfAddressRecordMatchCase` Work Items **Logic Rules:** • Returns `NationalChangeOfAddressRecordMatchCaseWorkItem` objects based on an array of `WorkItemId Guid` objects supplied as a parameter.

Web Service API Method

UpdateNationalChangeOfAddressRecordMatchCaseWorkItems

Applicable Work Item Type(s):

- NationalChangeOfAddressRecordMatchCase

Purpose:

- Updates the NationalChangeOfAddressRecordMatchCaseWorkItemStatus of a DMVChangeOfAddressRecordMatchCaseWorkItem indicating the determination county elections staff have made on the match case

Called When:

- A NationalChangeOfAddressRecordMatchCase is determined to be valid but will require some manual intervention by elections staff (NationalChangeOfAddressRecordMatchCaseWorkItemStatus = NationalChangeOfAddressRecordMatchCaseWorkItemStatus.StaffUpdating)
- A NationalChangeOfAddressRecordMatchCase is determined to be invalid and will be ignored (NationalChangeOfAddressRecordMatchCaseWorkItemStatus = NationalChangeOfAddressRecordMatchCaseWorkItemStatus.Ignored)

Logic Rules:

- Once the NationalChangeOfAddressRecordMatchCaseWorkItemStatus property of Work Item has been updated to a status other than Unhandled, VoteCal will set the overall status of the Work Item (WorkItemStatus) to Closed and will no longer return the Work Item from calls to GetOpenWorkItemList.

GetTransferOutWorkItems

Applicable Work Item Type(s):

- TransferOut

Purpose:

- Retrieves an array of TransferOutWorkItem objects, each of which contain the TransferOut information details needed for county elections staff to make a determination on the transfer

Called When:

- County elections staff requests one or more TransferOut Work Items

Logic Rules:

- Returns TransferOutWorkItem objects based on an array of WorkItemId Guid objects supplied as a parameter.

Web Service API Method

UpdateTransferOutWorkItems

Applicable Work Item Type(s):

- TransferOut

Purpose:

- Updates the TransferOutWorkItemStatus of a TransferOutWorkItem indicating the determination county elections staff have made on the match case

Called When:

- A TransferOut is accepted (TransferOutWorkItemStatus = TransferOutWorkItemStatus.Accepted)
- A TransferOut is rejected TransferOutWorkItemStatus = TransferOutWorkItemStatus.Rejected)

Logic Rules:

- Once the TransferOutWorkItemStatus property of Work Item has been updated to a status other than Unhandled, VoteCal will set the overall status of the Work Item (WorkItemStatus) to Closed and will no longer return the Work Item from calls to GetOpenWorkItemList.

GetDataValidationErrorWorkItems

Applicable Work Item Type(s):

- DataValidationError

Purpose:

- Retrieves an array of DataValidationError objects, each of which contain the DataValidationError information details needed to inform county elections staff of data validation issues with a voter's record

Called When:

- County elections staff requests one or more DataValidationError Work Items

Logic Rules:

- Returns DataValidationError objects based on an array of WorkItemId Guid objects supplied as a parameter.

Web Service API Method

`UpdateDataValidationErrorWorkItems`

Applicable Work Item Type(s):

- `DataValidationError`

Purpose:

- Updates the `DataValidationErrorWorkItemStatus` of a `DataValidationErrorWorkItem` to acknowledge that a county elections staff member has been made aware of the data validation issues with a voter's record

Called When:

- A `DataValidationError` Work Item is acknowledged (`DataValidtionErrorWorkItemStatus` = `DataValidationErrorWorkItemStatus`.Acknowledged)

Logic Rules:

- Once the `DataValidationErrorWorkItemStatus` property of Work Item has been updated to a status other than `Unhandled`, VoteCal will set the overall status of the Work Item (`WorkItemStatus`) to `Closed` and will no longer return the Work Item from calls to `GetOpenWorkItemList`.

2.2.3.1 Vote By Mail Status Update

`UpdateStateVoterVoteByMailActivity`

Purpose:

- Communicating to VoteCal Vote By Mail activity for a given voter

Called When:

- A voter requests a Vote By Mail ballot
- A Vote By Mail ballot is mailed to a voter
- A Vote By Mail ballot is received
- A Vote By Mail Ballot is counted (or rejected)

Logic Rules:

None

2.2.3.2 Web Service API Use Cases

Use Case Number	Use Case Name	Associated Web Service API Methods

01.03.01	Record Voter Registration Information through EMS	**Direct**
		`SearchStateVoters`
		`UpdateStateVoter`

01.11.01	Initiate Voter Registration and Run Local EMS Search	**Direct**
		`VerifyVoterIdentification`
		`SearchStateVoters`

01.11.02	Transfer Out through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetTransferOutWorkItems` 3. `UpdateTransferOutWorkItems`
01.11.04	Enter Declined Voter through EMS	**Direct** `UpdateStateVoter`
01.13.01	Check for Duplicate Driver License or State ID Rejections	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` 2. `UpdateStateVoterStatusUpdatedMessages`
01.14.01	Respond to Notice of Data Deficiency	**Direct** ***Work Item API Call Sequence*** 1. `UpdateStateVoter` 2. `GetOpenWorkItemList` 3. `GetDataValidationErrorWorkItems` 4. `UpdateDataValidationErrorWorkItems`
01.19.01	Handle Local Voter Registration Contingency	**Direct** All Update, Message, and Work Item API Methods

01.21.01	Run Pre-Registration Check	**Direct**
		`VerifyVoterIdentification`
		`SearchStateVoters`

01.22.01	Assign Precinct through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDataValidationErrorWorkItems` 3. `UpdateStateVoter`

Use Case Number	Use Case Name	Associated Web Service API Methods
01.23.01	Process Unprecinctable Work Item through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDataValidationErrorWorkItems` 3. `UpdateStateVoter`
01.24.01	State Update Record (Register, Update, Transfer)	**Direct** `UpdateStateVoter`

01.25.01	Process IDV Job	**Direct**
		`VerifyVoterIdentification`
		`UpdateStateVoter`

01.26.01	Process VoteCal Validations and Store VR Information	**Direct** `UpdateStateVoter` **Auxiliary** *All Message and Work Item Methods*
02.02.01	Search Voter through EMS Interface	**Direct** `SearchStateVoters`
02.04.01	Retrieve Voter Details through Web Service	**Direct** `GetStateVoters`
03.01.01	Process CDPH Death Records File	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` 2. `UpdateStateVoterStatusUpdatedMessages`
03.02.02	Accept or Reject Death Record Match Case through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDeathRecordMatchCaseWorkItems` 3. `UpdateDeathRecordMatchCaseWorkItems` 4. `GetOpenMessages` 5. `UpdateStateVoterStatusUpdatedMessages`
03.03.02	Undo Death Record Match Case through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDeathRecordMatchCaseWorkItems` 3. `UpdateDeathRecordMatchCaseWorkItems` 4. `GetOpenMessages` 5. `UpdateStateVoterStatusUpdatedMessages` ***Direct Update*** `UpdateStateVoter`

Use Case Number	Use Case Name	Associated Web Service API Methods
03.11.01	Process CDCR Felon Records File	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` 2. `UpdateStateVoterStatusUpdatedMessages`
03.12.02	Accept or Reject CDCR Felon Record Match Case through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetFelonRecordMatchCaseWorkItems` 3. `UpdateFelonRecordMatchCaseWorkItems`
03.13.02	Undo CDCR Felon Match Case through EMS	**Direct** **Work Item *API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetFelonRecordMatchCaseWorkItems` 3. `UpdateFelonRecordMatchCaseWorkItems` ***Direct Update*** `UpdateStateVoter`
03.21.01	Process DMV COA Record File	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` *2. Can process:* `UpdateStateVoter` 3. *Cannot process:* `UpdateDMVChangeOfAddressRecordReceivedMessages`
03.22.01	Process DMV COA Work Item through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDMVChangeOfAddressRecordMatchCaseWorkItems` 3. `UpdateDMVChangeOfAddressRecordMatchCaseWorkItems` 4. `UpdateStateVoter`
03.32.01	Process NCOA Record File	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` *2. Can process:* `UpdateStateVoter` 3. *Cannot process:* `UpdateNationalChangeOfAddressRecordReceivedMessages`
03.33.01	Process NCOA Work Item through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetNationalChangeOfAddressRecordMatchCaseWorkItems` 3. `UpdateNationalChangeOfAddressRecordMatchCaseWorkItems` 4. `UpdateStateVoter`
03.41.01	Process Duplicate Voter Detection Job	**Direct** ***Message API Call Sequence*** 1. `GetOpenMessages` 2. `UpdateStateVoterStatusUpdatedMessages`

Use Case Number	Use Case Name	Associated Web Service API Methods
03.42.02	Accept or Reject Duplicate Voter Match Case through EMS	**Direct** ***Work Item API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDuplicateRecordMatchCaseWorkItems` 3. `UpdateDuplicateRecordMatchCaseWorkItems`
03.43.02	Undo Accepted Duplicate Voter Match Case through EMS	**Direct** **Work Item *API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDuplicateRecordMatchCaseWorkItems` 3. `UpdateDuplicateRecordMatchCaseWorkItems` ***Direct Update*** `UpdateStateVoter`
03.45.01	Create Duplicate Match Case Manually	**Direct** *Change request awaiting approval.*
03.50.01	Undo Transfer Out through EMS	**Direct** **Work Item *API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDuplicateRecordMatchCaseWorkItems` 3. `UpdateDuplicateRecordMatchCaseWorkItems` ***Direct Update*** `UpdateStateVoter`
03.51.02	Review Unresolved List Maintenance Issues through EMS	**Direct** `GetOpenWorkItemList`

04.06.01	EMS Processes Message Queue	**Direct**
		`GetOpenMessages`
		`UpdateStateVoterStatusUpdatedMessages`
		`UpdateStateVoterUpdatedMessages`
		`UpdateStateVoterActivityCreatedMessages`
		`UpdateNewVoterReceivedMessages`
		`UpdateDMVChangeOfAddressRecordReceivedMessages`
		`UpdateNationalChangeOfAddressRecordReceivedMessages`
		`UpdateStateVoter`

04.19.01	Create or Update Vote-by-Mail Record	**Direct** `UpdateStateVoterVoteByMailActivity`
04.20.01	Process DMV Signature Request	**Direct** ***New Voter Message API Call Sequence*** 1. `GetOpenMessages` 2. `UpdateNewVoterReceivedMessages` ***Get Voter API Call Sequence*** 1. `GetStateVoters`

Use Case Number	Use Case Name	Associated Web Service API Methods

07.18.01	Generate Report or Correspondence	**Auxiliary**
		`GetOpenMessages`
		`UpdateStateVoterUpdatedMessages`

08.01.01	Review Orphaned Voters	**Direct** **Work Item *API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDataValidationErrorWorkItems` 3. `UpdateDataValidationErrorWorkItems` 4. `UpdateStateVoter`
08.11.01	Review Orphaned Precincts	**Direct** **Work Item *API Call Sequence*** 1. `GetOpenWorkItemList` 2. `GetDataValidationErrorWorkItems` 3. `UpdateDataValidationErrorWorkItems`

2.2.3.4 Acknowledging Messages

Various points to keep in mind when interfacing with the Web Service API are detailed in this section.

Authentication

The first parameter of every Web Service API method is a countyEMSAuth object. In cases where communication with the Web Service API is initiated by automatic processes occurring within the EMS (e.g. the job processing service making calls to `GetOpenMessages`)the username and password credentials provided by SOS staff for the county EMS should be supplied in the countyEMSAuth object. In cases where communication with the API is initiated by some county elections staff user action (e.g. updating the status of a work item) the username and password credentials supplied by SOS staff for the county elections staff user initiating the action should be supplied in the countyEMSAuth object.

Updating the Status of Messages

Messages have two statuses: An overall open/closed status indicating whether or not the Message has been handled by the county EMS; and a Message type specific status indicating how the Message was handled. Though the update Web Service API methods corresponding to each Message type accept the entire message object as a parameter, only the message type specific status property of the parameter object is examined and updated by VoteCal. If the message type specific status indicates that the overall message status should be changed from open to closed, VoteCal will perform that update as well. The update methods for each corresponding Message type return the Message as updated in the response. The EMS job processing service may examine the updated Message object returned in the update call response to validate that VoteCal has updated the Message as expected.

Updating the Status of Work Items

As is the case with Messages, Work Items have two statuses: An overall open/closed status indicating whether or not the Work Item has been handled by the county elections staff (through the county EMS), and a Work Item type specific status indicating how the Work Item was handled. Though the update Web Service API methods corresponding to each Work Item type accept the entire Work Item object as a parameter, only the Work Item type specific status property of the parameter object is examined and updated by VoteCal. If the Work Item type specific status indicates that the overall Work Item status should be changed from open to closed, VoteCal will perform that update as well. The update methods for each corresponding Work Item type return the Work Item as updated in the response. The county EMS may examine the updated Work Item object returned in the update call response to validate that VoteCal has updated the Work Item as expected.

Translation of Codes

As specified by VoteCal requirements, certain standard codes (e.g. County Code, Registration Source Code, Registration Method Code, etc.) must be able to be modified by SOS VoteCal administrators. The net result of this requirement is that where an enumeration might be used for standard code based properties of objects in the Web Service API, string properties are used instead. The compiled help Web Service API Reference documents those cases where a string property of an object must map to a standard code in the description for the property. Modifications to any and all standard codes by SOS VoteCal administrators will be communicated to EMS vendors prior to the modification.

Use of Base 64 Encoding for Binary Data

Since the XML SOAP protocol does not natively support binary data, all Web Service API object properties representing binary data (Affidavit Signatures, etc.) will be represented as strings of Base 64 encoded binary data. The compiled help Web Service API Reference documents those cases where a string property should be a Base 64 representation of binary data. More information on Base 64 encoding can be found here: http://en.wikipedia.org/wiki/Base64

2.2.4 Development and Testing Methodology and Resources

EMS Vendors will be responsible for the remediation of their systems in accordance with the terms of the contract entered into with the Secretary of State's office. Catalyst resources will remain available for consultation throughout the development process, and feedback on methods and data will be considered as development progresses.

2.2.4.1 Initial EMS Software Development

Initially, the vendor will work within the vendor's own software development environment on its products and rely on this specification document. Catalyst will be developing the VoteCal system in parallel.

The vendors' software developers will be able to use the attached WSDL file within their chosen integrated development environment to automate much of the work which must be done to encapsulate the communication protocols and data formats shared between the systems. Because the WSDL file is machine generated from the same set of stub VoteCal Web services being built by Catalyst, the two development teams may proceed with confidence that they are building around an identical representation of the APIs.

Catalyst will be developing a sample Web Service API client application containing unit tests to test and validate all functionality and logic rules of the Web Service API. At the request of an EMS vendor, the source code for the sample client application will be provided for reference purposes.

2.2.4.2 Continued EMS Software Development Using Sandbox

At a later stage of development, a VoteCal Sandbox Environment will be provided based on a minimal set of VoteCal system functionality and test data. The vendor may connect their prototype EMS to this prototype version of the VoteCal system to help confirm the course of their software development efforts through preliminary integration testing.

From the sandbox, the vendor will have access to self-test features such as data validation reports and user interface features for execution of specific test cases. A separate environment will be established for each vendor, and a fourth environment will also be established for interactivity and validation of cross-jurisdictional functionality.

In order to ensure data security while still allowing for effective development, each vendor will need to provide Catalyst with a set of valid addresses from within one of their customer jurisdictions. Using these addresses, Catalyst will create artificial voter records that can be precincted within each vendor's jurisdiction but pose no threat to individual information and return these to the vendor for development and testing purposes. The size of this data set must remain manageable, yet provide sufficient data for the vendors to develop against. The initial recommendation is that each vendor provides 100 valid addresses, though this may be subject to specific development requirements.

2.2.4.3 Remediated EMS Testing

The purpose of the remediated EMS testing process is to establish adherence to the specifications that the EMS vendors are required to implement prior to user acceptance testing (UAT) and pilot deployment.

Prior to this testing, Catalyst will have utilized a model EMS software application to simulate relevant expected behavior of an integrated EMS system. The purpose of this testing is to isolate errors that might exist in VoteCal.

Remediated EMS testing processes includes validation of data standards, API usage, and batch transfer processes through the EMS implemented database views.

The tests will be configured to run in a test environment to be established for the vendor by one of their chosen pilot counties. The remediated EMS will be deployed and configured in this environment with a remediated database adapted from a branch copy of the county's production data.

The EMS vendors' personnel will execute test procedures developed by Catalyst Consulting Group as part of Deliverable 3.3. For estimating purposes, there will be at least one test procedure per normal and alternate flow of each use case. Catalyst will coordinate and monitor the test execution. Some tests may require connecting the EMS product to the VoteCal production environment and then back to the test environment.

Catalyst will be responsible for issuing a certification report after testing is complete.

2.2.4.4 User Acceptance Testing

The UAT is performed after remediated EMS testing and prior to deployment in a pilot environment. This testing is expected to take place in a test environment in a selected pilot county.

The SOS will perform tests that duplicate those performed by Catalyst during remediated EMS testing. This testing will be conducted independently from Catalyst.

It will be the responsibility of the counties and EMS vendors to perform user regression testing to ensure that modifications made to the EMS for VoteCal have not negatively impacted other non-VoteCal functionality within the EMS.

2.2.4.5 Pilot Testing

The EMS vendors will assist SOS in the selection of pilot county sites.

After SOS has concluded UAT, each EMS vendor will work with Catalyst to transition the pilot county from the test environment to the production environment. This process will be documented in *Deliverable 3.9 VoteCal System Data Conversion and Data Integration Plan.*

As part of the pilot testing phase, EMS vendors will also implement their proposed training plan. This will give the EMS vendor an opportunity to fine tune the training plan prior to statewide system rollout as well as provide the SOS an opportunity to evaluate the plan.

Catalyst will provide a highly optimized support environment during pilot testing so that all issues are responded to in as expedited a manner as possible. The EMS vendors are expected to provide the same level of support for their product in pilot testing as well.

The pilot tests are currently scheduled to run through a live election cycle (June 2011 UDELs).

At the end of the pilot testing phase there will be an opportunity to implement software changes and refine the conversion and deployment. Additional testing may be required based on the scope of software changes made.

2.2.5 Transaction Timing

Transaction timing requirements are detailed in the following table:

Transaction Type	Timing Requirement
Single call to any Web Service API method	Response returned within 500ms

Transaction Type	Timing Requirement
Single call to `UpdateStateVoter` Web Service API method	Response returned within 1000ms The IDV interface round trip response time will be added to this in some cases.
Routine calls to `GetOpenMessages`	Approximately every 10 seconds.
Call to UpdateStateVoter after an update to a voter has occurred in the county EMS database	Within 20 seconds of voter data update to EMS database
VoteCal Data Exchange Batch Service data transfer	To occur nightly or less frequently as scheduled through controls in the VoteCal user interface.

To increase speed and efficiency, Web Service API calls may be made in parallel by establishing multiple simultaneous connections to the Web Service API on multiple threads. For example, based on the timing requirements it is assumed that a single process can achieve 1 `UpdateStateVoter` transaction per second when executing calls serially. Using 5 threads to execute Web Service API calls in parallel, it is feasible to achieve 5 transactions per second. Limits to the number of simultaneous Web Service API call connections will be provided to EMS vendors on a county-by-county basis as determined by the size of the county.

2.2.6 Security

All connections to the Web Service API Sandbox, testing, training, and production environments will be secured using SSL over an HTTPS connection on port 443. The SSL connection to the non-production environments will utilize a self signed x.509 certificate. The SSL connection to the production environment will utilize an x.509 certificate provided by VeriSign.

Authentication of county EMS systems and county elections staff users (through the county EMS system) to the Web Service API utilizes Active Directory user accounts and roles administered by SOS staff. Please refer to prior section ***Authentication*** for additional information on Web Service API call authentication.

The remediated EMS system will have embedded credentials for accessing VoteCal Web Service APIs. The EMS system must only store these credentials in an encrypted form.

The VoteCal Data Exchange Service will also communicate with a controlling VoteCal Web service API using HTTPS on Port 443. This software will stream batch files to and from VoteCal using a secure file transfer protocol over a custom configurable port number.

In addition to security requirements related to communications, the functionality of VoteCal requires that confidential voter records be maintained at the county level in order to ensure that VoteCal retains a complete listing of all eligible voters in California. This may require that vendors implement database level encryption at either a row or table level within each local database to protect the confidentiality of these registrant records. In addition, logic will need to be incorporated based on existing statutes to manage the confidential status of these records.

2.2.7 Network Connectivity

The Production Web Service API and VoteCal Data Exchange Interface will be hosted at primary and secondary failover sites. An SOS managed MPLS WAN will connect all county sites to the primary and secondary data center in order to facilitate county connectivity to the VoteCal production infrastructure located at the SOS primary data center site and the Office of Technology (OTech) secondary data center site.

Each county on the MPLS cloud should be provided with a /29 subnet. This subnet allocation will provide each county site with six usable IP addresses to be used for the following functions:

- IP 1—Production Web services & VoteCal Batch Data Exchange Service.
- IP 2—Testing Web services & VoteCal Batch Data Exchange Service.
- IP 3—Training Web services & VoteCal Batch Data Exchange Service.
- IP 4—Available for future use.
- IP 5—Available for future use.
- IP 6—Available for future use.

Additionally, access to the sandbox (Integration Testing) Web Service API environment from within the development offices of EMS vendors will be provided by a means to be determined by SOS.

3.0 Appendices

3.1 Appendix A – Standardized Code Tables

This appendix provides representative values of standard code values for common transactions communicated between Local EMS systems and VoteCal. These values are subject to change based on business need and must remain flexible and configurable. Tables A-1 thru A-12 were adapted from existing CalVoter standards where applicable to VoteCal. Tables A-13 and higher were devised based on feedback from EMS vendors to represent existing standards used in California.

Table A-1 Jurisdiction Codes

County Code	County Name
01	Alameda
02	Alpine
03	Amador
04	Butte
05	Calaveras
06	Colusa
07	Contra Costa
08	Del Norte
09	El Dorado
10	Fresno
12	Humboldt
13	Imperial
14	Inyo
15	Kern
16	Kings
17	Lake
18	Lassen
19	Los Angeles
20	Madera
21	Marin
22	Mariposa
23	Mendocino
24	Merced
25	Modoc
26	Mono
27	Monterey
28	Napa
29	Nevada
30	Orange
31	Placer
32	Plumas
33	Riverside
34	Sacramento
35	San Benito
36	San Bernardino
37	San Diego
38	San Francisco
39	San Joaquin
40	San Luis Obispo
41	San Mateo
42	Santa Barbara
43	Santa Clara
44	Santa Cruz
45	Shasta
46	Sierra
47	Siskiyou
48	Solano
49	Sonoma
50	Stanislaus
51	Sutter
52	Tehama
53	Trinity
54	Tulare
55	Tuolumne
56	Ventura
57	Yolo
58	Yuba

Table A-2 Registration Source Codes

Registration Source Code	Registration Source Description
DMV	Department of Motor Vehicles
PAA	Public Assistance Agencies
DSO	State-assisted Disability Service Organizations
AFR	Armed Forces Recruiting Centers
ODA	Other Designated Agencies not listed above
MAI	Others received by mail and not included above
UNK	Unknown
DMV	Department of Motor Vehicles
WEB	Online Voter Registration (This code was not present in the original CalVoter standards)

Table A-3 Registration Method Codes

Registration Method Code	Registration Method Description
D	DMV
I	In Person
M	Mail (must have a postmark)
R	Registration Drive
O	Other social service agencies
U	Unknown or Secretary of State
W	Online Voter Registration (This code was not present in the original CalVoter standards)

Table A-4 Language Codes

Language Preference Code	Language Name
null	English or no preference
C	Chinese
J	Japanese
K	Korean

Language Preference Code	Language Name
S	Spanish
T	Tagalog
V	Vietnamese

Table A-5 Place of Birth Codes

Place of Birth Code	Place of Birth Description
AA	Australia
AB	Albania
AC	Africa
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AI	Anguilla
AK	Alaska
AL	Alabama
AM	Armenia
AO	Angola
AQ	Antarctica
AR	Arkansas
AS	American Samoa
AT	Argentina
AU	Austria
AW	Aruba
AX	Aland Islands
AZ	Arizona
BA	Bosnia & Herzegovina
BB	Barbados
BD	Bangladesh
BE	Bouvet Island
BF	Burkina Faso
BG	Belgium
BH	Bahrain
BI	Burundi
BJ	Benin
BM	Bermuda
BN	Brunei Darussalam
BO	British Indian Ocean Territory
BR	Brazil
BS	Bahamas
BT	Bhutan
BU	Bulgaria
BV	Bolivia
BW	Botswana
BY	Belarus
BZ	Belize
CA	California
CB	Colombia
CC	Cocos (Keeling) Islands
CD	Canada
CE	Cameroon
CF	Central African Republic
CG	Congo
CI	Côte D'Ivoire
CK	Cook Islands
CM	Cambodia
CN	China
CO	Colorado
CQ	Chile
CR	Costa Rica
CS	Serbia and Montenegro
CT	Connecticut
CU	Cuba
CV	Cape Verde
CX	Christmas Island
CY	Cyprus
CZ	Czechoslovakia/Czech Republic
DC	District of Columbia
DE	Delaware
DJ	Djibouti
DK	Denmark
DM	Dominica

Place of Birth Code	Place of Birth Description
DR	Dominican Republic
DZ	Algeria
EE	Estonia
EM	East Germany (Former)
EN	England
EO	Ethiopia
ER	Eritrea
EU	Ecuador
EY	Egypt
FJ	Fiji
FK	Falkland Islands
FL	Florida
FM	Federated States of Micronesia
FN	Finland
FO	Faroe Islands
FOR	foreign born
FR	France
GA	Georgia
GB	Gabon
GC	Greece
GD	Grenada
GE	Germany
GF	French Guiana
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GO	Georgia (former USSR Republic)
GP	Guadeloupe
GQ	Equatorial Guinea
GS	South Georgia and the South Sandwich Islands
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HI	Hawaii
HK	Hong Kong
HM	Heard Island & McDonald Islands
HN	Honduras
HR	Croatia
HT	Haiti
HU	Hungary

Place of Birth Code	Place of Birth Description
IA	Iowa
IC	Iceland
ID	Idaho
IE	Ireland
II	India
IL	Illinois
IN	Indiana
IO	Indonesia
IQ	Iraq
IR	Iran
IS	Israel
IT	Italy
JA	Japan
JM	Jamaica
JO	Jordan
KE	Kenya
KG	Kyrgyzstan
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis
KP	Korea, Democratic People's Republic of
KR	South Korea
KS	Kansas
KW	Kuwait
KY	Kentucky
KZ	Kazakhstan
LA	Louisiana
LB	Lebanon
LC	Saint Lucia
LE	Liberia
LI	Liechtenstein
LK	Sri Lanka
LO	Lao People's Democratic Republic
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libyan Arab Jamahiriya
MA	Massachusetts
MB	Morocco
MC	Monaco
MD	Maryland
ME	Maine

Place of Birth Code	Place of Birth Description
MF	Macao
MG	Madagascar
MH	Marshall Islands
MI	Michigan
MJ	Malta
MK	Macedonia, The Former Yugoslav Republic
ML	Mali
MM	Myanmar, Burma
MN	Minnesota
MO	Missouri
MP	Mongolia
MQ	Martinique
MR	Mauritania
MS	Mississippi
MT	Montana
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NC	North Carolina
ND	North Dakota
NE	Nebraska
NF	Norfolk Island
NG	Niger
NH	New Hampshire
NI	Northern Mariana Islands
NJ	New Jersey
NL	Netherlands Antilles
NM	New Mexico
NP	Nepal
NR	Nauru
NT	Netherlands
NU	Nicaragua
NV	Nevada
NW	Norway
NY	New York
NZ	New Zealand
OH	Ohio
OK	Oklahoma
OM	Oman

Place of Birth Code	Place of Birth Description
OR	Oregon
PA	Pennsylvania
PF	French Polynesia
PI	Philippines
PK	Pakistan
PM	Panama
PN	Pitcairn
PO	Poland
PP	Papua New Guinea
PR	Puerto Rico
PS	Palestine Territory, Occupied
PT	Portugal
PU	Peru
PV	Paraguay
PW	Palau
QA	Qatar
RE	Reunion
RH	Rhodesia
RI	Rhode Island
RM	Moldova, Republic of
RO	Romania
RU	Russia
RW	Rwanda
SA	Saudi Arabia
SB	Solomon Islands
SC	South Carolina
SD	South Dakota
SE	Sweden
SF	Seychelles
SG	Singapore
SH	Saint Helena
SI	Slovenia
SJ	Svalbard and Jan Mayen
SK	Slovakia
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SP	Spain
SQ	Saint Pierre and Miquelon
SR	Suriname
ST	Sao Tome and Principe
SU	Sudan
SV	El Salvador

Place of Birth Code	Place of Birth Description
SW	Swaziland
SY	Syrian Arab Republic
SZ	Switzerland
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau
TL	Timor-Leste
TM	Turkmenistan
TN	Tennessee
TO	Tonga
TR	Turkey
TT	Trinidad and Tobago
TU	Tunisia
TV	Tuvalu
TW	Taiwan, Province of China
TX	Texas
TZ	Tanzania, United Republic of
UA	Ukraine
UG	Uganda
UK	United Kingdom
UM	US Minor Outlying Islands (Wake, Midway, etc.)
UNK	unknown
UR	USSR (Former)
US	United States
UT	Utah
UY	Uruguay
UZ	Uzbekistan
VA	Virginia

Place of Birth Code	Place of Birth Description
VC	Saint Vincent and the Grenadines
VG	Virgin Islands, British
VI	Virgin Islands, U.S.
VM	Vietnam
VT	Vermont
VU	Vanuatu
VZ	Venezuela
WA	Washington
WF	Wallis and Futuna
WG	West Germany (Former)
WI	Wisconsin
WN	West Indies
WS	Western Sahara
WV	West Virginia
WY	Wyoming
XC	Congo, The Democratic Republic of
XG	Nigeria
XI	Cayman Islands
XM	Montserrat
XN	Niue
XS	Samoa
YE	Yemen
YG	Yugoslavia
YS	Scotland
YT	Mayotte
YW	Wales
ZA	South Africa
ZC	New Caledonia
ZM	Zambia
ZW	Zimbabwe

Table A-6 Political Parties Table

Party Code	Party Name
DEM	Democratic
REP	Republican
AI	American Independent
CAA	Conscious African American
EGA	Egalitarian
HUM	Humanitarian
PF	Peace and Freedom
LIB	Libertarian
GRN	Green Party

Table A-7 Election Type Codes

Election Code	Election Description
PP	Presidential Primary
PG	Presidential General
GP	Gubernatorial Primary
GG	Gubernatorial General
SS	Special Statewide
CP	Congressional District Special Primary
CG	Congressional District Special General
SP	Other Legislative District Special Primary
SG	Other Legislative District Special General
RC	Recall
IN	Initiative
RE	Referendum
CO	Consolidated
GD	General District
SB	School Bond
ST	Special Tax

Table A-8 Gender Codes

Sex Code	Sex Description
M	Male
F	Female
<null>	Unknown

Table A-9 ID Requirements Codes

Code	Description
Y	Voter is required to show ID the first time he or she votes in a Federal election
A	Registrant has previously voted in a federal election in California
B	Registrant submitted copy of identification with their registration application
C	Registrant's ID was verified against DMV or SSA records
D	Registrant is entitled to vote absentee under the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA)
E	Registrant is entitled to vote otherwise than in person under the Voting Accessibility for the Elderly and Handicapped Act (VAEHA)
F	Registrant is entitled to vote otherwise than in person under other federal law
G	Voter has provided ID prior to voting
null	Registrant has not registered by mail

Table A-10 Name Suffix Codes

Name Suffix Code	Name Suffix
JR	Junior
SR	Senior
I	1st
II	2nd
III	3rd
IV	4th
V	5th
VI	6th
VII	7th

Name Suffix Code	Name Suffix
VIII	8th
IX	9th
X	10th
null	Blank or none

Table A-11 State Code

State Code	State Name
AK	Alaska
AL	Alabama
AR	Arkansas
AS	American Samoa
AZ	Arizona
CA	California
CO	Colorado
CT	Connecticut
DC	District of Columbia
DE	Delaware
FL	Florida
FM	Federated States of Micronesia
GA	Georgia
GU	Guam
HI	Hawaii
IA	Iowa
ID	Idaho
IL	Illinois
IN	Indiana
KS	Kansas
KY	Kentucky
LA	Louisiana
MA	Massachusetts
MD	Maryland
ME	Maine
MH	Marshall Islands
MI	Michigan
MN	Minnesota
MO	Missouri
MS	Mississippi
MT	Montana
NE	Nebraska
NH	New Hampshire
NV	Nevada
NY	New York
NJ	New Jersey
NM	New Mexico
NC	North Carolina
ND	North Dakota
MP	Northern Mariana Islands
OK	Oklahoma
OH	Ohio
PW	Palau
OR	Oregon
PA	Pennsylvania
PR	Puerto Rico
RI	Rhode Island
TX	Texas
SC	South Carolina
SD	South Dakota
TN	Tennessee
UT	Utah
VT	Vermont
VA	Virginia
WI	Wisconsin
WV	West Virginia

State Code	State Name
VI	Virgin Islands, U.S.
WA	Washington
WY	Wyoming
AA	Armed Forces of the Americas
AE	Armed Forces Europe

State Code	State Name
AP	Armed Forces Pacific
null	Unknown

Table A-12 Street Directions

Street Direction Code	Street Direction
N	North
S	South
E	East
W	West
NE	Northeast
NW	Northwest
SE	Southeast
SW	Southwest
null	Blank/None

Table A-13 Street Suffix Codes

Code	Street Suffix
null	no value
ALY	Alley
ANX	Annex
APN	Assessor's Parcel Number
ARC	Arcade
AREA	Area
AVE	Avenue
BCH	Beach
BG	Burg
BGS	Burgs
BLF	Bluff
BLFS	Bluffs
BLVD	Boulevard
BND	Bend
BR	Branch
BRG	Bridge
BRK	Brook
BRKS	Brooks
BTM	Bottom
BYP	Bypass
BYU	Bayou
BYWY	Byway
CIR	Circle
CIRS	Circles
CLB	Club
CLF	Cliff
CLFS	Cliffs
CMN	Common
CMNS	Commons
COR	Corner
CORS	Corners
CP	Camp
CPE	Cape
CRES	Crescent
CRK	Creek
CRSE	Course
CRST	Crest
CSWY	Causeway
CT	Court
CTR	Center
CTRS	Centers
CTS	Courts
CURV	Curve
CV	Cove
CVS	Coves
CYN	Canyon

Code	Street Suffix
DL	Dale
DM	Dam
DR	Drive
DRS	Drives
DV	Divide
EST	Estate
ESTS	Estates
EXPY	Expressway
EXT	Extension
EXTS	Extensions
FALL	Fall
FLD	Field
FLDS	Fields
FLS	Falls
FLT	Flat
FLTS	Flats
FRD	Ford
FRDS	Fords
FRG	Forge
FRGS	Forges
FRK	Fork
FRKS	Forks
FRST	Forest
FRY	Ferry
FT	Fort
FWY	Freeway
GDN	Garden
GDNS	Gardens
GLN	Glen
GLNS	Glens
GRD	Grade
GRN	Green
GRNS	Greens
GRV	Grove
GRVS	Groves
GTWY	Gateway
HBR	Harbor
HBRS	Harbors
HL	Hill
HLS	Hills
HOLW	Hollow
HTS	Heights
HVN	Haven
HWY	Highway
INLT	Inlet

Code	Street Suffix
IS	Island
ISLE	Isle
ISS	Islands
JCT	Junction
JCTS	Junctions
KNL	Knoll
KNLS	Knolls
KY	Key
KYS	Keys
LAND	Land
LCK	Lock
LCKS	Locks
LDG	Lodge
LF	Loaf
LGT	Light
LGTS	Lights
LK	Lake
LKS	Lakes
LN	Lane
LNDG	Landing
LOOP	Loop
MALL	Mall
MDW	Meadow
MDWS	Meadows
MEWS	Mews
ML	Mill
MLS	Mills
MNR	Manor
MNRS	Manors
MSN	Mission
MT	Mount
MTN	Mountain
MTNS	Mountains
MTWY	Motorway
NCK	Neck
OPAS	Overpass
ORCH	Orchard
OVAL	Oval
PARK	Park
PARK	Parks
PASS	Pass
PATH	Path
PIKE	Pike
PKWY	Parkway
PKWY	Parkways

Code	Street Suffix
PL	Place
PLN	Plain
PLNS	Plains
PLZ	Plaza
PNE	Pine
PNES	Pines
PR	Prairie
PRT	Port
PRTS	Ports
PSGE	Passage
PT	Point
PTS	Points
RADL	Radial
RAMP	Ramp
RD	Road
RDG	Ridge
RDGS	Ridges
RDS	Roads
RIV	River
RNCH	Ranch
ROW	Row
RPD	Rapid
RPDS	Rapids
RST	Rest
RTE	Route
RUE	Rue
RUN	Run
SHL	Shoal
SHLS	Shoals
SHR	Shore
SHRS	Shores
SKWY	Skyway
SMT	Summit
SPG	Spring
SPGS	Springs
SPUR	Spur
SPUR	Spurs
SQ	Square
XRDS	Crossroads

Code	Street Suffix
SQS	Squares
ST	Street
STA	Station
STRA	Stravenue
STRM	Stream
STS	Streets
TER	Terrace
TPKE	Turnpike
TRAK	Track
TRCE	Trace
TRFY	Trafficway
TRL	Trail
TRLR	Trailer
TRWY	Throughway
TT	Truck Trail
TUNL	Tunnel
UN	Union
UNS	Unions
UPAS	Underpass
VIA	Viaduct
VIS	Vista
VL	Ville
VLG	Village
VLGS	Villages
VLY	Valley
VLYS	Valleys
VW	View
VWS	Views
WALK	Walk
WALK	Walks
WALL	Wall
WAY	Way
WAYS	Ways
WL	Well
WLS	Wells
XING	Crossing
XRD	Crossroad

Table A-14 Voting Methods

Code	Voting Method
EV	Early Voting by Personal Appearance
VM	Vote By Mail
PP	Polling Place on Election Day
PR	Provisional

Table A-15 Provisional Ballot Rejection Reasons

Code	Description
ABS	absentee Requested
EE	Envelope Empty
ID	No ID Provided
INC	Incomplete
IPP	Incorrect Party-Prim
NR	Not Registered
WP	Voted in Wrong Precinct
NV	New Voter Past Close of Registration
CV	Change on Existing Voter

Table A-16 Vote-By-Mail Categories

Code	Description
EV	Early Voter
IP	In Person
MI	Military
MO	Military Overseas
OV	Overseas Voter
PA	Permanent Absentee
ST	Student
VM	Vote-By-Mail Precinct
SB	Submarine
HO	Hospital

Table A-17 Vote-By-Mail Application Methods

Code	Description
OF	Office/Counter
EV	Early Voting
ML	Mail
SB	Sample Ballot Cover
ST	State Form

Table A-18 Vote-By-Mail Application Rejection Reason Codes

Code	Description
UNR	Unregistered/Late Registration
SIG	No Signature/Signature Mismatch
ADR	No Address Provided
DUP	Duplicate Application
LAT	Late Application

Table A-19 Vote-By-Mail Transmission Methods

Code	Description
M	Mail
F	Fax
E	Email

Table A-20 Vote-By-Mail Return Methods

Code	Description
M	Mail
O	Office
P	In-Person

Table A-21 Vote-By-Mail Voting Methods

Code	Description
C	County Issued Vote-By-Mail
F	Federal Write-In Vote-By-Mail

Table A-22 Vote-By-Mail Rejection Codes

Code	Description
LT	Ballot Returned Late
CA	Cancelled At Voter Request
MO	Voter Moved Out of County
DC	Voter Deceased
SG	Signature Mismatch
RB	Replacement Original Spoiled
SP	Surrendered at Poll
UD	Undeliverable (Returned by P.O.)
NS	No Signature
FS	Failsafe Challenge
IV	Inactive Voter
PS	Printed Signature
DU	Duplicate Ballot

Table A-23 District Type Codes

Code	Description
BED	Board of Education
BEQ	Board of Equalization
CIT	City
CCL	Community College
CSD	Community Service District
CNG	Congress
SUP	County Supervisor
CTY	Countywide
CRT	Court
FIR	Fire
HOS	Hospital District
IRR	Irrigation
LIB	Library
LSD	Local School District

PKR	Parks and Recreation
PUD	Public Utilities District
RCD	Resource Conservation Dist
SAN	Sanitary
STA	State Assembly
SEN	State Senate
WRD	Ward
WTR	Water

Table A-24 Document/Image Types

Code	Description
A	Voter Affidavit/Application
S	Signature
V	Vote-By-Mail Application
C	Correspondence
P	Picture
O	Other

Table A-25 Document/Image Format

Code	Description
G	GIF
J	JPEG/JPG
P	PDF
T	TIF

Table A-26 Voter Status Codes

Code	Description
A	Active
C	Cancelled
D	Declined
I	Inactive
P	Pending

Table A-27 Voter Status Reason Codes

Voter Status	Reason Code	Description
A	CON	Confirmation
A	AREG	Re-Registered
A	TRNS	Transfer
A	T18	Turned 18
A	ADDR	Address Change
A	PROV	Apply Provisional
A	CORR	Correct Deficiency

A	PET	Activate Inactive Petition Signer
A	MAIL	Mailing Address Update
A	NAM	Name Change
A	PERM	Permanent Absentee Request
A	PEND	Pending Complete
A	REAC	Reactivate
A	VOTE	Voted
C	CO	Conservatorship Order
C	CON	Confirmation
C	DEC	Voter Deceased
C	DUP	Duplicate Registration
C	FEL	Felon
C	MOVC	Moved Out of County
C	OTH	Other
C	PUR	Purged
C	PUR2	Inactive For 2 Fed Gen Elections
C	CREG	Re-Registered
C	VREQ	Voter Requested Cancellation
C	MGIC	Merged In-County
C	MGOC	Merged Out-of-County
C	MOVS	Moved Out-of-State
C	NVOT	No Response – Failure to Vote
I	ARES	Alt. Residency Confirmation
I	MPU	Mail Precinct Undeliverable
I	UND	Undeliverable Mail
I	RETB	Returned Ballot
P	U18	Under 18
P	DOB	Date of Birth Missing
P	PINC	Incomplete
P	STX	Street Exception
P	CONF	Confirm or Provide ID
D	NCIT	Non-Citizen
D	DINC	Incomplete

Table A-28 Voter Assistance Codes

Code	Description
B	Blind (Audio Vote)
G	General Assistance Requested
H	Hearing Impaired
V	Visually Impaired (Large Type)
P	Paraplegic
Q	Quadriplegic

Table A-29 UOCAVA Categories

Code	Description
DM	Domestic Military
OC	Overseas Civilian
OM	Overseas Military

Table A-30 Unit Type Codes

Code	Description
APT	Apartment
BLDG	Building
BSMT	Basement
DEPT	Department
FL	Floor
FRNT	Front
HNGR	Hangar
LOT	Lot
LOWR	Lower
OFC	Office
PH	Penthouse
PIER	Pier
REAR	Rear
RM	Room
SIDE	Side

SLIP	Slip
SPC	Space
STE	Suite
STOP	Stop
TRLR	Trailer
UNIT	Unit
UPPR	Upper

Table A-31 Event Codes

Code	Description
ADDR	Address change
CONF	Confidential Status
PREC	Home Precinct
NREG	New Registration
PTY	Party change
REGM	Registration Method
REGS	Registration Status
UOCAVA	UOCAVA code
VCON	Voter Contacts (mailing, phone calls, etc)

Table A-32 Voter Confidentiality Codes

Code	Description
C	Court Ordered
L	LEO
S	Safe At Home

Table A-33 Telephone Types

Code	Description
D	Daytime
E	Evening
C	Cell
F	Fax

Table A-34 Name Prefixes

Prefix	Description
MS	Ms
MISS	Miss
MRS	Mrs.
MR	Mr.
REV	Reverend
FR	Father
DR	Doctor
ATTY	Attorney
PROF	Professor
HON	Honorable
MSGR	Monsignor
SR	Sister
BR	Brother
AMB	Ambassador
PVT	Private
CPL	Corporal
SGT	Sergeant
MAJ	Major
CAPT	Captain
CMDR	Commander
LT	Lieutenant
LTCOL	Lieutenant Colonel
COL	Colonel
GEN	General