mCDF Profile for Contest Selection Capture

This profile of the NIST Cast Vote Records Common Data Format 1.0 is designed to support the capture of contest selections encoded onto paper vote records using various machine readable symbologies. The profile includes only the CVR classes and properties required to convey the following:

* The election associated with the ballot;
* The ballot style associated with the ballot;
* The election authority who created the ballot style;
* The contest option selections made, including any write-ins; and
* An optional link to the ballot definition associated with the ballot style

## Reading the profile

This profile is a subset of the NIST Cast Vote Records CDF Data Model. The profile cannot be fully understood without a working understanding of that document. Each field in a segment (detailed below) contains a “CDF Mapping” column, which points back to the CDF property profiled, if any. It is strongly recommended to look up these cross-references to understand the correct use of each property.

The profile is structured in terms of messages, segments, data types and enumerations. Each segment is accompanied by a table listing each field or component, if it is required, whether it has a default value, and its name. Messages, segments, and data types are often referenced using three character codes.

## Messages

### CSC (Contest Selection Capture)

The mCDF Profile for Contest Selection Capture supports a single message, Contest Selection Capture (CSC) to capture votes. Its structure is given by the following abstract syntax:

NS1 BAL ELE? CCO\*

That is, a valid CSC message consists of a NS1 segment, a BAL segment, an optional ELE segment and zero or more CCO segment(s).

NB: The NS1 Segment is defined in the mCDF specification.

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

|  |  |
| --- | --- |
| **Field** | **Value** |
| NS1-3 | CSC |
| NS1-4 | 1 |

#### Segments

##### BAL (BallotStyle)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | **Datatype** | **Multiplicity** | **Default** | **Attr Name** | **CDF Mapping** |
| 1 | String | 1..1 |  | BallotStyleId | BallotStyle::ExternalIdentifier |
| 2 | Integer | 0..1 | 1 | Side |  |
| 3 | Integer | 0..1 | 1 | Sheet |  |
| 4 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 1..1 |  | BallotFormat Code | BallotFormat::ExternalIdentifier |
| 5 | anyURI | 0..1 |  | DefinitionUri |  |

**BAL-1 BallotStyleId**

For associating a mCDF instance with the ballot style in the ballot definition. This structure is somewhat ambiguous as CVR does not have an equivalent BallotStyle class to BallotDefinition.

**BAL-2 Side**

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

**BAL-3 Sheet**

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

**BAL-4 BallotFormat Code**

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

**BAL-5 DefinitionUri**

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

##### ELE (Election)

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | **Datatype** | **Multiplicity** | **Default** | **Attr Name** | **CDF Mapping** |
| 1 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 1..1 |  | Election Code | Election::ExternalIdentifier |
| 2 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 0..1 |  | ElectionScope Code | GpUnit::ExternalIdentifier |

**ELE-1 Election Code**

For associating the mCDF instance with a particular election.

**ELE-2 ElectionScope**

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

##### CCO (CVRContest)

For associating a mCDF instance with the selections made in a particular contest during a voting session.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | **Datatype** | **Multiplicity** | **Default** | **Attr Name** | **CDF Mapping** |
| 1 | CVRContestSelection ([CCS](#X0d10c679e6153eb2ba15f01069c9cdb357a46a2)) | 1..\* |  | CVRContestSelection |  |
| 2 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 0..1 |  | Contest Code | Contest::Code |

**CCO-1 CVRContestSelection**

For associating one or more selections with a contest.

**CCO-2 Contest Code**

The code associated with the contest

#### Abstract Data Types

#### CDE (Code)

Used across segments to provide short identifiers for various data points. When a codebook is provided, these codes can be looked up to find the values they stand for.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | **Datatype** | **Multiplicity** | **Default** | **Attr Name** | **CDF Mapping** |
| 1 | String | 0..1 |  | Value | Code::Value |
| 2 | IdentifierType ([IDT](#X3f2fda675897d70db26215e8df1f91ae30d4fe0)) | 0..1 | 2 | Type | Code::Type |
| 3 | String | 0..1 |  | OtherType | Code::OtherType |

#### CCS (CVRContestSelection)

For associating one or more selections with a contest. The CVRContestSelection is used to convey a selection made for a single contest. The selection is indicated by providing a value for ContestSelection::Code. All other fields are optional. SelectionPosition::NumberVotes may be provided if the number of votes allocated is an integer other than 1. SelectionPosition::Rank may be provided if the contest option selection is ranked (e.g. Rank Choice Voting is used). SelectionPosition::FractionalVotes may be provided if the number of votes includes a fractional component. CVRWriteIn::Text may be provided if the voter has made a write-in entry.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | **Datatype** | **Multiplicity** | **Default** | **Attr Name** | **CDF Mapping** |
| 1 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 0..1 |  | Contest Selection Code | ContestSelection::Code |
| 2 | Integer | 0..1 | 1 | NumberVotes | SelectionPosition::NumberVotes |
| 3 | Integer | 0..1 |  | Rank | SelectionPosition::Rank |
| 4 | FractionalNumber | 0..1 |  | FractionalVotes | SelectionPosition::FractionalVotes |
| 5 | boolean | 0..1 |  | IsGenerated | SelectionPosition::IsGenerated |
| 6 | String | 0..1 |  | WriteInText | CVRWriteIn::Text |
| 7 | Code ([CDE](#Xaa6b8efb3c295a0dcb65f9f4730b8c7c84fbf67)) | 0..1 |  | Selection Position Code | SelectionPosition::Code |

#### Enumerations

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

##### BST (BallotSideType)

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

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

##### IDT (IdentifierType)

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

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

## Usage Notes

* It is considered an error to produce multiple CCO segments for the same contest. All selections in a given contest should be expressed in a single CCO segment.
* There is no way to convey absence of a selection, as in the CVR CDF.

# mCDF Processing

* Sequence numbers must be unique per sheet. If a VVPAT is used, sequence numbers should be unique across the roll.

# Examples

## Basic Usage

The following example incorporates the one above and demonstrates use of the “CSC” message. The CSC instance will be encoded on the front of a single sheet ballot style, identified using a `local-level` identifier. The ballot format identifier (`local-level`) is also given to associate the ballot with the correct physical details in the ballot definition. The election code is given as a `local-level` identifier. Finally, the election jurisdiction is given as a `ocd-id` code.

Graphical user interface, text, application, chat or text message

Description automatically generated

In this example, the ballot contains four candidate contests, all of which use n-of-m voting method. The first three contests contain a single vote, the fourth contest contains a regular vote and a write-in.

|  |
| --- |
| NS1|^~\&;|CSC|1|1;BAL|112115|1|1|1-ess|http://go.usa.gov/ballotdef;ELE|331332219|ocd-division/country:us/state:ca/sldu:assigned-31^4;CCO|CRC|CON-1;CCO|CHL|CON-2;CCO|CMA|CON-3;CCO|CMT~^^^^^GINTAUTUS \U0104\\U017D\DUOLAS|CON-4; |

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

Information about the election (ELE) with which the BallotStyle is associated concludes the message and is composed of the Election Code (331332219) and the ElectionScopeCode (ocd-division/country:us/state:ca/sldu:assigned-31) with an explicitly specified Type indicating that the scope is represented with a OCD-ID code (4). ^ is used to delimit a component in a field. All segments end with a semicolon (;).

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

Voter selections in each contest follow, using CCO. As multiple selections are sometimes being recorded in the CCO segment, we can use a repetition delimiter (~) to avoid repeating CCO for each one. Contests 1-3 allow no more than one selection and so only the selected candidate ID is recorded (CRC, CHL, and CMA). The fourth contest permits two votes, and two are selected, so the CCO segment includes two contest selection codes. The first choice (CMT) follows the pattern of the previous contests, but the second is a write-in selection. This is indicated by the value being recorded as a string in CCO-1.6 rather than a candidate ID. In this case, the candidate's name includes special characters which are expressed by escaped Unicode code point references (\U0104\, \U017D\).

## Usage with Continuations

It is possible that the symbology used reaches a technical maximum number of characters that it can store before the logical message completes. In this case, the mCDF continuation feature can be used. In this example, the symbology used has a maximum limit of 300 characters. The full “CSC” message is recorded first from which two physical mCDF messages are constructed to hold the data. The message is identical to the previous example except the fourth contest includes selections for two write in candidates.

Full message:

|  |
| --- |
| NS1|^~\&;|CSC|1|1;BAL|112115|1|1|1-ess|http://go.usa.gov/ballotdef;ELE|331332219|ocd-division/country:us/state:ca/sldu:assigned-31^4;CCO|CRC|CON-1;CCO|CHL|CON-2;CCO|CMA|CON-3;CCO|^^^^^ Salvador Felipe Jacinto Dal\U00ED\ y Domenech~^^^^^ Pablo Diego Jos\U00E9\ Francisco de Paula Juan Nepomuceno Mar\U00ED\a de los Remedios Cipriano de la Sant\U00ED\sima Trinidad Ruiz y Picasso |CON-4; |

Fragment 1:

|  |
| --- |
| NS1|^~\&|CSC|1|1;BAL|112115|1|1|1-ess|http://go.usa.gov/ballotdef;ELE|331332219|ocd-division/country:us/state:ca/sldu:assigned-31^4;DSC|123; |

Fragment 2:

|  |
| --- |
| NS1|^~\&;|CSC|1|1|123; CCO|CRC|CON-1;CCO|CHL|CON-2;CCO|CMA|CON-3;CCO|^^^^^ Salvador Felipe Jacinto Dal\U00ED\ y Domenech~^^^^^ Pablo Diego Jos\U00E9\ Francisco de Paula Juan Nepomuceno Mar\U00ED\a de los Remedios Cipriano de la Sant\U00ED\sima Trinidad Ruiz y Picasso |CON-4; |

# Mapping from mCDF to CVR

The CSC message stands at the intersection of the ballot definition and the cast vote record. In order to produce a cast vote record, information about the election such as contests and candidates is required, and which resides in the ballot definition. This example will show how structures in the CSC message map back to the NIST Ballot Definition, and the steps required to map forward into a NIST Cast Vote Record.

This section references fragments from code examples mapping\_csc\_mcdf\_to\_bd.xml and mapping\_csc\_mcdf\_to\_cvr.xml.

Consider the following mCDF CSC message:

NS1|^~&;|CSC|1|1;BAL|052001|||BF1|http://go.usa.gov/Tla9;ELE|20141104|39153;CCO|1HBR~1HDW~1HJD|9CC;

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

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

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

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

BAL-2 and BAL-3 represent details more appropriate to OMR ballots and are omitted. BAL-4 maps to BallotDefinition::BallotFormat::ExternalIdentifier, represented as the following XML fragment:

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

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

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

ELE-1 maps to the following BD XML fragment:

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

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

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

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

The CCO segment, unlike the others, maps to fields in both the ballot definition and cast vote record.

The CCO-1 field includes components, defined by the data type CCS (CVRContestSelection). CCO-1.1 represents a BallotDefinition::ContestOption::Code, which is defined to the ballot definition's ExternalIdentifier seen earlier. The three selections in the contest and the contest code (CCO-2), map to the following CVR fragment:

<Contest xsi:type="CandidateContest" ObjectId="cc-1">  
 <ContestOption xsi:type="CandidateOption" ObjectId="cs-1hbr">  
 <ExternalIdentifier>  
 <Type>local-level</Type>  
 <Value>1HBR</Value>  
 </ExternalIdentifier>  
 </ContestOption>  
 <ContestOption xsi:type="CandidateOption" ObjectId="cs-1hdw">  
 <ExternalIdentifier>  
 <Type>local-level</Type>  
 <Value>1HDW</Value>  
 </ExternalIdentifier>  
 </ContestOption>  
 <ContestOption xsi:type="CandidateOption" ObjectId="cs-1hjd">  
 <ExternalIdentifier>  
 <Type>local-level</Type>  
 <Value>1HJD</Value>  
 </ExternalIdentifier>  
 </ContestOption>   
 ...  
 <ElectionDistrictId>ed-1</ElectionDistrictId>  
 <ExternalIdentifier>  
 <Type>local-level</Type>  
 <Value>9CC</Value>  
 </ExternalIdentifier>  
 <Name>Contest A</Name>  
 <VoteVariation>n-of-m</VoteVariation>  
 <VotesAllowed>3</VotesAllowed>  
</Contest>

Note that throughout these mappings, all have been in to the ExternalIdentifier (BallotDefinition) or Code (CastVoteRecord), never to the ObjectId. ObjectId is used for internal wiring of the file and should never been used to convey meaning.

With the Contest and ContestOption identified, the production of a cast vote record for this ballot is a matter of creating equivalent structures in the NIST CVR (e.g. Contest and ContestSelection) and wiring them to a CVR record.

<CVR>  
 <CurrentSnapshotId>cvr-1</CurrentSnapshotId>  
 <CVRSnapshot ObjectId="cs-1">  
 <CVRContest>  
 <ContestId>cc-1</ContestId>  
 <CVRContestSelection>  
 <ContestSelectionId>cs-1hbr</ContestSelectionId>  
 <SelectionPosition>  
 <HasIndication>yes</HasIndication>  
 <NumberVotes>1</NumberVotes>  
 </SelectionPosition>  
 </CVRContestSelection>  
 <CVRContestSelection>  
 <ContestSelectionId>cs-1hdw</ContestSelectionId>  
 <SelectionPosition>  
 <HasIndication>yes</HasIndication>  
 <NumberVotes>1</NumberVotes>  
 </SelectionPosition>  
 </CVRContestSelection>  
 <CVRContestSelection>  
 <ContestSelectionId>cs-1hjd</ContestSelectionId>  
 <SelectionPosition>  
 <HasIndication>yes</HasIndication>  
 <NumberVotes>1</NumberVotes>  
 </SelectionPosition>  
 </CVRContestSelection>  
 </CVRContest>  
 <Type>original</Type>  
 </CVRSnapshot>  
 <ElectionId>ele-20141104</ElectionId>  
</CVR>

While CCO-1.2 represents the NumberVotes, is has been omitted and taken the default value of 1. HasIndication simply means there is some indication for the selection of the options given, and will always be yes for output from ballot marking devices.

NB: ContestOption and ContestSelection refer to the same concept. Future versions of Cast Vote Record and Election Results Reporting CDF will be updated to use the term ContestOption.

## Retrieval with XPath

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

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