EAVS Section B Data Standard Documentation

Release 0.10.0

Jared Marcotte

Jan 21, 2020

Table of Contents

Welcome to the EAVS Section B Data Standard documentation. This data standard aims to provide a simple way to provide data for Section B of the Election Administration and Voting Survey, or EAVS. While the EAVS collects data that allows for comparison of election administration at the state-level, the EAVS can tell us very little about the voter-level transactions and what contributes to a voter successfully receiving and casting a ballot. This standard gives states and jurisdictions a mechanism to provide data in a structured format that allows for deeper analysis of voter behavior and isolate potential pitfalls in the administrative process.

CHAPTER 1

CSV Documentation

- *BallotTransmissionType*
- *BallotTransmissionOtherType*
- *BallotType*
- *ElectionDate*
- *ElectionName*
- *JurisdictionId*
- *JurisdictionIdType*
- *JurisdictionName*
- *StateName*
- *VoterMailingCountry*
- *VoterType*
- *WasApplicationRejected*

1.1 Getting Started

If you're a state or local election official reading this, first, thank you for your work and your interest in the project. The data required to participate in this project is typically found in absentee voter roster files, voter registration databases, or voter history files. Sometimes all three... and maybe others. The standard looks to capture Uniformed and Overseas Citizen Absentee Voting Act (UOCAVA) transactions, such as:

- when a voter requests a ballot;
- when the application is processed;
- to what country the ballot is sent and when;
- why an application/ballot is rejected; and,
- whether a voter is military or overseas citizen.

To make the process of collecting data easier, this project borrows lessons from the Voting Information Project's (VIP's) approach to data collection: use a flat file format for data collection and push any transformations downstream.

1.2 Naming Convention

Due to the many issues around elections data collection and cleanliness, much of the data still requires some form of human oversight. Therefore, having a relatively accessible file naming convention is beneficial. While this is not a hard-and-fast rule, anyone producing data in this standard should use the following naming convention, which borrows from a yet-unreleased version of the Voting Information Project's documentation.

```
esbdata-${ELECTION_DATE}-${STATE}[-${LOCAL}].{csv|zip}
```

An explanation of each of the segments of the file naming convention above are as follows:

- `${ELECTION_DATE}` - The date of the election in ISO 8601 format.
- `${STATE}` - The full state name (e.g. Alaska, Arkansas, etc...) and not the abbreviation. If there are spaces in the state name, they should be substituted with underscores (e.g. New York -> New_York).

- `${LOCAL}` (optional) - This additional identifier should be used if the file contains data from a specific jurisdiction. As with `${STATE}` above, all spaces should be substituted with underscores. For example, if the data contained in the file only covers Maricopa County, AZ for the November 6, 2012 election, the file name would be `esbdata-2012-11-06-Arizona-Maricopa_County.csv`.
- `{csv|zip}` - If the file is an uncompressed CSV document, the extension should be `.csv`. If the file is zipped, the file extension should end with `.zip`.

For a final example, `esbdata-2012-11-06-Iowa.zip` denotes Iowa's data for the Nov 6, 2012 election that has been compressed.

1.3 Data Fields

The following fields are possible data points for each transaction that the standard hopes to collect.

1.3.1 ID

Data Type: string

This field is required and must be unique.

Unique identifier for the transaction (i.e. not the voter). IDs should remain consistent, regardless of when an export is created.

1.3.2 VoterID

Data Type: string

This field is required.

Identifier that uniquely identifies a voter, but is not required to be unique within the dataset. This number must be obfuscated from the original voter identifier in the jurisdiction's registration database. This field helps analyze how effective each type of transaction is for an individual voter. In addition, if there are questions about a particular voter's transactions, an organization could provide this identifier back to the jurisdiction to glean more information about the voter's experience.

1.3.3 ApplicationRequestMethod

Data Type: string

This field is required.

Specifies the method by which the application request was received. Possible values are **email**, which signifies the application was received via email; **fax**, which denotes the application was received by fax; **in-person**, which means the application was delivered by a person (NB: this does not necessarily signify the voter themselves delivered the application); **mail**, which signifies the application was received via mail; **online**, which means the application was submitted via online form or system; **phone**, which means the request was placed via phone call; **untracked**, which signifies that this variable is not tracked in the system; and, **other**, which is a catch-all for any values that fall outside of the other values.

The value of `ApplicationRequestMethod` must be one of the following:

- email
- fax

- in-person
- mail
- online
- phone
- untracked
- other

1.3.4 ApplicationRequestOtherMethod

Data Type: string

If *ApplicationRequestMethod* fails to cover all of the possible request methods for a particular voter transaction, this field specifies an alternate method by which the application request was received.

1.3.5 ApplicationRequestPostmarkDate

Data Type: date

Specifies the date the request was postmarked. If there is any confusion as to the whether the date the application was recorded was the date on the postmark or the date the application was physically received by the election administration entity, assume postmark and use this field rather than *ApplicationRequestReceivedDate*.

1.3.6 ApplicationRequestProcessedDate

Data Type: date

Specifies the date the request was processed by the local or state election official.

1.3.7 ApplicationRequestReceivedDate

Data Type: date

Specifies the date the application request was received by the local or state election official. In mail-only states, this date is the most recent date the voter became an active UOCAVA voter.

1.3.8 ApplicationRequestRejectionType

Data Type: array

This field is required.

If *ApplicationRequestStatusType* is **rejected** or **cured**, then specifies the reason the application was rejected. Possible values are **duplicate**, which denotes that the election official already received an application for the voter, **invalid**, which means the application was invalid, **mismatch-voter-signature**, which means the application signature did not match the voter signature on file, **missing-identification** which means that an unspecified identifier was required and missing from the application, **missing-ssn**, denoting that the Social Security Number was missing from the application, **missing-state-id-number**, which means the state identifier was required and missing from the application, **missing-voter-signature**, meaning the voter did not sign the application, and **other**, which is a catch-all for any values that fall outside of the other values. Multiple values are possible for this field.

The value of `ApplicationRequestRejectionType` must be one of the following:

- duplicate
- invalid
- mismatch-voter-signature
- missing-identification
- missing-ssn
- missing-state-id-number
- missing-voter-signature
- other

1.3.9 ApplicationRequestRejectionOtherType

Data Type: string

If *ApplicationRequestRejectionType* fails to cover all possible types of Application Rejection reasons, this field will list an alternate type.

1.3.10 ApplicationRequestStatusType

Data Type: string

This field is required.

Specifies the current status of the application. Possible values are **accepted**, which denotes that the application was successfully processed; **cured** which means the application was supplemented with additional information to make it valid; **pending**, which signifies that application is still being processed; **rejected** which means the application was not successfully processed.

The value of `ApplicationRequestStatusType` must be one of the following:

- accepted
- cured
- pending
- rejected

1.3.11 ApplicationRequestType

Data Type: string

This field is required.

Specifies the type of ballot application request. Possible values are **fpca**, which means a voter submitted a Federal Postcard Application; **fwab**, which denotes a voter returned a Federal Write-In Absentee Ballot prior to receiving or requesting a ballot through other means; **informal-request**, which signifies a voter requested a ballot through a less formal process, such as a letter or phone call; **nvra**, which means a voter used a National Voter Registration (Act) form; **state-application**, which denotes a state form used for ballot requests; and **untracked**, which means the type of application is not tracked.

The value of `ApplicationRequestType` must be one of the following:

- fpca

- fwab
- informal-request
- nvra
- state-application
- untracked

1.3.12 BallotRejectionType

Data Type: array

Specifies the reason why the ballot was rejected. Possible values are **mismatch-voter-signature**, which denotes the signature on file did not match the signature on the returned envelope or document; **missing-voter-signature**, which signifies that the voter signature is missing on the returned documents; **not-timely**, which means the ballot was received after the required deadline; **postmark**, which signifies the ballot was postmarked after the required date; **rejected**, which signifies the ballot was rejected for an unspecified reason; **undeliverable**, which denotes the ballot was returned as undeliverable to the address where it was sent; **untracked**, which signifies that this variable is not tracked in the system; **voided-spoiled**, which means the ballot had to be voided or spoiled either by the administration or the voter; **voter-died**, which signifies the voter died thereby impacting the ability to count the ballot (NB: this is largely dependent on state law); **voter-moved**, which signifies the voter moved from their registered address; **voter-unregistered**, which means the voter is not registered to vote; and, **other**, which is a catch-all for any values that fall outside of the other values.

The value of `BallotRejectionType` must be one of the following:

- mismatch-voter-signature
- missing-voter-signature
- not-timely
- postmark
- rejected
- undeliverable
- untracked
- voided-spoiled
- voter-died
- voter-moved
- voter-unregistered
- other

1.3.13 BallotRejectionOtherType

Data Type: string

If *BallotRejectionType* fails to cover all of the possible ways a ballot may be rejected, this field specifies an alternate reason why the reason the ballot was rejected.

1.3.14 BallotReturnDate

Data Type: date

Specifies the date the voter returned the ballot to the local or state election official.

1.3.15 BallotReturnType

Data Type: string

Specifies the method by which the voter returned the ballot to the local or state election official. Possible values are **email**, which signifies the ballot was received via email; **fax**, which denotes the ballot was received by fax; **in-person**, which means the ballot was delivered by a person (NB: this does not necessarily signify the voter themselves delivered the ballot); **mail**, which signifies the ballot was received via mail; **online**, which means the ballot was submitted via online form or system; **untracked**, which signifies that this variable is not tracked in the system; and, **other**, which is a catch-all for any values that fall outside of the other values.

The value of `BallotReturnType` must be one of the following:

- email
- fax
- in-person
- mail
- online
- untracked
- other

1.3.16 BallotReturnOtherType

Data Type: string

If *BallotReturnType* fails to cover all of the possible return types for a particular voter transaction, this field specifies an alternate method by which the voter returned the ballot to the local or state election official.

1.3.17 BallotTransmissionDate

Data Type: date

Specifies the date the local or state election official sent the ballot to the voter.

1.3.18 BallotTransmissionType

Data Type: string

Specifies the method in which the ballot was sent to the voter. Possible values are **email**, which signifies the ballot was transmitted via email; **fax**, which denotes the ballot was transmitted by fax; **in-person**, which means the ballot given to a person; **mail**, which signifies the ballot was transmitted via mail; **online**, which means the ballot was transmitted via online form or system; **untracked**, which signifies that this variable is not tracked in the system; and, **other**, which is a catch-all for any values that fall outside of the other values.

The value of `BallotTransmissionType` must be one of the following:

- email
- fax
- in-person
- mail
- online
- untracked
- other

1.3.19 BallotTransmissionOtherType

Data Type: string

If *BallotTransmissionType* fails to cover all of the possible transmission types for a particular transaction, this field specifies an alternate method in which the local or state election official sent the ballot to the voter.

1.3.20 BallotType

Data Type: string

Specifies the type of ballot transferred to the voter. Possible values are **absentee**, which is a general catch-all for any type of absentee ballot, **federal**, which denotes the voter only received a ballot that allows voting on federal contests, **full**, which denotes a ballot with all possible contests in the voter's assigned precinct; **fwab**, which specifies a Federal Write-In Absentee Ballot, and **provisional**, which denotes a provisional ballot; and, **untracked**, which signifies that this variable is not tracked in the system.

The value of `BallotType` must be one of the following:

- absentee
- federal
- full
- fwab
- provisional
- untracked

1.3.21 ElectionDate

Data Type: date

This field is required.

Specifies the date the election took place.

1.3.22 ElectionName

Data Type: string

This field is required.

Specifies the name of the election.

1.3.23 JurisdictionId

Data Type: string

This field is required.

Specifies a unique identifier for the jurisdiction.

1.3.24 JurisdictionIdType

Data Type: string

This field is required.

Specifies the type of identifier used to identify the jurisdiction in *JurisdictionId*. Possible values are **local**, which dictates the jurisdiction created the identifier, **fips**, which means the identifier follows the Federal Information Processing Standard (FIPS) code, or **ocd-id**, which means the identifier follows the Open Civic Data Division Identifier (OCDID) standard.

The value of `JurisdictionIdType` must be one of the following:

- ocd-id
- fips
- local

1.3.25 JurisdictionName

Data Type: string

This field is required.

Specifies the name of the jurisdiction in which the voter is registered.

1.3.26 StateName

Data Type: string

This field is required.

Specifies the name of the state in which the voter is registered.

The value of `StateName` must be one of the following:

- Alabama
- Alaska
- American Samoa
- Arizona
- Arkansas
- California
- Colorado
- Connecticut
- Delaware

- District Of Columbia
- Federated States Of Micronesia
- Florida
- Georgia
- Guam
- Hawaii
- Idaho
- Illinois
- Indiana
- Iowa
- Kansas
- Kentucky
- Louisiana
- Maine
- Marshall Islands
- Maryland
- Massachusetts
- Michigan
- Minnesota
- Mississippi
- Missouri
- Montana
- Nebraska
- Nevada
- New Hampshire
- New Jersey
- New Mexico
- New York
- North Carolina
- North Dakota
- Northern Mariana Islands
- Ohio
- Oklahoma
- Oregon
- Palau
- Pennsylvania

- Puerto Rico
- Rhode Island
- South Carolina
- South Dakota
- Tennessee
- Texas
- Utah
- Vermont
- Virgin Islands
- Virginia
- Washington
- West Virginia
- Wisconsin
- Wyoming

1.3.27 VoterMailingCountry

Data Type: string

Specifies what country to which the ballot was mailed. The country should be a recognized country name according to ISO 3166.

1.3.28 VoterType

Data Type: string

Specifies whether the voter is domestic or overseas military (or, if the domestic or overseas designation is unknown, simply whether the voter is military), overseas citizen, or other.

The value of `VoterType` must be one of the following:

- domestic-military
- military
- overseas-citizen
- overseas-military
- other

1.3.29 WasApplicationRejected

Data Type: boolean

Deprecated. Specifies whether the application request was accepted or rejected.

CHAPTER 2

ESB Formatting

2.1 ESB File Structure

The CSV file must be comma-delimited, UTF-8 (with BOM omitted) .csv files, named according to the CSV documentation.

2.2 Header Row

The first line of the file, or header row, must conform to the following:

- The CSV file must include a header row containing a comma-delimited list of all data fields. NB: All data fields must be included in the header row, even if those fields are never used.

- All fields names must be in the same letter casing as they appear in the CSV documentation, e.g. `BallotReturnDate` is acceptable, but `ballotreturndate` is not.
- All fields must be emitted in the order they appear in the CSV documentation.

2.3 Control Characters

The CSV file must be free of control characters, except for newlines, which should follow UNIX syntax, i.e. `\n`.

2.4 Missing Values

If a data point specified by a field is unavailable, place the value `na` in that field. Note that:

- `na` should not be used in a field when a value is not expected, such as an `OtherType`, when the base field is not set to `other`.
- `na` must not be used in required fields.

2.5 Data Types

Each `field` belongs to a particular data type. These data types constrain the allowed values for a particular field. For example, a field of type `date` must follow the conventions of a ISO 8601.

2.5.1 array

An `array` is used to store multiple values in a single field.

```
[""first-value"",""second-value""]
```

NB: Even if a single value is to be emitted, it still must be surrounded by brackets.

2.5.2 boolean

A `boolean` represents a true or false value. Possible values are `true` or `false`.

2.5.3 date

All dates in the ESB feed must conform to the ISO 8601 standard on dates. The format model of such a date is `YYYY-MM-DD`.

2.5.4 string

A `string` is a sequence of characters. Most fields in the ESB Data Standard are strings. Be sure to read the documentation for each `string` field, as some fields only allow values that are specified in an enumerated list. String fields should be free of leading or training whitespace.